A kismalac és a farkasok – Az nmap használata

Cikkemben megpróbálom a teljesség igénye nélkül, de a lehetőségek legszélesebb skáláját érintve bemutatni az ***nmap*** (***Network Mapper***) programot, vagyis egy olyan népszerű és sokat próbált hálózati eszközt, melynek rendszeres és szakszerű használatával jelentősen növelhetjük hálózatba kötött számítógépeink biztonságát.
Egyesek szerint a rendszergazdákkal szemben a hackerek egyik legnagyobb előnye az, hogy ők 24 órában tevékenykednek, s így mindig lépéselőnyben vannak. Ezen lépéselőny leküzdésére a rendszergazdák a hackerek által használt eszközöket vetik be saját hálózataik ellen, hogy kiderítsék, milyen sebezhetőségre lelhet rá egy betolakodó, amíg ők otthon éppen a legédesebb álmukat alusszák.
Ehhez a rendszerellenőrzéshez jelenleg az egyik legalkalmasabb eszköz az ***nmap*** pásztázó (portscanner), ami ellenőrzi, hogy egy adott számítógépen mely portok vannak nyitva, azokon milyen, és hányas verziószámú szolgáltatások üzemelnek, valamint milyen operációs rendszer fut rajta. Ezek alapján talán már érthető, miért elengedhetetlen az ismerete.

Mik azok a portok?

Mielőtt elmélyülnénk a program adta lehetőségekben, tegyünk egy rövid hálózatelméleti kitekintést. A portok vagy kapuk lehetővé teszik, hogy egy adott ***IP*** címmel rendelkező számítógépen egyszerre több különböző szolgáltatást is el lehessen érni. Minden démon és program, ami hálózati kapcsolatokat használ, legyen az ***TCP*** vagy ***UDP*** kapcsolat, portokon keresztül kommunikál a hálózaton. A kapuszámok 1-től 65536-ig terjednek. Ezen belül három csoportot lehet megkülönböztetni, az 1-1023-ig terjedő ún. rendszerkapukat, ahol csak rendszergazdai jogosultságú folyamatok működhetnek, az 1024-49151-ig terjedő skálát, ami a nem rendszergazdai jogokkal bíró folyamatok kapuit tartalmazza, és az ezen felüli ún. magánkapukat, amiken azok a szolgáltatások futnak, melyek nem kapcsolódnak az internethez. A ***TCP*** és az ***UDP*** protokollhoz külön-külön 65536 port érhető el, így előfordulhat, hogy ugyanazon a számú kapun szolgáltat egy ***TCP*** és egy ***UDP*** folyamat.
Szót kell még ejteni a hivatalos port-hozzárendelésekről, melyeket Linux rendszereken a ***/etc/services*** állomány tartalmaz. A lényege, hogy egy adott számú ***TCP*** vagy ***UDP*** kapun minden rendszeren ugyanaz a szolgáltatás fut. Például a 23-as ***TCP*** porton a ***telnet*** szolgáltatás érhető el. Ami miatt ez nem egy szabvány, az az, hogy egy szolgáltatást könnyen áthelyezhetünk egy másik kapura. Ezért ha azt látjuk, hogy egy számítógépen nyitott a 23-as kapu, azaz lehet rá csatlakozni, még nem biztos, csak valószínű, hogy azt egy ***telnet*** démon használja.
A számítógép egy másikhoz történő kapcsolódásakor forrás- és célkapukat használ. A célkapu az előbb említett némileg szabványos port-hozzárendelések közül az egyik, amin egy szolgáltatást szeretnénk elérni. A forráskapu – ahonnan a kapcsolatot kezdeményezzük – sorszáma előre nem meghatározott, alkalmazásfüggő, amit a rendszer dinamikusan oszt ki.

Mit jelent a kapupásztázás?

A kapupásztázás leegyszerűsítve annyit tesz, hogy megpróbálunk a célgép minden egyes portjára csatlakozni. Ha sikerrel járunk, elkönyvelhetjük, hogy a port nyitott, sikertelen csatlakozás esetén pedig azt, hogy a vizsgált kapu zárva van.
Legegyszerűbb példa erre ***TCP*** portok estében a ***telnet*** program, amivel minden egyes portra csatlakozva ellenőrizhetjük, fut-e azon valamilyen szolgáltatás. Beláthatjuk, hogy ez nem egy kényelmes és fejlett módja a felderítésnek.
Itt jön a képbe ***Fjodor*** és az általa írt ***nmap*** ➲ http://insecure.org/nmap/ pásztázó. Az ***nmap*** három különböző fázisban gyűjt információt a célrendszerről:

- Kiszolgáló felderítése (***host scan***): A hálózatban elérhető számítógépek felderítése.
- Kapupásztázás (***port scan***): Egy adott számítógépen futó szolgáltatások és azok verzióinak felderítése.
- Operációs rendszer ujjlenyomat általi azonosítása (***OS scan***): Egy adott számítógépen futó operációs rendszer felderítése.

Az nmap beszerzése és telepítése

Az ***nmap*** népszerűségének köszönhetően minden ***Linux*** terjesztésnek a részét képezi. Ezenkívül elérhető még ***Windows***, ***Free/Open/NetBSD***, ***MacOS X***, ***Solaris***, ***Amiga*** és ***HP-UX*** rendszerekre is. ***Linux*** alatt a legegyszerűbben az általunk használt disztribúció csomagkezelőjével telepíthetjük. Ha a legfrissebb stabil változatot szeretnénk használni, töltsük le a forrást a ➲ http://insecure.org/nmap/download.html oldalról, majd a kicsomagolás után a forráskód könyvtárába lépve a

```
./configure
make
su root
make install
```

parancsokkal telepítjük.

A grafikus felület kedvelőire gondolva az *nmap* rendelkezik grafikus kezelőfelülettel is, de mivel ez nem tartalmaz annyi lehetőséget és nem olyan finoman hangolható, mint parancssorból, ezért itt csak megemlítjük a létezését. Az *nmap* nem rendelkezik konfigurációs fájllal, minden opciót parancssori kapcsolók segítségével adhatunk meg. Mivel a program nagyrészt nyers csomagokat küld a hálózatra, futtatásához több esetben rendszergazdai hozzáférésre lesz szükségünk. Most pedig a három fő pásztázási fázist áttekintve ismerkedjünk meg a programban rejlő lehetőségekkel. Pataméterezésének az alaplogikája a következő:

```
nmap [Pásztázás típusa(i)]
➥[Kapcsolók] {Célgép(ek)}
```

Célgépek meghatározása

Mindenek előtt azt kell tudnunk, hogyan adhatjuk át az *nmap* számára a pásztázni kívánt gépek neveit. Mindez történhet egyszerű felsorolással, vagy egy már előre elkészített lista beolvasásával. Az *nmap* nagyon rugalmasan kezeli a célgépek IP címeit. A 10.200.21.8, 10.200.21-23.8, 10.200.21.0/24, 10.200.21.1,2,3,4,21,54 mind elfogadott formátum.

- `-iL <lista.txt>`: A listában felsorolt számítógépek ellenőrzése
- `-iR <szám>`: `<szám>` darabnyi véletlen *IP* címet generál, és azokat pásztázza végig. Akkor hasznos, ha az interneten olyan számítógépeket keresünk, amelyeken nyitva van egy adott port. Ha a szám értéke nulla, végtelen számú *IP* címet generál. A multicast és belső hálózati *IP* címek automatikusan át lesznek ugorva.
- `--exclude <ipcím1,ipcím2>`: Ide jönnek vesszővel elválasztva azoknak a gépeknek az *IP* címei, amiket ki akarunk hagyni a tartomány vizsgálatakor. Ez akkor hasznos, ha egy teljes hálózatot ellenőrzünk, de néhány gépre vagy alhálózatra nem mi felügyelünk.
- `--excludefile <kihagy.txt>`: Ugyanaz, mint az előző, csak egy fájlban szerepel az ellenőrzésből kihagyni kívánt számítógépek listája.

Kiszolgáló felderítése

- `-sP` ***(Ping scan)***: Végigpingeli azoknak a számítógépeknek a listáját, amit megadunk neki, ezáltal kideríthető, melyek a hálózaton aktuálisan elérhető gépek.
- `-P0` ***(P nulla)***: A kapupásztázást nem előzi meg az aktivitást ellenőrző ***ping***. Akkor hasznos, ha tudjuk, hogy a célszámítógép üzemel, illetve ha az úgy van „tanítva", hogy ne válaszoljon a ***ping***-re. Az ***nmap*** ezzel a kapcsolóval minden esetben megpróbálja pásztázni az adott gépet.
- `-PS <portlista>` ***(TCP SYN Ping)***: SYN jelzőbittel ellátott csomagot küld a gép adott portjára. Ez a port alapértelmezetten a 80-as port, de a <portlistában> megadható, hogy mely portokra küldje ezeket a csomagokat. Ha RST csomag érkezik vissza, zárva van a port, ha SYN/ACK csomag, akkor nyitva.
- `-PA <portlista>` ***(TCP ACK Ping)***: ACK jelzőbites csomagok használata a pásztázáskor. Működése megegyezik a ***SYN Pingével***. Érdemes mind a kettőt használni, mert sokszor a ***SYN*** csomagokat eldobják a tűzfalak.
- `-PU <portlista>` ***(UDP Ping)***: Üres ***UDP*** csomagot küld a 31338-es portra, ezzel ellenőrizve a gép állapotát.
- `-PR` ***(ARP Ping)***: Ethernet hálózatok ellenőrzésekor ez a javasolt eljárás. Olyannyira, hogy az ***nmap*** automatikusan erre vált át, ha azt érzékeli, ha helyi hálózaton pásztázunk működő gépek után.

Kapupásztázási technikák

Az ***nmap*** a portok állapotát 6 különféle csoportba sorolja.

- ***Nyitott (open)***: Valamilyen ***TCP*** vagy ***UDP*** protokollt használó szolgáltatás fut az adott porton. A hackerek számára ez jelenti a főnyereményt, hiszen minden nyitott port egy potenciális támadási lehetőség.
- ***Zárt (closed)***: Elérhető, de nem fut rajta semmilyen szolgáltatás.

1. Lista Egyszerű kapupásztázás pingelés nélkül

```
csaba@lcs:~$ nmap -sT -P0
➥scanme.nmap.org

Starting Nmap 4.20
➥( http://insecure.org ) at
➥2006-12-28 18:08 CET
Interesting ports on
➥scanme.nmap.org
➥(205.217.153.62):
Not shown: 1691 filtered
➥ports
PORT      STATE     SERVICE
22/tcp    open      ssh
25/tcp    closed    smtp
53/tcp    open      domain
70/tcp    closed    gopher
80/tcp    open      http
113/tcp   closed    auth

Nmap finished: 1 IP address
➥(1 host up) scanned in
➥68.421 seconds
csaba@lcs:~$
```

- ***Szűrt (filtered)***: Az ***nmap*** nem tudja eldönteni, hogy az adott port nyitva van-e, ugyanis valamilyen csomagszűrési eljárás megakadályozza, hogy elérje a kaput.
- ***Nem szűrt (unfiltered)***: A port elérhető, de az ***nmap*** nem tudja eldönteni, hogy nyitva van-e vagy sem. Csak a tűzfalszabályok felfedezésére használatos ***ACK*** pásztázás sorolja ebbe a csoportba a portokat.
- ***Nyitott|Szűrt (open|filtered)***: nem lehet eldönteni, h a port nyitva van, vagy szűrt. Az ***UDP***, ***IP*** protokoll, ***FIN***, ***Null***, és ***Xmas*** pásztázás sorolja ebbe a csoportba a kapukat.
- ***Zárt|Szűrt (closed|filtered)***: Nem eldönthető, hogy a port zárt vagy szűrt. Az ***Idle pásztázás*** által használt jelölés.

  - ***-sS (TCP SYN scan)***: Az alapértelmezett pásztázási eljárás, ami SYN csomagokat használ. Gyors és pontos. A ***SYN*** pásztázás nem fejezi be a háromlépéses ***TCP*** kézfogást.

A második lépés, a ***SYN/ACK*** csomag után ***RST*** csomaggal zárja a kapcsolatot.

- `-sT` ***(TCP Connect() scan)***: Az alapértelmezett pásztázási eljárás abban az esetben, ha nincs megfelelő jogosultságunk ahhoz, hogy nyers csomagokat tudjunk küldeni a hálózatra, illetve ***IPv6*** hálózatok pásztázásakor. A `Connect()` rendszerhívást használva teljes kapcsolatot épít fel az ellenőrizni kívánt géppel, ezáltal könnyebb felfedezni.
- `-sU` ***(UDP scan)***: Annak ellenére, hogy a legnépszerűbb hálózati szolgáltatások ***TCP*** protokollt használnak, az ***UDP***-t használó programok közül is akad néhány híresebb. Ilyen például a ***DNS***, a ***DHCP***, ***SNMP***. Úgy működik, hogy küld egy üres ***UDP*** fejlécet a célportra, és a válaszból következtet a port állapotára. Működik a ***TCP*** pásztázásokkal együtt.
- `-sF` ***(TCP FIN scan)***: A ***FIN*** bit van beállítva a csomagban, vagyis a kapcsolat lezárását jelzi a célgépnek.
- `-sN` ***(TCP Null scan)***: Jelöletlen ***TCP*** csomagot küld a célkapura.
- `-sX` ***(Xmas scan)***: A ***FIN***, ***PSH***, ***URG*** jelzőbitek vannak beállítva a pásztázáshoz használt csomagokban.

Ez az utóbbi három pásztázási módszer ugyanarra az ***RFC 793***-ban leírt eljárásra támaszkodik, amelynek lényege, hogy a zárt portok ***RST*** választ küldenek, míg a nyitott portok eldobják a küldött csomagot.

- `-sA` ***(TCP ACK scan)***: Arra való, hogy feltérképezzük egy adott tűzfal szabályait. Ez a gyakorlatban úgy néz ki, hogy ha tűzfallal nem védett portokat pásztázunk, a nyitott és a zárt kapuk is ***RST*** csomaggal válaszolnak, amit az ***nmap*** a nem szűrt (unfiltered) csoportba sorol, függetlenül attól, hogy nyitott vagy zárt kapuról van szó. Azonban a tűzfallal védett portok vagy nem, vagy ***ICMP*** hibaüzenettel válaszolnak, így ezek a szűrt (filtered) csoportba sorolódnak.
- `-sW` ***(TCP Window scan)***: Hasonló, mint az ***ACK*** pásztázás, azzal a különbséggel, hogy a nyitott és a zárt kapukat képes megkülönböztetni egymástól.
- `--scanflags`: Mi adhatjuk meg, hogy milyen jelzőbiteket szeretnénk beállítani a pásztázáshoz használt csomagokban. Az ***URG***, ***ACK***, ***PSH***, ***RST***, ***SYN***, és ***FIN*** jelzőbitek használata a megengedett.

Pásztázni kívánt portok szűkítése

- `-p <port tartomány>`: Csak bizonyos portok vizsgálata. Megadhatunk konkrét portszámokat, de tartományokat is. Ezen kívül szűkíthetjük a vizsgálatot ***TCP*** vagy ***UDP*** kapuk szerint is. A ***TCP*** portok 21-től 80-ig, míg az ***UDP*** kapuk közül csak az 53-as ellenőrzése: `-p T:21-80,U:53`.
- `-F` ***(Fast (limited port) scan)***: Gyorsabb ellenőrzés azáltal, hogy az ***nmap*** csak azon kapukat vizsgálja, melyek szerepelnek az ***nmap-services*** fájlban.

2. Lista FIN, UDP és RCP pásztázás portszűkítéssel

```
csaba@lcs:~$ sudo nmap -sFUR
➥-p 20-25,53,68,110-140 lcs

Starting Nmap 4.20
➥( http://insecure.org ) at
➥2006-12-28 18:26 CET
Interesting ports on lcs
➥(192.168.1.1):
Not shown: 71 closed ports
PORT    STATE   SERVICE VERSION
21/tcp  open|filtered  ftp
22/tcp  open|filtered  ssh
25/tcp  open|filtered  smtp
139/tcp open|filtered
➥netbios-ssn
68/udp  open|filtered  dhcpc
137/udp open|filtered
➥netbios-ns
138/udp open|filtered
➥netbios-dgm

Nmap finished: 1 IP address
➥(1 host up) scanned in
➥4.865 seconds
csaba@lcs:~$
```

3. Lista A szolgáltatások verzióinak kiderítése

```
csaba@lcs:~$ sudo nmap -sSURV -P0 -T5 lcs

Starting Nmap 4.20 ( http://insecure.org ) at 2006-12-28 18:33 CET
Interesting ports on lcs (192.168.1.1):
Not shown: 3177 closed ports
PORT     STATE     SERVICE      VERSION
21/tcp   open      ftp          ProFTPD 1.3.0
22/tcp   open      ssh          OpenSSH 4.3p2 Debian 5ubuntu1
                                ➥(protocol 2.0)
25/tcp   open      smtp         Postfix smtpd
139/tcp  open      netbios-ssn  Samba smbd 3.X (workgroup: MSHOME)
445/tcp  open      netbios-ssn  Samba smbd 3.X (workgroup: MSHOME)
68/udp   open|     dhcpc
         filtered
137/udp  open      netbios-ns   Microsoft Windows XP netbios-ssn
138/udp  open|filtered netbios-dgm
Service Info: Host:  lcs; OSs: Unix, Linux, Windows

Service detection performed. Please report any incorrect results
➥at http://insecure.org/nmap/submit/ .
Nmap finished: 1 IP address (1 host up) scanned in 51.379 seconds
csaba@lcs:~$
```

4. Lista A számítógépen futó operációs rendszer kiderítése

```
csaba@lcs:~$ sudo nmap -sS -O -T4 -P0 scanme.nmap.org

Starting Nmap 4.20 ( http://insecure.org ) at 2006-12-28
➥18:27 CET
Interesting ports on scanme.nmap.org (205.217.153.62):
Not shown: 1691 filtered ports
PORT       STATE      SERVICE
22/tcp     open       ssh
25/tcp     closed     smtp
53/tcp     open       domain
70/tcp     closed     gopher
80/tcp     open       http
113/tcp    closed     auth
Device type: general purpose
Running: Linux 2.6.X
OS details: Linux 2.6.15-27-686 (Ubuntu Dapper, X86)
Uptime: 11.093 days (since Sun Dec 17 16:14:38 2006)

OS detection performed. Please report any incorrect results at
➥http://insecure.org/nmap/submit/ .
Nmap finished: 1 IP address (1 host up) scanned in 45.142
➥seconds
csaba@lcs:~$
```

Verzióinformációk

Ahogyan azt már említettem a cikk elején, attól függetlenül, hogy létezik egy szabvány arra vonatkozóan, melyik számú *TCP* illetve *UDP* kapun milyen szolgáltatás figyel, még nincs rá garancia, hogy a vizsgált számítógépen is hasonló a helyzet. Egy sima pásztázással csak annyit tudhatunk meg, hogy az adott kapu nyitva van-e vagy sem, de a mögötte lévő szolgáltatás rejtve marad.
Ehhez nyújt segítséget az `nmap -sV` kapcsolója, amely a szolgáltatások nevét és verziószámát hivatott kifürkészni. Itt említhető meg a `-sR` kapcsoló is, aminek a segítségével az úgynevezett *RPC* portok és a szolgáltatásaik verziószáma tudható meg. Ez a kapcsoló is használható együtt a *TCP* és *UDP* pásztázásokkal.

Az operációs rendszer felderítése

Az *nmap* az alapján, hogy a célgép bizonyos csomagokra milyen válaszcsomagot (ujjlenyomatot) generál, képes megállapítani az adott gépen futó operációs rendszer típusát, a kernelverzió számát, valamint az üzemidőt. Talán nem kell túlzottan részleteznem, mennyire megkönnyítheti egy támadó dolgát, ha tudja, milyen operációs rendszerrel van dolga. Ehhez mindössze egy nyitott és egy zárt kapura van szüksége a célrendszeren.
Az ehhez használt kapcsoló a `-O`. Ilyenkor a folyamat a második generációs felderítéssel kezdődik, majd annak sikertelensége esetén az első generációs is lefut. Kérhetjük a programot, hogy kifejezetten csak első vagy második generációs felderítést használjon. Erre valók az `-O1` illetve az `-O2` kapcsolók.
Előfordulhat, hogy az *nmap* nem tudja megállapítani a használt operációs rendszer típusát, ilyenkor a beérkezett ujjlenyomatot jeleníti meg a képernyőn. Ha segíteni akarjuk *Fjodor* fejlesztői munkáját, a ➲ http://insecure.org/nmap/submit oldalon elküldhetjük ezt az ujjlenyomatot a pásztázott rendszer pontos paramétereivel együtt.

Kimenet és egyéb érdekességek

Előfordulhat, hogy a pásztázás eredményét a későbbiekben még látni szeretnénk, ha másért nem, a rendszerbiztonság megerősítése utáni összehasonlítás végett. Ezen kívül ebbe a bekezdésbe kerültek azok a kapcsolók, melyek hasznosak, de nem tartoznak szervesen egyik tárgyalt témához sem.

- `-oN <fájlnév>`: A kimenet egy egyszerű szövegfájlba történő mentése.
- `-oX <fájlnév>`: A kimenetet *XML* formátumba menti.
- `--append-output <fájlnév>`: A fájlt nem írja felül, hanem hozzáfűzi az újabb kimenetet.
- `-v`: *Verbose*, azaz beszédes kimenetet biztosít. Kétszeri ismétlésére az *nmap* további részleteket árul el.
- `-T<0-5>`: A kapcsoló és a mögötte álló szám segítségével közölhetjük a programmal, milyen időközönként küldjön csomagokat a célgépnek. A 0 érték jelenti a legnagyobb időközt, ami 15 perc minden elküldött csomag után, míg az 5 a legkisebbet, ami 5ms várakozási időt jelent. *IDS* rendszerek ellen a 0 és az 1 érték a javasolt, míg a hétköznapokban a 4 tűnik jó választásnak.
- `-D <csalétek1, csalétek2, csalétek3...>`: A célgép szemszögéből úgy látszik, mintha a csalétek gépek pásztáznák a portjait.
- `-6`: Ipv6 protokoll használata. Csak a Ping scan (`-sP`), TCP Connect() scan (`-sT`) és List scan (`-sL`) esetén működik.
- `-A`: Operációs rendszer detektálás és verzióinformációk lekérése. Ugyanaz, mint a `-sV -O` páros. *Fjodor* a későbbiekben tervezi ennek a kapcsolónak a további szolgáltatásokkal történő bővítését, azért, hogy a felhasználóknak ne kelljen annyi kapcsolót megjegyezniük.

Összegzés

Az *nmap* további, olykor megdöbbentő képességeinek felfedezését már az Olvasóra bízom. Remélem a fentiekből remekül kitűnt az az összetettség és rugalmasság, amit az *nmap* kapupásztázó program magában hordoz, és amivel elősegíti számítógépeink hálózati biztonságának megerősítését. Forgassuk nagy haszonnal és megelégedéssel, de ne feledjük, csak jó célra!

Leszkoven Csaba
(leszkovencsaba@gmail.com)