Ellenőrzés pásztázókkal (1. rész)

Az nmap használatával felderítheted rendszered gyenge pontjait.

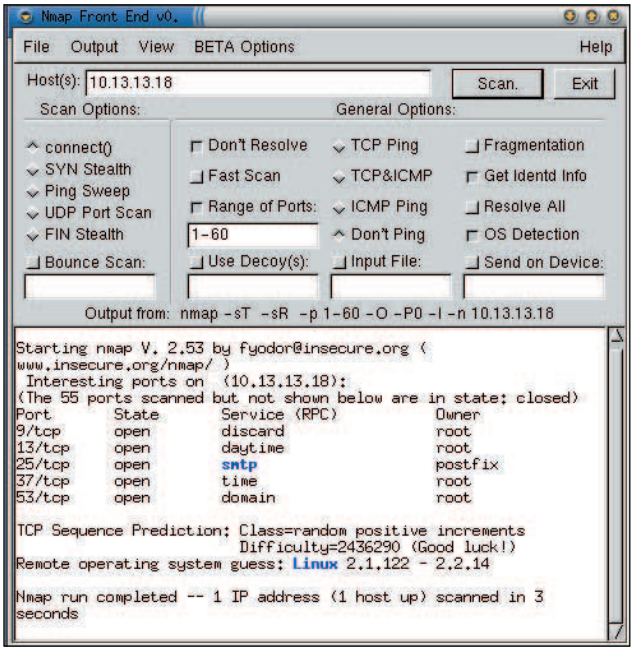
Talán hallottál már rémtörténeteket, hogy a rossz szándékú betörők milyen egyszerűen felderíthetik az áldozat rendszerének sebezhető pontjait az Internetről készen letölthető eszközök segítségével. Ezek a történetek általában sajnos igazak. Szerencsére sok ilyen program rendkívül jól használható törvényes célokra (némelyiket ilyen szándékkal tervezték), azaz felderítheted velük a rendszered fogyatékosságait.
Ez a cikk és a következő hónapban megjelenő folytatása bemutatja, hogyan használhatja az átlagos rendszergazda és a képzett biztonsági szakember az nmapot és a nessust, ezt a két kiemelkedő szabad forrású programot a rendszer biztonságának fejlesztésére. De ne feledjük, a tudás hatalom, rajtunk áll, hogy felelősséggel használjuk (és olyan módon, hogy ne kényszerítsük arra az egyenruhásokat, hogy elkobozzák kedvenc linuxos gépeinket)!

Miért pásztázunk mi, a jó fiúk?

Miért pásztázzunk? A betörő azért pásztáz, hogy megállapítsa, milyen szolgáltatások futnak a célba vett rendszeren, és milyen jól ismert támadási felületei vannak azoknak. A rendszergazda tulajdonképpen ugyanezért pásztáz, de az ő célja a rendszer hibáinak kijavítása (vagy legalább megértése), és természetesen nem akar betörni saját gépére.
A jó fiúk furcsállhatják, hogy ugyanazokat az eszközöket kell használniuk, mint a rossz fiúknak, akiket meg akarnak fékezni. Végül is nem úgy szoktuk az ajtózárunkat ellenőrizni, hogy a saját ajtónkat rugdossuk. A rendszerbiztonság azonban sokkal bonyolultabb, mint a fizikai biztonság. Közel sem olyan egyszerű felmérni a hálózatba kapcsolt számítógéprendszer viszonylagos biztonságát, mint a ház ajtajáét. Eszményi esetben mindig az összes általunk üzemeltetett hálózatba kapcsolt gépről pontosan tudjuk, hogy milyen hálózati folyamatok futnak rajta, gyakorlatilag viszont nehéz nyomon követni az adatokat a mai behálózott világban.
Ezért minden biztonságért aggódó szakembernek kötelező komolyan vennie minden eszközt, annak ellenére, hogy ezeket vetik be a rossz fiúk is. Ezekkel a programokkal nem érhetünk el sem teljes körű, sem tökéletes védelmet, de ez természetes is, ha olyan állandóan változó célponttal van dolgunk, mint a rendszerbiztonság. Egyetlen ilyen program, eljárás vagy eszköz sem szavatolhatja a teljes biztonságot. Ismételjük fennhangon: a biztonság folyamat, nem termék!
Az elmondottakon kívül még egy oka van, amiért tanulmányozzuk a biztonsági pásztázókat: a hírnév. Jó érzés s00p3r 3L33T HaX0rnek kiadni magunkat. Felülmúlhatatlan érzés, amikor az nmap, a nessus vagy valami hasonló program használata közben ártatlan arccal mondhatjuk: dolgozunk.

A pásztázók típusai és használatuk

Kétféle rendszerpásztázó ismeretes. A kapupásztázók nyitott TCP- és UDP-kapukat keresnek, amelyek mögött szolgáltatások figyelnek. A biztonsági pásztázók még egy lépéssel továbbmennek, és a beazonosított szolgáltatásokat ellenőrzik az ismert gyenge pontok szempontjából. A hétköznapi életből vett példával úgy mondhatjuk, hogy a kapupásztázó megszámolja egy ház ablakait és ajtóit, a biztonsági pásztázó mindenhova becsönget és ellenőrzi a riasztót az ablakokon. Ó, majdnem elfeledkeztünk az általános pingről, mely a harmadikfajta pásztázásnak tekinthető! Ez a hálózat egy megadott IP-címtartományát ellenőrzi végig, és a működő (azaz a pingre válaszoló) gépeket megtalálja. Ez is hasznos lehet, főleg ha pont ilyesmire van szükségünk, de terjedelmi okokból ez alkalommal csak a kapupásztázókat és a biztonsági pásztázókat fogjuk bemutatni. Minden itt tárgyalt eset igaz akár öt, akár 65 500 gépet pásztázunk.



Jellegzetes nmapfe munkafolyamat

Az nmap, a kapupásztázás világbajnoka

A kapupásztázás elve egyszerű: ha kapcsolódni próbálunk egy kapuhoz, megállapíthatjuk, hogy zárva van, vagy egy alkalmazás (pl.: webkiszolgáló, FTP démon stb.) fogadja a kapcsolatot. Könnyű írni olyan egyszerű kapupásztázót, amely a helyi connect() rendszerhívást használva csatlakozik különféle TCP-kapukhoz. A megfelelő modulok segítségével még Perlben is lehet ilyet készíteni. Csakhogy ez a módszer a legerőszakosabb és legnyilvánvalóbb módja a pásztázásnak, az eredmény nagyszámú naplóbejegyzés lesz a célrendszeren. Itt kerül a képbe *Fjodor* nmap programja. Az nmap tud egyszerű connect() pásztázást is, de az igazi erőssége a „lopakodó pásztázás”. A lopakodó pásztázás hamis TCP-csomagok használatával történik, amelyek anélkül késztetik válaszra a célpontot, hogy a TCP-kapcsolat létrejött volna.
Az nmap nem egy, hanem négy lopakodó módszert ismer azonkívül, hogy ismeri a TCP-kapcsolat pásztázását, az UDP-pásztázást, az RPC-pásztázást, az általános ping, és még az operációs rendszer „ujjlenyomatát” is le tudja venni. Van még csomó más tulajdonsága is, amelyeket inkább a sötét oldal tud kihasználni, például az FTP visszhang pásztázása, ACK-pásztázás és Window tűzfalpásztázás, viszont ezek aligha érdeklik a Linuxvilág törvénytisztelő, erkölcsös

olvasóit. Jelenleg az nmap messze a legnagyobb tudású és legsokoldalúbb kapupásztázó.

Foglaljuk össze az nmap által ismert legfontosabb pásztázási típusokat:

TCP-kapcsolat pásztázása – az OS beépített connect() rendszerhívását használja. A kiszemelt kapun teljes, háromlépcsős TCP-kézfogást kísérel meg (SYN, ACK-SYN, ACK). A sikertelen kapcsolatot az jelzi, ha a kiszolgáló a SYN csomagra ACK-RST-csomagot küld vissza, ez azt jelenti, hogy a kapu zárva van. Ez a módszer nem igényel rendszergazdai jogokat, és az egyik leggyorsabb is. Azonban ne lepődjünk meg azon, hogy a legtöbb kiszolgálóalkalmazás naplózza a megnyitás után azonnal bezárt kapukat, ezért ez elég „zajos" pásztázástípus.

TCP-SYN pásztázás – kétharmada a TCP-kapcsolat pásztázásának. Ha a célpont ACK-SYN csomagot küld vissza, az nmap rögtön RST-csomaggal felel, ahelyett, hogy felépítené a kézfogást ACK-csomaggal. Az ilyen „félig nyitott" kapcsolatokat sokkal kevésbé naplózza a rendszer, ezért a SYN pásztázás sokkal kevésbé észrevehető, mint a TCP-kapcsolat pásztázása. Ennek az az ára, hogy rendszergazdaként kell futtatni az nmapot az üzemmód eléréséhez, mivel ilyenkor nem a rendszermag állítja össze a csomagokat.

TCP-FIN-pásztázás – még csak nem is tesz úgy az nmap, mint ami szabványos TCP-kapcsolatot kezdeményez. A kiszemelt célpontnak elküld egy FIN (befejezés) csomagot. Ha a célpont TCP-, illetve IP-verme megfelel az RFC-793 előírásainak (MS-bármi, HP-UX, IRIX, MVS és Cisco IOS nem), akkor a nyitott kapuk eldobják a csomagot, a zárt kapuk pedig RST-t küldenek.

TCP NULL-pásztázás – hasonló a FIN-pásztázáshoz, de itt jelöletlen TCP-csomag megy ki (nullcsomag). Szintén az RFC-793 megfelelőséget használja ki.

TCP-karácsonyfa pásztázás – hasonló a FIN-pásztázáshoz, de itt a kimenő csomag FIN, PSH és URG jelölést kap (befejezés, adattolás és sürgős). Szintén függ a fent említett RFC-793 megfelelőségtől.

UDP-pásztázás – az UDP-kapcsolat nélküli protokoll, azaz semmilyen függőség nincs meghatározva a protokollban az egyes csomagok között semelyik irányban. Ezért az UDP-nél nem játszhatunk a TCP-nél megismert kézfogással. A legtöbb OS TCP-, illetve IP-verme azonban „elérhetetlen kapu" ICMP-csomagot küld vissza, ha UDP-csomag zárt UDP-kaput vesz célba. Ebből következik, hogy az a kapu, amely nem küld vissza ICMP-csomagot, feltehetőleg nyitva van. Mivel sem a próbacsomag, sem a válaszként érkező ICMP-csomag nem ér biztosan célba (ne feledjük, az UDP-kapcsolat nélküli protokoll, és az ICMP is az), az nmap általában több UDP-csomagot küld a kiszemelt UDP-kapura, hogy a hamis eredményeket kiszűrje. Tapasztalatunk szerint az nmap UDP-pásztázásának pontossága különböző a célba vett operációs rendszerek függvényében, de a semminél jobb.

RPC-pásztázás – egyéb pásztázási típusokkal együtt használva, ezzel deríthető fel, hogy mely kapuk vannak nyitva RPC (távoli eljáráshívás) céljára, mik a kapuk mögött figyelő szolgáltatások és mi a változatszámuk.

Általános ping (söprés) – lásd fenn a pásztázók típusai és használatuk részt.

Huhh! Szép kis lista, pedig kihagytuk az ACK-pásztázást és a Window pásztázást (ha érdekel, nézd meg az nmap(1) súgóoldalt). Az nmapnak van még egy nagyon hasznos tevékenysége, az OS ujjlenyomatának levétele. Az nmap a célpont a különböző csomagok adott válaszainak elemzéséből elég jól meg tudja határozni a célponton futó operációs rendszer típusát.

Az nmap beszerzése és telepítése

Manapság olyan népszerű az nmap, hogy a legtöbb Linux-változat tartalmazza. A RedHat 7.0 és a Debian 2.2, ezeket jelenleg is használom, tartalmazzák az nmapot (az Applications/System, illetve az Extra/Net alá besorolva). A Linux-felhasználók ezért a legegyszerűbben rendszerük csomagkezelőjével (pl.: RPM, dselect, YAST) telepíthetik az nmapot a nekik tetsző forrásból (CD-ROM, FTP stb.). Ha azonban a legfrissebb változatra, illetve a forráskódra vágyunk, mindkettőt letölthetjük a ➲ http://www.insecure.org/ webhelyről (Fjodor honlapja) RPM és TGZ formátumban. Ha a forrásból önállóan szeretnénk lefordítani az nmapot, adjuk ki a következő parancsokat (természetesen a forráskódot tartalmazó könyvtár neve más is lehet, mert az nmap 2.53 esetleg már elavul, mire e cikk eljut az olvasókhoz):

1. lista Az „alapértelmezett" nmap pásztázás

```
[root@sprecher /root]# nmap 10.123.123.9

Starting nmap V. 2.53 by fyodor@insecure.org
(www.insecure.org/nmap/)
Interesting ports on (10.123.123.9):
(The 1520 ports scanned but not shown below
are in state: closed)
Port        State        Service
80/tcp      open         http
139/tcp     open         netbios-ssn
1432/tcp    open         blueberry-lm

Nmap run completed-1 IP address (1 host up)
scanned
in 2 seconds
```

```
cd nmap-2.53
./configure
make
make install
```

Az nmap használata

Az nmap két különböző módon futtatható. A leghatékonyabb és legrugalmasabb mód a parancssor. Emellett nmapfe néven létezik grafikus felület is, mellyel összerakhatók az nmap kapcsolói, és a program futtatható (lásd az *1. ábrát*).

A GUI hasznos, ha csak gyorsan és felületesen akarunk pásztázni, vagy az nmap parancssori kapcsolóival ismerkedünk. Azonban nagyon ajánlott rendesen megtanulni az nmap parancssoros használatát, mert a GUI csupán a lehetőségek egy részét teszi elérhetővé. Arról nem is beszélve, milyen pazarló dolog X-et indítani a jó öreg nmap kedvéért… Az nmap parancsokat könnyű megtanulni. Az egyszerű pásztázás így érhető el: `nmap -s` *pásztázás típusa* `-p` *kaputartomány cél*.

A `-s` kapcsolót azonnal követi a következők egyike:

- T: TCP-kapcsolat pásztázása
- S: TCP-SYN-pásztázás
- F: TCP-FIN-pásztázás
- N: TCP-NULL-pásztázás
- X: TCP-karácsonyfa pásztázás
- U: UDP-pásztázás (a fentiekkel együtt is használható)
- R: RPC-pásztázás (a fentiekkel együtt is használható).

A `-s` kapcsoló adja meg, hogy milyen típusú pásztázást kell futtatni. Ezt követően megadhatjuk bármelyik TCP-pásztázási típust, U-t az UDP-pásztázáshoz, R-t az RPC-pásztázáshoz, illetve azonosításhoz, vagy ezek tetszőleges kombinációját. Az egyetlen megkötés, hogy egyszerre csak egyféle TCP-pásztázási típus adható meg. Ha a `-s` kapcsolót elhagyjuk, az alapértelmezés a TCP-kapcsolat pásztázása. Például a `-sSUR` arra utasítja az nmapot, hogy a célpontra végezzen el SYN pásztázást, majd UDP-pásztázást, végül RPC-pásztázást,

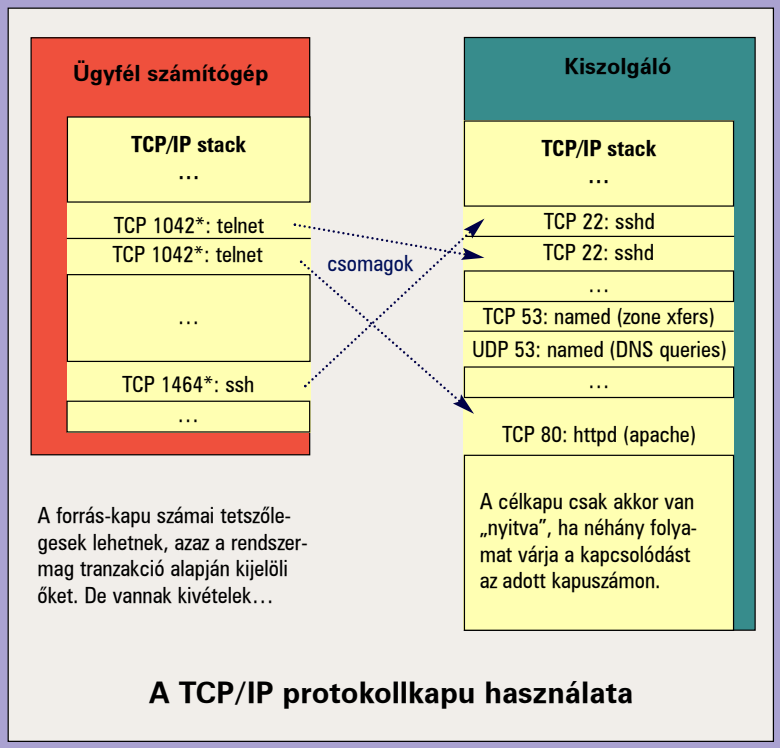
illetve azonosítást. A `-sTSR` nem működne, mert a TCP-kapcsolat pásztázása és a SYN pásztázás egyaránt TCP-pásztázások.
Ha megadod a kaputartományt a `-p` kapcsolóval, a vesszők és kötőjelek használatával pontosan szabályozhatod a pásztázandó kapuk csoportját. Például a `-p 20-23,80,53,600-1024` hatására az nmap 20-tól 23-ig és 600-tól 1024-ig pásztáz, valamint pásztázza a 8-as és 53-as kapukat. A kaputartományok felsorolásában nem lehet szóköz.
Hasonlóképpen a célpont kifejezése is lehet gépnév, gép IP-címe, hálózat IP-címe vagy IP-címtartomány. Például a 192.168.17.* mind a 255 IP-címet magában foglalja (valójában írhatnánk 192.168.17.0/24-et is); a 10.13.[1,2,4].* jelentése 10.13.1.0/24, 10.13.2.0/24 és 10.13.4.0/24. Láthatjuk, hogy az nmap nagyon rugalmas, sokféle célpontkifejezést megért.

Az életből ellesett példák – néhány egyszerű pásztázás

Mielőtt továbblépnénk, vizsgáljunk meg néhány egyszerű pásztázást, amelyek az eddig ismertetett kapcsolókat használják! Az ebben a fejezetben megadott példák az nmap 2.53 változatával (az írás pillanatában ez a legújabb) futnak RedHat 7.0-n. A célrendszer a példákban Windows 98-at futtat Samba kiszolgálóval.
Tegyük fel, hogy először „alapértelmezett” pásztázást szeretnénk végrehajtani az nmap programmal. Nem kell megadnunk kapcsolóját, ha nem szeretnénk. Ha csak a célpont IP-címét vagy egy IP kifejezést adunk meg, az nmap minden célgépet megpingel, azután végigpásztázza TPC-kapcsolat pásztázása üzemmódban az 1–1024 kaputartományt és a /usr/share/nmap/nmap-services fájlban (a fájl elérési útja különbözhet) felsorolt kapukat, összesen 1523-at. Az 1. lista megmutatja, hogyan néz ki egy ilyen alapértelmezett pásztázás, ha Windows 98 a célpont.
Csak két másodpercet vett igénybe az 1523 kapu lekérdezése, tehát nem a levegőbe beszéltünk, amikor gyorsnak neveztük a TCP-kapcsolat pásztázását.
A következő pásztázási példában vegyük az UDP-t is az eddigiek mellé, és ha már ott vagyunk, nézzük meg, hogy vannak-e RPC-alkalmazásokat futtató nyitott kapuk! Mivel az UDP-pásztázást nem a TCP-kapcsolat pásztázása helyett szeretnénk, hanem vele együtt, ezért most már azt is ki kell írni a parancssorban. A kiadandó parancs a *2. listában* látható.

Mi az a kapu?

A TCP/IP-verem (a rendszermag TCP/IP támogatását megvalósító protokollmeghajtók halmaza) az egyes hálózati alkalmazásokat és munkafolyamatokat a kapuszámok alapján különbözteti meg. Minden egyes alkalmazás/démon/folyamat, amelynek TCP-n vagy UDP-n keresztül kell kapcsolatokat kezelnie, rendelkezik kapuszámmal. A kapuszámok 0-tól 65.536-ig terjedhetnek, és két sorozat van, egy a TCP-hez és egy az UDP-hez.
Más szavakkal, a 2000. TCP-kaput használhatja egy folyamat, és a 2000. UDP-kaput egy másik, feltéve, hogy az első TCP protokollt használ, a második UDP-t. A 0-tól 1023-ig terjedő tartományt csak a rendszergazdai jogokkal felruházott folyamatok használhatják, ezért ezeket rendszerkapuknak vagy kivételezett kapuknak hívják. Az Internet Assigned Numbers Authority (IANA) fenntart egy listát a hivatalos kapuszám-hozzárendelésekről: például a telnetd többé-kevésbé mindenütt a 23-as TCP-kaput használja, mert az IANA szerint azt kell használnia. Akárcsak az Internet sok más vonatkozásában, ebben az esetben is inkább hagyományról van szó, mint nemzetközi szabványról. Senki sem akadályozhat meg egy programozót abban, hogy írjon egy olyan alkalmazást, amely nem telnet és mégis a 23-as TCP-kaput használja. Mégis, ha pásztázunk egy gépet, és a 23-as TCP-kapu működik, nagy valószínűséggel a Telnet démon figyel mögötte.
Az 1024-től 49151-ig terjedő tartományt a rendszergazdai jogokkal nem rendelkező folyamatok használják, és az IANA szintén bejegyezte ezeket (a „közösség kényelme érdekében”). Ezeket a kapukat bejegyzett kapuknak hívják. A 49152-től 65535-ig terjedő tartomány használható olyan folyamatok számára, amelyek nem kapcsolódnak az Internethez, ezért ezeket a kapukat dinamikus vagy magánkapuknak hívják.
Még egy fogalommal meg kell ismerkednünk a TCP-, illetve UDP-kapuk kérdéskörében, ez a forrás-, és célkapuk közötti különbség. Minden TCP- és UDP-csomag rendelkezik célkapuval (erre a kapura megy az elküldött csomag) és forráskapuval (innen küldték a csomagot, és a távoli szolgáltatás vagy démon ide küldi a válaszát).
A célkapukkal ellentétben a forráskapukat általában a rendszermag dinamikusan osztja ki, az alkalmazástól függően, minden egyes átvitelre külön.
A kapuszámok hozzárendelése a szolgáltatásokhoz a rendszer /etc/services fájljában csak a helyi figyelő-, illetve célkapukra vonatkozik (pl.: telnetd), és nem a helyi kifelé irányuló folyamatok forráskapuira (pl.: telnet).
Bizonyos alkalmazások, mint az NFS és a NIS, a célkapukra is dinamikus hozzárendelést alkalmaznak, de ez teljesen más téma.



A TCP/IP protokollkapu használata

2. lista nmap pásztázás TCP kapcsolatra, UDP-re és RPC-re

```
[root@sprecher /etc]# nmap -sTUR 10.123.123.9

Starting nmap V. 2.53 by fyodor@insecure.org
(www.insecure.org/nmap/)
Interesting ports on  (10.123.123.9):
(The 3075 ports scanned but not shown below
are in state: closed)

Port       State       Service (RPC)
80/tcp     open        http
111/udp    open        sunrpc (rpcbind V2)
137/udp    open        netbios-ns
138/udp    open        netbios-dgm
139/tcp    open        netbios-ssn
1026/udp   open        (rpcbind V2)
1432/tcp   open        blueberry-lm

Nmap run completed-1 IP address (1 host up)
scanned in 14 seconds
```

3. lista Kijelölt TCP- és UDP-kapuk pásztázása

```
[root@sprecher /root]#   nmap -sTU -p
1-1024,12345,12346,31336 10.123.123.9

Starting nmap V. 2.53 by fyodor@insecure.org
(www.insecure.org/nmap/)
Interesting ports on  (10.123.123.9):
(The 2049 ports scanned but not shown below
are in state: closed)

Port       State       Service
80/tcp     open        http
111/udp    open        sunrpc
137/udp    open        netbios-ns
138/udp    open        netbios-dgm
139/tcp    open        netbios-ssn

Nmap run completed-1 IP address (1 host up)
scanned in 7 seconds
```

A `-sU` és a `-sR` pásztázás (együtt `-sTUR`) különösen jól együttműködnek: az RCP sokat használja az UDP-t. Ha az nmap RPC-szolgáltatást talál egy nyitott kapun, kiírja zárójelben az alkalmazás nevét és változatszámát is (ha ki tudja találni).

Tegyük fel, hogy kicsit szűkíteni szeretnénk a pásztázás tartományát. Ez akkor lehetséges, ha sejtjük, hogy mit futtat a célgép, illetve a pásztázási időt szeretnénk lerövidíteni. Megadhatjuk a használandó kapukat a `-p` kapcsoló utáni felsorolásban. Vesszők és kötőjelek használhatók ebben a listában, de szóköz nem. A *3. lista* olyan pásztázást mutat be, amelyben minden kivételezett kaput ellenőrzünk, valamint néhány veszélyes kaput, a TCP 12345-öt és 12346-ot (NetBus alapértelmezett kapui) és az UDP 31337-et (BackOrifice alapértelmezett kapuja).

Végül lássuk, hogyan pásztázhatunk egyszerre több gépet! Könnyen. A gépeket még rugalmasabban adhatjuk meg az nmapnak, mint

4. lista Néhány lehetőség középhaladóknak

```
[root@sprecher]# nmap -sTUR -OIF -oN lamer.txt
  1.12.123.4

Starting nmap V. 2.53 by fyodor@insecure.org
(www.insecure.org/nmap/)
Interesting ports on bookoosvr (1.12.123.4):
(The 2153 ports scanned but not shown below
are in state: closed)
Port       State   Service (RPC)          Owner
22/tcp     open    ssh                    root
23/tcp     open    telnet                 root
25/tcp     open    smtp                   root
80/tcp     open    http                   apache
111/tcp    open    sunrpc (rpcbind V2)    rpc
111/udp    open    sunrpc (rpcbind V2)
113/tcp    open    auth                   nobody
443/tcp    open    https                  apache
515/tcp    open    printer                lp
587/tcp    open    submission             root
999/udp    open    applix
1024/tcp   open    kdm (status V1)        rpcuser
1024/udp   open    (RPC (Unknown Prog #))
1025/udp   open    blackjack (status V1)
1241/tcp   open    msg                    root
3001/tcp   open    nessusd                root
6000/tcp   open    X11                    root

TCP Sequence Prediction:
         Class=random positive increments
         Difficulty=613547 (Good luck!)
Remote operating system guess:
         Linux 2.1.122 - 2.2.14

Nmap run completed-1 IP address (1 host up)
scanned in 959 seconds
```

a kapukat: használhatók helyettesítő karakterek, szögletes zárójelek (listák) és „per/alhálózatbites” jelölés. Így nézne ki a 3. listán megadott pásztázás az egész hálózatra (254 cím, az eredményt most elhagytuk):

```
nmap -sTU -p 1-1024,12345,12346,31336
       ➥10.13.13.0/24
```

nmap középhaladóknak

Az nmap rengeteg ijesztő képességgel rendelkezik, átoson tűzfalakon, vigyáz, hogy ne hozza működésbe a behatolást figyelő programokat, és más módokon is segíti a felhasználót, hogy elkerülje a lebukást. Nem érzek késztetést arra, hogy ezeket a dolgokat itt tárgyaljam, bár kétségkívül ezeket is lehet törvényes célokra használni. A cikk hátralevő részét inkább olyan dolgoknak szentelem, amelyek nem függenek össze ilyen egyértelműen a betöréssel.

Tegyük fel, hogy nagy hálózat rendszergazdái vagyunk, és valaki telepít egy kiszolgálót a géptermünkben, amely úgy tűnik, elérhető az Internetről is. Ez sértheti a szervezet biztonsági szabályzatát (vagy csak a mi önérzetünket, mert nem kértek tőlünk engedélyt a telepítésre). Mielőtt dühödten érvényre juttatnánk jogainkat, előbb kicsit szimatoljunk körül, milyen veszélyeknek van kitéve a hálózatunk! Szerencsére a titokzatos kiszolgálóra valaki felírta lila krétával az IP-címét. Az is szerencse, hogy nmap-tudásunkkal felvértezve

a bosszú igazságos angyalaként láthatunk munkához. Íme néhány lehetőség az nmap használatára ebben a helyzetben.
Először is, milyen operációs rendszer fut a kiszolgálón? Az OS ujjlenyomat megmondja, ehhez a `-O` kapcsolót kell megadni. A `-O` használatakor az nmap mindenféle jelzéssel ellátott TCP-csomagokat küld, és a válaszokat összeveti az OS ujjlenyomat-adatbázissal (/usr/share/nmap/nmap-os-fingerprints a Red Hat 7.0 rendszeren). Tapasztalatunk szerint nagyon jól működik a felismerés, kivétel a MacOS 8 esetén, ez ugyanis összezavarja.
Következő kérdés, vannak-e rendszergazdai jogokkal rendelkező szolgáltatásokat futtató nyitott kapuk. Természetesen néhány szolgáltatásnak szüksége van ilyen széles körű jogosultságokra, a legtöbbnek azonban nincs. Ha a webkiszolgáló a rendszergazda nevében fut ezen a gépen, akkor valakinek ezért felelnie kell, az biztos. Használjuk a `-l` kapcsolót a célpont ident démonjának lekérdezésére, ennek egyetlen célja, hogy szétkürtölje a világban, hogy melyik felhasználó birtokában vannak az egyes szolgáltatások.
Lehetséges csökkenteni annak az esélyét, hogy erőszakos pásztázásunkkal túlterheljük a célrendszert vagy a hálózatot? Természetesen. A `-T` kapcsolóval megadható az időzítés módja. A lehetőségek: Paranoid, Sneaky, Polite, Normal, Aggressive és Insane (üldözési mániás, lopakodó, udvarias, szokásos, erőszakos és őrült). A sorrend a hálózat egyre nagyobb megterhelésének felel meg, és a háttérben az húzódik meg, hogy az nmap mennyi időt vár az egyes csomagok elküldése között, és hogy sorban vagy kötegekben küldi ki őket. A `-T Polite` jó választás, ha finoman szeretnénk bánni a célponttal, illetve a hálózattal.
Hogyan végezzünk gyorsan a pásztázással? Csak a valószínű szolgáltatásokat akarjuk ellenőrizni, és nem szeretnénk az összes kiemelt kaput végigpásztázni? A `-F` kapcsoló arra utasítja az nmapot, hogy csak az nmap-services fájlban felsorolt kapukat pásztázza. Így elkerülhető, hogy olyan kapukat is ellenőrizzünk, amelyek valószínűleg nem adnának érdekes eredményt.

Kapcsolódó címek

Fjodor hivatalos nmap oldala
➲ http://www.insecure.org/nmap/

Fjodor „The Art of Port Scanning” című cikke az nmap utasításforma tekintetében elavult, de nagyszerűen leírja, hogyan működik a kapupásztázás általában, és bemutatja a lopakodó pásztázás működési elvét
➲ http://www.insecure.org/nmap/nmap_doc.html

Fjodor szórakoztató cikke az nmap OS-ujjlenyomatfelismerő képességéről ➲ http://www.insecure.org/nmap/nmap-fingerprinting-article.html

Rik Farror cikke a Network Magazine-ból következő módon írja le az nmap OS-ujjlenyomatfelismerő képességét
➲ http://www.insecure.org/nmap/press/network_magazine-system_fingerprinting.txt

Az IANA hivatalos listája a jól ismert, bejegyzett és magán, illetve a dinamikus kapukról ➲ http://www.isi.edu/in-notes/iana/assignments/port-numbers/

A hivatalos Internet Engineering Task Force (IETF) RFC tárolóhely. Fúrja az oldalad, hogy mit jelent az RFC-793-megfelelőség? Ne töprengj tovább! Különös figyelmet érdemel az RFC 793 (TCP protokoll), az RFC 768 (UDP protokoll) és az RFC 1413 (Ident protokoll)
➲ http://www.ietf.org/rfc.html

Végül, van arra egyszerű lehetőség, hogy a bénaság bizonyítékát szövegfájlként mentsük? A `-oN fájlnév` az eredményeket szövegfájlba menti. Ha a V@gÁny DuMáT részesítjük előnyben, használhatjuk a `-oS` kapcsolót (az „S” a „Script-Kiddie-Duma” rövidítése). A *4. listán* láthatjuk, hogy az engedély nélkül telepített kiszolgáló sok más mellett fogadja a Secure Shell, a Telnet, a HTTP/SSL, az LPD, az X és a nessus kapcsolatokat. A nessus? Mi a baj vele, hiszen ez egy biztonsági pásztázó. Nem akarjuk, hogy a hálózatunkon található nessus-kiszolgáló látszódjék az Internetről – a következő hónap témája pont ez lesz.
Az nmap nagyon hatékony, de a nessus még egy lépéssel továbbvisz, és megszondázza azokat a kapukat, amelyeket az nmap talált, nincs-e ismert gyenge pontjuk. Ismét arra fogunk törekedni, hogy ezeket a hatékony eszközöket a jóra használjuk, és ne rosszra.

Mick Bauer (mick@visi.com) alkalmazott biztonsági vezető az ENRGI hálózatmérnöki és tanácsadó cég minneapolisi részlegénél. 1995 óta Linux-rajongó, és 1997 óta vakbuzgó OpenBSD-s. Különös élvezetét leli abban, hogy ezeket az élvonalbeli operációs rendszereket rávegye, hogy elavult roncsokon fussanak. Mick szívesen vesz minden kérdést és hozzászólást.

Typescript

Rögzítened kellene a terminál kimenetét, de nem mindent lehet átirányítani? Használd a `script` parancsot. A parancs kiadása után a terminálon megjelenő összes szöveget rögzíti a `typescript` nevű fájlban.

```
tux@coollinuxbox:/home/tux$ script
script: WARNING: script session is not secure
against eavesdropping/hijacking!
script: read /usr/doc/bsdutils/README.script
for details.
Script started, output file is typescript
tux@coollinuxbox:/home/tux$ python
python parancsok
Control-D
tux@coollinuxbox:/home/tux$ exit
Script done, output file is typescript
tux@coollinuxbox:/home/tux$ cat typescript
Script started on Thu Oct 12 12:03:22 2000
tux@coollinuxbox:/home/tux$ python
Python 1.5.2 (#1, Dec 15 1999, 11:15:06)  [GCC
2.7.2.3] on linux2
Copyright 1991-1995 Stichting Mathematisch
Centrum, Amsterdam
>> 45+89+12.25+63.21
209.46
>> 70/12
5
>> 70%12
10
>>
tux@coollinuxbox:/home/tux$ exit

Script done on Thu Oct 12 12:04:43 2000
tux@coollinuxbox:/home/tux$
```