Tűzfalszabályok felderítése

SZABÓ ISTVÁN
KFKI-LNX Zrt.
szabo.istvan@kfki-lnx.hu

Kulcsszavak: firewalk, tűzfal felderítés, tűzfalszabály felderítés, hálózati felderítés, portscan

Jelen publikáció a hálózati felderítés egy konkrét területére fókuszál: a tűzfalszabályok vizsgálatára és felderítésére. A szükséges elméleti háttér után bemutatjuk, hogy milyen módszerek állnak rendelkezésre a szabályrendszer megismeréséhez, részletesen elemezzük a FireWalk technikát, ismertetjük ennek a javasolt kiterjesztését, és végül a lehetséges ellenlépéseket.

1. Bevezetés

Napjainkban sajnos nehéz a számítógépes bűnözők élete. Temérdek biztonsági berendezés, tűzfal, egyre jobban képzett rendszergazdák hada keseríti meg az életüket. Egy jól képzett támadó azonban mindenre képes: elszánt, egy lépéssel a jó oldal előtt áll és gondosan előkészül az akciókra.

Sun Tzu, az emberiség történelmének egyik legnagyobb hadvezére óta tudjuk: *„Ha ismered ellenségedet és ismered önmagad, nem kell félned száz csatától sem."* A támadás tervezésének szerves része a célpont minél jobb megismerése. Egy profi hacker vagy cracker is ennek fényében jár el. Írásom célja annak bemutatása, hogyan tudjuk ezt minél jobban megakadályozni.

Legfőbb célunk az elméleti és gyakorlati ismeretek átadása mellett az, hogy felhívjuk a figyelmet a hálózati biztonság aktualitására. A crackerek, vírusok, trójai programok nagy része akár több éve ismert elvek, megoldások segítségével is elhárítható lenne. Nem lenne szabad utat engedni a céltalan rombolásnak.

2. Elméleti háttér

A szabály-felderítési módszerek elemzése előtt foglaljuk össze a lényegi részt, a FireWalk technika feldolgozásához szükséges specifikus tudnivalókat.

Szó lesz az ICMP üzenetekről, IP TTL mezőről, az ehhez kapcsolódó algoritmusokról, valamint a tűzfalak típusairól. Végül bemutatjuk a felderítés alapeszközét, a portscan-t (portpásztázás). Ez utóbbi módszer használható egy célgép nyitott, illetve zárt portjainak a feltérképezéséhez, ha nem számolunk tűzfal jelenlétével. Amennyiben a célgép tűzfallal van védve, akkor csak a nyitott portokat lehet meghatározni ezzel a technikával. A zárt portokról nem lehet eldönteni, hogy a célponton van-e zárva, vagy az őt védő tűzfal bontja a kapcsolatot.

2.1. ICMP

Az ICMP (RFC 792 vagy pl. [5]), az Internet Control Message Protocol rövidítése. Az IP-re épülve végez hibajelzési és hibakeresési feladatokat. Az ICMP üzenet IP csomagba van ágyazva, az ICMP fejléc az IP fejléc után következik.

Feladata, hogy különböző jelzéseket adjon a hálózat, illetve eszköz működéséről. Az üzenetek nagy része automatikusan keletkezik, ilyen lehet például, ha a célpont nem elérhető. Létezik olyan üzenet is, amit főleg diagnosztikára használunk, ilyen az echo-request és reply, melyet a ping programmal küldhetünk diagnosztikához.

Az ICMP üzenetek az információt két mezőben szállítják. A jelzés fő kategóriáját a típus (type) határozza meg. Ezen belül pontosít a kód, utóbbit nem minden típus esetén használják.

A FireWalk-hoz kapcsolódó ICMP típusok:

- *3-as típus –*
  destination unreachable (cél elérhetetlen).
  Ilyen üzenet több okból keletkezhet:
  ha a routing-táblában nincs meg a célhálózat,
  illetve számunkra a fontos eset az,
  amikor tűzfal szabály tiltja az adott forgalmat.
  Ritka, hogy a tűzfal ilyet küldjön,
  biztonsági szempontból nem javasolt,
  elégtelen konfiguráció esetén azonban előfordul.
- *11-es típus –*
  time-exceeded (időzítő lejárt).
  Ezt akkor küldik a routerek, amikor egy csomag
  TTL (time to live – „életkor") értéke 0-ra csökken.
  Az eldobó router így jelez a küldőnek,
  hogy nem ért célt az általa kibocsátott IP csomag.
  Fontos, hogy az ICMP üzenetet
  nem az előző router kapja meg,
  hanem a tényleges forrás (tipikusan egy PC)!

2.2. IP TTL

A TTL mező része az IP fejlécnek, 1 byte hosszú, így értéke 0-255 lehet. Az IP protokoll tervezőinek az volt a célja, hogy a route-olt hálózatban egy esetleges konfigurációs hiba esetén ne keringjenek a végtelenségig a csomagok.

Ezért implementálták a TTL mezőt, melynek értéke induláskor egy magas szám (tipikusan 128 vagy 255),

ezt az értéket minden router eggyel csökkenti. Ha 0-ra csökken, akkor a csomag eldobásra kerül. Ez a legtöbb esetben véd a hurkok ellen és az egy byte is elégnek bizonyul, hiszen még az Interneten is minden elérhető maximum 20-40 ugrással.

TTL=0 esetén a router, amelyiknél lejárt az időzítő, értesíti a küldőt arról, hogy nem ért célba a csomag és a hiba okáról is tudósít. Mindezt az információt egy ICMP time-exceeded (type 11, code 0) üzenet formájában küldi el.

2.3. Portscan

Mielőtt a támadás bekövetkezne, szükséges az egyes hostok által nyitva tartott TCP portok azonosítása, az általuk nyújtott szolgáltatások feltérképezése. Erre alkalmas módszer a portscan [3].

A támadó egymás után csatlakozik a célpont portjaihoz és megnézi, hogy ezek közül melyek vannak nyitva. Ezt teheti egyszerű TCP kapcsolat-felépítésekkel is, ennek azonban több hátránya van: lassú és biztosan felkelti a rendszeradminisztrátorok, esetleg egy IDS/IPS figyelmét.

Ezért ennél kifinomultabb módszerek születtek a portok feltérképezésére. A 80-as, 90-es évek hackerei nem kevés kreativitással egészen zseniális módszereket dolgoztak ki arra, hogyan lehetne ezt a célt kevés erőforrással, minél nehezebben detektálható módon végrehajtani.

A lehetőségek száma szinte végtelen, jelen cikknek nem is célja, hogy sorra végignézzük az összes módszert. A lényeg megértéséhez azonban hasznos lehet egy áttekintés a legismertebb, legeredményesebb módszerekről.

Ahogy azt már említettük, a portscan ugyan kiválóan alkalmas a nyitott portok felderítéséhez, de nem képes különbséget tenni a zárt és szűrt portok között, ha a forgalom tűzfalon is átmegy! Erre csak a FireWalk módszer alkalmas.

- *TCP egyszerű scan:* sima TCP kapcsolatfelvétel, azaz a három utas TCP kézfogás használata. Egyszerűen implementálható, de nagyon primitív módszer.
- *TCP SYN scan:* a kapcsolat létesítés harmadik lépését kihagyja. A célpont persze az első SYN-re a szabványnak megfelelően ACK, SYN-el, vagy RST-vel válaszol, attól függően, hogy a port nyitva van-e. Sok rendszer nem naplózza az ilyen portscant, mert ez nem egy érvényes kapcsolatfelvétel.
- *TCP FIN scan:* a SYN csomagokat kívülről gyakran nem engedi be a tűzfal, de a FIN-ek átcsúsznak rajta. Ekkor használható a FIN scan, itt nem SYN, hanem FIN csomagokat küld a támadó. A cél RFC szerint RST-vel válaszol, ha a port csukva van, illetve nem válaszol, ha az nyitva volt. Egyes operációs rendszerek rosszul implementálták az RFC-t és mindig RST-vel válaszolnak, így immunisak ezzel a módszerrel szemben.
- *Fragmentation (töredezés) scan:* a kapcsolódáshoz használt csomagok sok kis darabra tördelésével megnehezítjük a védelmi berendezések munkáját. Gyakran a tördelt csomagokat nem is kísérelik meg összerakni, így ezek átmehetnek a tűzfalon, illetve nem detektálódnak az IDS által. Biztonságosan konfigurált rendszerek figyelmen kívül hagyják a sok darabban érkező csomagokat és azok azonnal eldobásra kerülnek.
- *UDP scan:* mivel az UDP nem kapcsolatorientált, ezért sokkal nehezebb megállapítani, hogy mely UDP portok vannak nyitva. RFC szerint nem kell válaszolni egy zárt UDP portra történő kapcsolatkérelemre, de a legtöbb implementáció mégis megteszi. Ez lehetővé teszi, hogy a támadó azonosítson néhány biztosan zárt portot, de a többiről nem ad információt.
- *FTP proxy scan:* szabvány szerint az FTP szerverek támogatják a proxy módot, azaz meg lehet határozni, hogy melyik IP-re menjenek az adatok a szervertől a klienshez. Ez manapság inkább biztonsági hiányosságnak nevezhető, hiszen amellett, hogy felhasználható portscan-re, tetszőleges mennyiségű adatot küldhetünk akárhova az Internetre, hamisított levelet írhatunk stb. A portpásztázás úgy lehetséges, hogy célként az áldozatot jelöli meg a támadó, PORT parancscsal pedig kiválasztva minden pásztázandó portot, adatot küld. Az FTP szerver tájékoztatást nyújt arról, hogy sikerült-e az átvitel vagy nem, ebből a támadó megtudja, hogy nyitva van-e a célport.
- *Zombie scan:* ez a technika talán a legkreatívabb az összes közül. Az IP sorszám mező (nem TCP sorszám!) lehetőséget ad egy olyan portscan végrehajtására, ahol a támadó rejtve marad. Szükséges egy olyan zombi gépet találni valahol az Interneten, ami kevés forgalmat generál, és a következő biztonsági hibával rendelkezik: az IP sorszám mező minden küldött csomagnál konstans értékkel nő. Ez a sorszám, a TCP sorszámmal ellentétben nem változik kapcsolatonként, hanem egy globális érték, minden egyes elküldött csomagnál nő. A támadó folyamatosan kommunikál a zombival, és figyeli, hogy mennyivel nő ez az érték a zombitól jövő válaszokban. Közben kapcsolódni próbál a célpont egyik portjára, de az IP forrás mező értékét kicseréli a zombi címére. Ha a port nyitva van, akkor a cél egy SYN, ACK-t küld a zombinak, mire ő egy RST-vel válaszol. Ha zárva, akkor egy RST-t küld, erre a zombi nem válaszol. Látható, hogy előbbi esetben egy csomagot kibocsátott a zombi, míg utóbb nem, ebből a támadó tudhatja, hogy a port nyitva volt-e.
- *Snow blind (hóvakság) scan:* sok hamis címről indít scant a támadó, a valós IP címe mellett. Így a célpontnál gyakorlatilag lehetetlen kitalálni, hogy melyik volt az igazi.

2.4. A tűzfalak típusai

A hálózatbiztonság alapeszközének tekintett eszközöket több szempont alapján lehet csoportosítani [4,5]. Fontosabb csoportosítási szempontok:

- *Implementáció típusa szerint:* hardver-szoftver.

Előbbi esetén a tűzfal célhardveren fut, nem egyszerű PC-n. Hardveres megoldásra példa a Cisco PIX tűzfalcsalád. Szoftveres például a Linux iptables.

• *Elemzés módja szerint:* csomagszűrő, kapcsolatalapú, alkalmazásszintű és proxy.

Csomagszűrőnek nevezzük a kapcsolatokat nyilván nem tartó, egyszerű IP, TCP adatok alapján szűrést végző eszközöket. A kapcsolatalapú tűzfalak figyelik az átmenő forgalmat és ha egy kapcsolat engedélyezett a forrás irányából, akkor nem kell külön a visszairányú forgalmat is engedélyezni. Alkalmazásszintű tűzfalak a magasabb rétegek információit is elemzik, például a HTTP kérésekben az URL hosszát és tartalmát. Proxy tűzfalak nem engednek direkt kapcsolatot a védett hálózatra, hanem a tűzfal folytatja le a védett gépek helyett a kommunikációt, (mint egy web proxy) az egyes átmenő protokollokat mélyrehatóan elemzi és ha kell, megszakítja.

Tipikus esetben a négy mód közül többet is megvalósít egy jó tűzfal, például: kapcsolatalapú + alkalmazásszintű, proxy + kapcsolatalapú. Az ilyen hibrid megoldások előre meghatározott forgalmat az egyik, illetve másik módon szűrik. (Példa: egy proxy-kapcsolatalapú hibrid az ismert, proxyzásra alkalmas protokollokat proxy a többi protokollt kapcsolatalapú tűzfal módon szűri.)

• *TTL mező kezelés alapján.*

Ez a FireWalk vizsgálatakor kulcsszerepet játszik. Fontos kérdés, hogy csökkenti-e a TTL mezőt vagy nem?

3. Tűzfalszabályok felderítése

A támadó egyik fontos célja lehet a hálózati topológia megismerése mellett a szűrési szabályok felderítése.

Ez az információ több előnnyel is kecsegtet: meghatározható, milyen forgalom megy át a tűzfalon, egycsomagos támadások ilyen forgalomnál gond nélkül végrehajthatók. Másrészt ha ismert a tűzfal szabályrendszere, akkor a szűrt portok megkerülésével célszerű támadásokat indítani. (A tűzfalszabályok megkerüléséről röviden lesz még szó a továbbiakban.)

Az előző fejezetben bemutatott portscan technikák képesek meghatározni egy hálózati berendezésen futó alkalmazásokat, illetve a zárt portokat, ha nincs tűzfal a célhoz vezető úton. Amennyiben van (ez a tipikus eset), akkor csak a nyitott portokat képes meghatározni.

Tekintsük egy port scannelését, valamilyen SYN alapú módszerrel. (A FIN scan jellegű módszerek által nyerhető információ még kevesebb, lásd fentebb.)

A lehetséges válaszok a SYN csomagra:

• *SYN, ACK* – ez egyértelműen jelzi, hogy a port nyitva van.
• *RST* – két eset lehetséges: zárva van a port és a célpont válaszol, vagy tűzfalszabály tiltja a forgalmat és a tűzfal reset-tel bont.
• *Nincs válasz* – a port szűrve van, a tűzfal drop módra van konfigurálva, illetve a célponton futó operációs rendszer vagy személyes tűzfal nem válaszol.

Az első esetben nincs további teendő, megállapítható a port állapota. Az utóbbi két esetben azonban további vizsgálat szükséges.

3.1. A FireWalk technika

A FireWalk pont a fenti kérdést képes eldönteni. A tradicionális FireWalk algoritmus a következő [1]:

0) *Bemenetek:* célpont IP-je, tűzfal IP-je. (A módszer feltételezi, hogy csak egy tűzfal van a célhoz vezető úton.)
1) *Tűzfal távolságának meghatározása:* hány ugrásra (hop) van. Ez történhet egyszerű traceroute-tal. Ez a számot jelöljük N-nel.
2) *Scannelés N+1 TTL értékkel.* Ekkor (optimális esetben, ld. lejjebb) a tűzfal a következőképpen reagál:
   – A célportot átengedi a szabályrendszer. Ekkor a router továbbítani próbálja a csomagot és lejár a TTL. ICMP time-exceeded üzenetet generál a forrásnak.
   – A célport szűrve van: ekkor azonnal eldobásra kerül a csomag és nincs válasz.
3) *2. pont ismétlése* az összes portra.

A fenti módszer elvben tökéletes, a gyakorlatban viszont ritkán használható.

Optimális eset alatt a következőt értjük:

– A tűzfal csökkenti a TTL-t és a szűrés a TTL csökkentés előtt valósul meg. Az utóbbi jellemzően így történik, de a TTL csökkentés nem minden tűzfalra igaz. Példa a Cisco PIX tűzfalcsalád: ezek nem csökkentik a TTL-t. Ilyen eszközök ellen nem lehet ezt a típusú felderítést végrehajtani, hiszen annak lényegi része (TTL csökkentés) ellen immunisak.
– A tűzfal a szűrt forgalmat egyszerűen eldobja.
– A tűzfal generál ICMP üzeneteket. Ennek köszönhetően kapunk vissza a 2. pontban ICMP time-exceeded üzenetet. Ha a tűzfal nem generál ICMP-t, akkor egyik esetben sem kapunk választ, nem lehet eldönteni, hogy szűrve van-e a port.

3.2. A FireWalk továbbfejlesztése

Amennyiben ezek nem teljesülnek, nehezebb végrehajtani a támadást. Ilyenkor más szempontokat is figyelembe kell venni:

Ha RST-vel bontja a kapcsolatokat a tűzfal, akkor az algoritmus 2. lépésének egyszerű változtatásával ebben az esetben is végrehajtható a támadás:

2) *Scannelés N+1 TTL értékkel.*
   Ekkor a tűzfal a következőképpen reagál (RST szűrés esetén):
   – A célportra irányuló kapcsolatkéréseket átengedi a szabályrendszer. Ekkor a tűzfal továbbítani próbálja a csomagot és lejár a TTL. ICMP time-exceeded üzenetet generál a forrásnak.
   – A célport szűrve van: ekkor azonnal elutasításra kerül a csomag, és RST válasz érkezik.

Ez volt a két alapeset. Innentől különböző korlátozások mellett vizsgáljuk meg a módszer kivitelezhetőségét.

1) *A tűzfal nem generál ICMP-t.*
   Ilyenkor rögtön problémát okoz a szükséges N TTL érték megállapítása. Mivel nem mindig ismert a tá-

volság, vagy lépésről lépésre kell a módszert alkalmazni, vagy feltételezhetjük hogy a vizsgálandó tűzfal az első, ahonnan már nem jön ICMP válasz, esetleg a traceroute kimenetből következtethetünk. (A szolgáltatói hálózat után tipikusan van egy ügyfél oldali router, utána pedig egy tűzfal. Ez persze nem törvényszerű, de kiindulásnak jó lehet.)

RST bontás esetén akkor is el lehet dönteni, hogy a port szűrve van-e: ilyen esetben nincs válasz. Ez megkülönböztethető a másik választól, az RST-től.

DROP mód esetén azonban nem ilyen egyszerű a helyzet, egyik esetben sincs válasz: engedélyezés esetén az ICMP time-exceeded nem keletkezik, szűrés esetén egyszerű eldobás miatt nincs válasz. Ekkor a TTL további növelésével érhető el a kívánt siker: még eggyel megnövelt TTL (N+2) esetén, ha van még egy router a célpont és a tűzfal között. Ekkor szűrés esetén nincs válasz, egyébként a következő router fog ICMP time-exceeded üzenetet generálni, ami átmegy a tűzfalon, hiszen csak az ICMP generálás van tiltva, az átmenő forgalomban engedélyezett az ICMP.

Ennek az esetnek változata, amikor nincs még egy ugrás (router) a tűzfal és a cél között. Ekkor a csomag eljut a célig, és vagy itt, vagy a tűzfalon eldobásra kerül. A két esetet nem lehet megkülönböztetni.

Szintén elképzelhető, hogy ugyan van még egy ugrás, de ezen a routeren is korlátozva van az ICMP generálás. Ilyenkor szükség lehet a következő ugrás vizsgálatára.

2) *Az átmenő ICMP tiltva van.*
   Sem RST, sem DROP esetben nincs jelentősége, mivel az alapalgoritmushoz nincs szükség átmenő ICMP-re.
3) *Mind az átmenő,*
   mind a tűzfalon generált ICMP tiltva van.
   RST bontás esetén lehetséges a támadás kivitelezése, az első pontban leírtak szerint. Itt nincs szükség átmenő ICMP-re.
   DROP esetben nem lehetséges a támadás végrehajtása, hiszen nem tudunk különbséget tenni az első pontban leírt „nincs válasz”, illetve ICMP time-exceeded között.

Összegezve: ha a tűzfal RST-vel bont és csökkenti a TTL-t, akkor az ICMP szűréstől függetlenül a támadás kivitelezhető. DROP bontás esetén is végrehajtható, ha nincs semmilyen ICMP szűrés implementálva. Részleges szűrés esetén: ha a tűzfalon generált ICMP van tiltva, és még van legalább egy router a célponthoz vezető úton a tűzfaltól számítva, akkor kivitelezhető. Ha csak az átmenő ICMP van szűrve, akkor is lehetséges a támadás. (További háttérismeret a routing, switching témakörhöz: [6,7].)

3.3. Egyéb módszerek

A tűzfalak szabályrendszerének kitalálására léteznek egyéb módszerek is, azonban a FireWalk a leghatékonyabb, legtöbb esetben alkalmazható.

Következzen ezekről egy rövid lista:

• *Tördelés*

IP csomagok kis darabokra tördelésével elérhető, hogy bizonyos tűzfalak figyelmen kívül hagyják az így álcázott csomagokat. A tördelt adatokat a tűzfalnak össze kell állítania ahhoz, hogy értelmezni és szűrni tudja, ez igen erőforrásigényes művelet. Ez ellen védekezni kell, a legtöbb hálózatba be vannak engedve a tördelt csomagok. Erre manapság csak speciális esetben van szükség, célszerű tiltani illetve korlátozni a tördelést.

• *Forrásport-hamisítás*

Csak egyszerű csomagszűrők ellen hajtható végre. Azt használja ki, hogy ezen tűzfalakon engedélyezni kell a válaszforgalmat is. (Ezzel szemben a kapcsolatalapúak nyomon követik a kimenő/bejövő kapcsolatokat, és a válaszforgalmat automatikusan engedélyezik. Proxy tűzfalak ellen a beépített protokollvizsgálat miatt szintén nem működik.) Ilyen esetben ismert portok forrásként beállításával tetszőleges port elérhető a tűzfalon keresztül. Például: ha engedélyezve van két interfész között az FTP, akkor a támadó ha FTP portról scannel, a forgalmát a tűzfal (illetve az adminisztrátor által létrehozott szabályrendszer) egy – nem létező – FTP kapcsolat válaszforgalmának minősíti, és engedélyezi. Védekezésként érdemes kapcsolatalapú tűzfalat használni.

• *TCP, IP nem használt mezők, illegális értékek*

A kreatív hackerek újra és újra találnak olyan, gyakorlatban nem használt, értelmetlen fejlécbeállításokat, amik átengedésre késztetik, vagy leállítják a tűzfalat. Ezek ellen a fölösleges szolgáltatások tiltásával, korlátozásokkal lehet védekezni.

• *Támadás a tűzfal ellen*

Ritkán fordul elő, de a tűzfal operációs rendszere ellen is jöhet ki exploit (biztonsági hiányosságot kihasználó program). Így elképzelhető, hogy a támadó hozzá tud férni a tűzfal konfigurációjához, és ebből meg tudja szerezni, esetleg módosítani tudja a szabályokat. A védekezési lehetőség azonos az előző pontban írtakkal.

4. Összefoglalás

Láttuk, hogy milyen módon lehet egy tűzfal szabályrendszerét felderíteni. A fentieket összegezve megállapíthatjuk, hogy a következő általános elvek sok veszélytől védenek [2]:

– Fölösleges szolgáltatások tiltása.
– Mindent tiltani, kivéve ami engedélyezett – és nem fordítva!
– Erős szűrési szabályok konfigurálása az átmenő forgalomra.
– A lehető legkevesebb információt nyújtani a tűzfalról, illetve a védett hálózatról.
– Szoftver (operációs rendszer) gyakori frissítése.
– Lehetőleg célhardver alkalmazása.

Ezeknek az elvi, absztrakt szabályoknak az átültetése a gyakorlatba erős védelmet nyújt a FireWalk és az összes többi módszerrel szemben. Az előbbi módszer lehetséges alkalmazási területének bővítésére hívtuk fel a figyelmet és számba vettük a különböző eseteket, védelmi megfontolásokat.

Konkrétan a következőkre van szükség ahhoz, hogy FireWalk támadást ne lehessen végrehajtani a tűzfal ellen:

– A tűzfalat DROP módra kell állítani.
– Mind a tűzfalon generált, mind az átmenő ICMP forgalmat szűrni kell.
– Ezek mellett javasolt olyan tűzfal használata, amelyik nem csökkenti a TTL-t.

Az ICMP szűrés megvalósítására egyes tűzfalak képesek intelligens módon is, a kapcsolatalapú módszerhez hasonlóan engedik át az ICMP üzenetekhez tartozó legitim válaszokat. Amennyiben az ICMP teljes szűrése nem megoldható – például szükség van rá diagnosztikai vagy egyéb célból –, akkor javasolt ilyen konfiguráció használata.

Mivel a proxy tűzfalak alkalmazása rendkívül hatásos védelmet biztosít több felderítés, illetve támadás ellen, felmerül a kérdés, hogy mi értelme van akkor egyáltalán nem proxy tűzfal használatának? Nos, ez a kérdés a hálózati biztonsági viták egyik meghatározó témája volt az elmúlt 10-15 évnek, napjainkban a használt tűzfalak több mint 90%-a nem ilyen.

Íme a legfőbb okok, amik a proxy tűzfalak háttérbe szorulását okozták:

• *Teljesítmény*

Lassabbak, mint kapcsolatalapú társaik, hiszen összetettebb elemzést végeznek, és kétszer annyi kapcsolatot kell kezelniük.

• *Alkalmazhatóság*

Ugyan a proxy technika jól működik ismert protokolloknál, a nem szokványos alkalmazásokon elveszik a biztonsági többlet, rosszabb esetben az adott protokoll nem működik proxy-n keresztül. A HTTP, FTP SMTP stb. igen, de saját alkalmazások protokolljai, például egyes VPN megoldások nem alkalmasak proxy átvitelre.

• *Drágább üzemeltetési költségek*

Bonyolultabb konfiguráció és szabályrendszer, az egyes protokollok mély ismerete szükséges. Nehezen azonosítható hibák gyakrabban fordulnak elő mint a nem proxy megoldásoknál.

Még egyszerűbb tűzfal-implementációk esetében is igaz, hogy a szabályrendszert gyakran kell módosítani, átalakítani, emiatt szükséges a rendszeres karbantartás. Ekkor célszerű a nem használt szabályokat törölni, felülvizsgálni a biztonsági politikát.

Ha kész a változtatás, akkor mindenképpen javasolt a tűzfal auditálása. Ennek a folyamatnak szükséges lépése a teljes védett hálózat portscannelése, felderítési kísérlet végzése. A feladatra több program is a segítségére lehet a rendszeradminisztrátornak, talán legismertebb az nmap ingyenes portscanner. A tűzfalak auditálása egy külön területté nőtte ki magát. Részletesen elemezni és ismertetni itt nincs lehetőségünk, az Interneten számos remek írás olvasható a témában.

A fentiek figyelembe vételével nehéz órákat okozhatunk a rendszerünket éppen feltörni készülő crackernek, illetve az auditot készítő biztonságtechnikai kollégának.

A cél ugyanaz mindkét esetben: a védett hálózat legyen minél kevésbé megismerhető a külső szemlélő számára.

Irodalom

[1] David Goldsmith, Michael Schiffman: A Traceroute-Like Analysis of IP Packet Responses to Determine Gateway Access Control Lists (Az eredeti Firewalk-ot ismertető publikáció)
[2] David M. Piscitello: Firewall Best Practices – Egress Traffic Filtering
[3] www.insecure.org – Port Scanning Techniques: http://www.insecure.org/nmap/man/man-port-scanning-techniques.html
[4] Andrew G. Mason, Mark J. Newcomb: Cisco Secure Internet Security Solutions
[5] Richard A. Deal: Cisco Router Firewall Security
[6] Richard Froom, Balaji Sivasubramanian, Erum Frahim: Building Cisco Multilayer Switched Networks (BCMSN), Second Edition
[7] Catherine Paquet, Diane Teare: Building Scalable Cisco Internetworks (BSCI), Second Edition