IT biztonság laboratórium

(VIHIMB01)

Mérési segédlet és utasítás

FW - Végponti tűzfalak

A mérést kidolgozta:
Ládi Gergő

Utoljára frissült: 2023. március 22.

Tartalomjegyzék

1. A mérés célja

Ezen mérés célja, hogy a hallgatók megismerjék a végponti (más néven host-alapú) tűzfal megoldások működését és beállítási lehetőségeit, majd ezen tudásukat a gyakorlatban is kipróbálhassák életszerű példákon keresztül. A mérés során a két leggyakrabban előforduló ilyen megoldással ismerkedhettek meg, melyek a *netfilter (iptables)* és a *Windows Firewall*.

Megjegyzés: mindkét fent említett megoldásról volt szó korábban, a Hálózatbiztonság című tárgy keretein belül. A Linux rendszerek tűzfalazásával, a netfilterrel és az iptablesszel a Firewall című előadáson, míg a Windows Filtering Platformmal és a Windows tűzfallal az Understanding and Managing Windows Firewall című gyakorlaton ismerkedtünk meg. Bár ez a segédlet igyekszik a mérés elvégzéséhez legfontosabb tudásanyagot összefoglalni, nem helyettesíti a vonatkozó előadás és gyakorlat meghallgatását, elvégzését. Ha a segédlet elolvasása után úgy érzed, hogy sok a homályos folt, érdemes lehet átismételni a kapcsolódó anyagrészeket. Segítségképp ezeket megtalálod a mérési segédlet mellet a Moodle felületen.

2. Elméleti összefoglaló

A mérésen tárgyalt tűzfal megoldások úgynevezett host alapú tűzfalak (host-based firewall), azaz olyan tűzfalak, melyek elsődleges – és általában egyetlen – célja, hogy az általa védett eszköznek szóló, valamint az erről az eszközről kiküldendő forgalmat szűrje az adminisztrátorok által beállított szabályok alapján.

2.1. Linux rendszerek tűzfala

Napjainkban az összes Linux tűzfal alapját a 2001-ben megjelent, 2.4-es Linux kernel részét képező netfilter képezi. A netfilter lehetőséget biztosít arra, hogy bizonyos, alacsonyabb szintű hálózati események bekövetkeztekor (pl. *a hálózati kártya fogadott egy keretet, és most kellene valamit kezdenie ezzel az operációs rendszernek*) a különböző kernelmodulok különféle saját, előre beregisztrált függvényeket futtathassanak le. A forgalom szűrését, módosítását, egyéb feldolgozását ezek a függvények végzik.

Ezek a függvények mindig egy jól definiálható jellegű műveletet végeznek

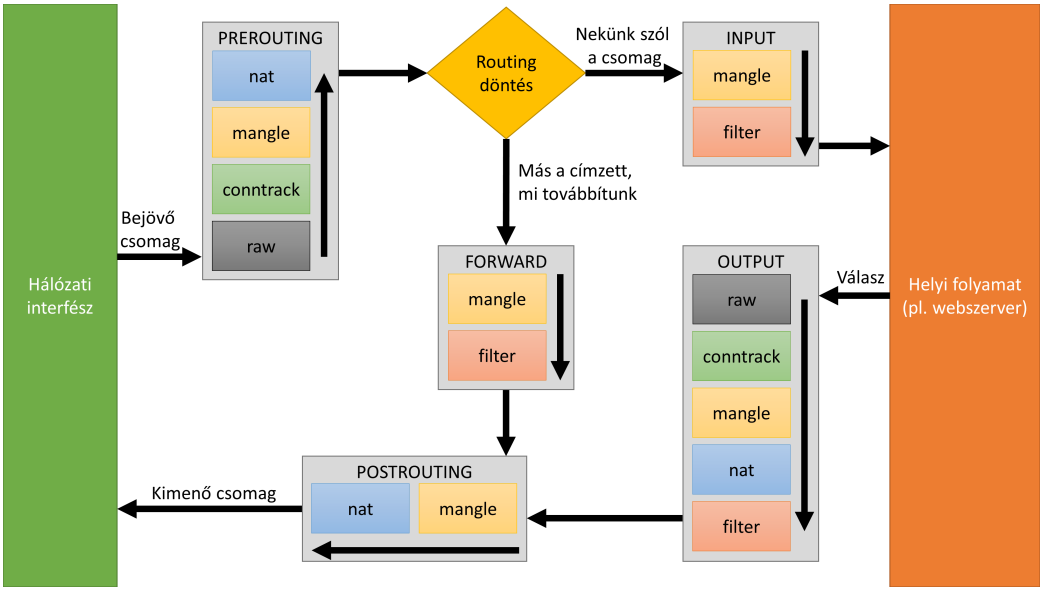
a csomagokon, a rendszerben pedig úgynevezett táblákként (*table*) jelennek meg. Ezek közül leggyakrabban az alábbi háromra van szükségünk (de nem csak ezek léteznek):

- ***filter*** – forgalomszűrésre használatos
- ***nat*** – NAT, azaz hálózati címfordítás valósítható meg vele
- ***mangle*** – a csomagok tartalma módosítható segítségével

A táblákon belül láncok (*chain*) találhatók, melyek a szabályokat (*rule*) ténylegesen tartalmazzák. A beérkező csomagok ezeken a láncokon haladnak végig sorban, a szabályoknak megfelelő módon. Láncból alapesetben öt létezik, melyek a következők:

- ***PREROUTING*** – bejövő csomag, routing döntés előtti helyzet
- ***INPUT*** – routing után vagyunk, a csomag nekünk szól
- ***FORWARD*** – routing után, a csomag másnak szól (és mi továbbítjuk)
- ***OUTPUT*** – kimenő csomag, a csomagot mi küldjük
- ***POSTROUTING*** – kimenő csomag, a csomagot vagy mi küldjük, vagy másét továbbítjuk (routerként működünk)

A táblák és láncok kapcsolata és sorrendje az 1. ábrán látható.



1. ábra. Egyszerűsített ábra a *netfilter* működéséről.

Szükség esetén létrehozhatunk saját láncokat, és egy szabállyal átugorhatunk egy ilyenre. Ez például akkor lehet hasznos, ha több különböző forgalommal ugyanazt az összetettebb műveletsort szeretnénk végrehajtani. Érdemes megfigyelni, hogy nem minden táblában létezik mindegyik lánc, például a filter táblával szűrni csak az INPUT, FORWARD és OUTPUT chainekről tudunk. Ebből következően az egyes pontokon elvégezhető műveletek sora változó. De mik is a tipikus műveletek?

- ***ACCEPT*** – csomag továbbengedése a következő láncra
- ***REJECT*** – csomag eldobása a feladó értesítése mellett
- ***DROP*** – csomag csendes eldobása
- ***LOG*** – csomag adatainak feljegyzése a rendszernaplóba és a csomag továbbengedése
- ***RETURN*** – ugrás a lánc végére (a láncon hátralévő szabályokat átugorjuk, ha vannak); felhasználó által definiált lánc esetén visszaugrunk oda, ahonnan jöttünk

Hogy pontosan milyen forgalomra illeszkedjen egy-egy szabály, számtalan szempont szerint megadhatjuk. Természetesen van lehetőség forrás/cél IP/port és protokoll szerint szűrni, kapcsolat állapota szerint szűrni (ugye a netfilter egy stateful tűzfalat valósít meg), de szűrhetünk konkrét tartalom vagy az elmúlt x időben elküldött csomagok száma alapján is. Mivel a kiértékelés módja *first match*, kiemelten fontos a szabályok sorrendje, hiszen az első találat esetén megáll az adott láncon a kiértékelés, a csomag sorsát pedig a találatban megadott művelet dönti el[1]. Amennyiben egy láncon egyetlen szabály sincs, úgy a lánc alapértelmezett szabálya, a *policy* érvényesül, ami alapesetben *ACCEPT* (illetve felhasználó által definiált láncok esetén *RETURN*).

A netfilter, a szabályok kezelésére számtalan módszer létezik. Grafikus felhasználói felülettel rendelkező rendszerek esetében találkozhatunk grafikus menedzsment megoldásokkal (pl. *firestarter* vagy *fwbuilder*), de tipikusan parancssoros eszközöket használunk (a szervereken úgyis csak ilyennel találkozni). Ezek közül a jelenlegi legnépszerűbb eszköz az *iptables*, amelynek hátránya ugyan, hogy használata relatíve bonyolult, cserében viszont a rendszer

[1]Ez alól kivételt képeznek az úgynevezett nemterminál célpontok (*non-terminating target*), melyek esetén az adott művelet végrehajtása után folytatódik a láncon a kiértékelés. Ezek közül a leggyakrabban használt a *LOG*.

telepítésekor tipikusan meg is kapjuk külön telepítés nélkül, így bízhatunk jelenlétében. Az alternatívák, még ha esetenként könnyebben is kezelhetőek, általában nem rendelkeznek ugyanazzal a tudással, és egy ismeretlen rendszeren egyáltalán nem biztos, hogy rendelkezésre fognak állni. Mindebből kiindulva, a mérés során *iptables*t fogunk használni.

Fontos: a beállított szabályok alapesetben nem mentődnek, azok a számítógép újraindításakor elvesznek. A probléma orvosolható az *iptables-persistent* csomag telepítésével, amely induláskor gondoskodik a mentett szabályok visszatöltésén. **Menteni viszont továbbra is a rendszergazdának kell**, az *iptables-save* parancs segítségével!

Érdeklődők figyelmébe ajánlom: Terjedőben van egy ígéretesnek látszó iptables-alternatíva is, az *nftables*. Hátterében ugyanúgy a *netfilter* áll, koncepciójában ugyanúgy a táblák és láncok logikáját követi. Hasonlóan nehézkesen konfigurálható, de jobb teljesítményt és néhány hasznos új funkciót ígér.

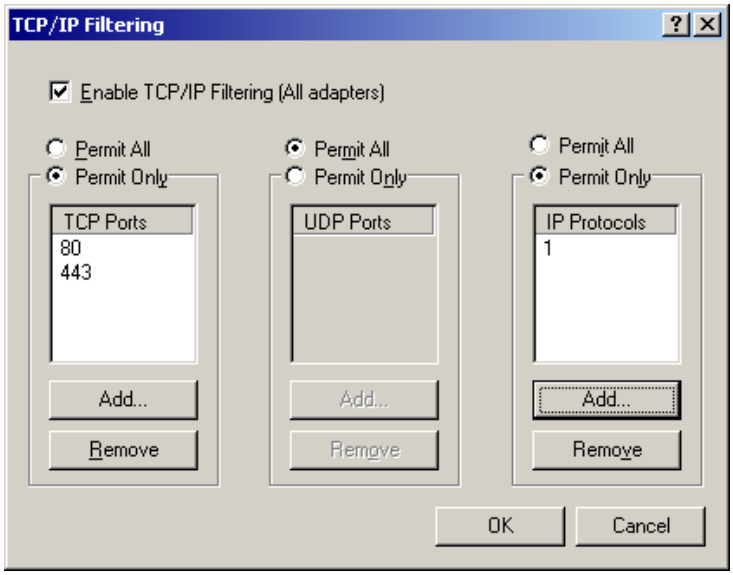
2.2. Windows rendszerek tűzfala

2.2.1. Egy kis történelem

A Windows tűzfalak múltja kevésbé hosszan nyúlik vissza az időben. A Windows 98 Millennium Edition és korábbi változatai semmilyen beépített tűzfallal nem rendelkeztek, így ha a felhasználók szerettek volna tűzfalat a számítógépükre, maguknak kellett keresniük, letölteniük és telepíteniük egyet. A fejlesztők számára nem állt rendelkezésre semmilyen API, amivel célirányosan lehetett volna tűzfalat fejleszteni, így gyakoriak voltak a programhibák és a kompatibilitási problémák is.

Az első beépített tűzfal a Windows 2000-ben jelent meg, amely nem volt nagy tudású: mindösszesen TCP és UDP portszám, valamint IP protokollazonosító alapján lehetett vele szűrni (tehát például IP-cím alapján nem), de csakis a bejövő forgalmat (ld.: 2. ábra). Ez a verzió már tartalmazott némi támogatást is a fejlesztők számára.

Az első nagyobb minőségi ugrást a Windows XP SP2 hozta 2004-ben, amely kapott egy nagyobb tudású tűzfalat és egy könnyen kezelhető grafikus felületet is. Itt már lehetőség nyílt adott alkalmazások felé irányuló forgalom



2. ábra. A Windows 2000 "tűzfala".

engedélyezésére, valamint már IP-cím szerint is lehetett szűrni (de mindezt továbbra is csak a bejövő irányban). Megjelent a szolgáltatás (*service*) fogalma, amely lehetővé tette, hogy egy szolgáltatásnév kiválasztásával egyszerre több portot is kinyissunk (például a *File and Printer Sharing* opciót kiválasztva megnyíltak a 139-es és 445-ös TCP portok, valamint a 137-es és 138-as UDP portok), anélkül, hogy tudnunk kellett volna, pontosan mely portokat kell engedélyezni. Lehetőség nyílt naplózni a csomagokat (metaadat szintjén), valamint a *netsh firewall* parancson keresztül programozhatóvá (szkriptelhetővé) vált a tűzfal, a sok rendszert kezelő rendszergazdák nagy örömére.

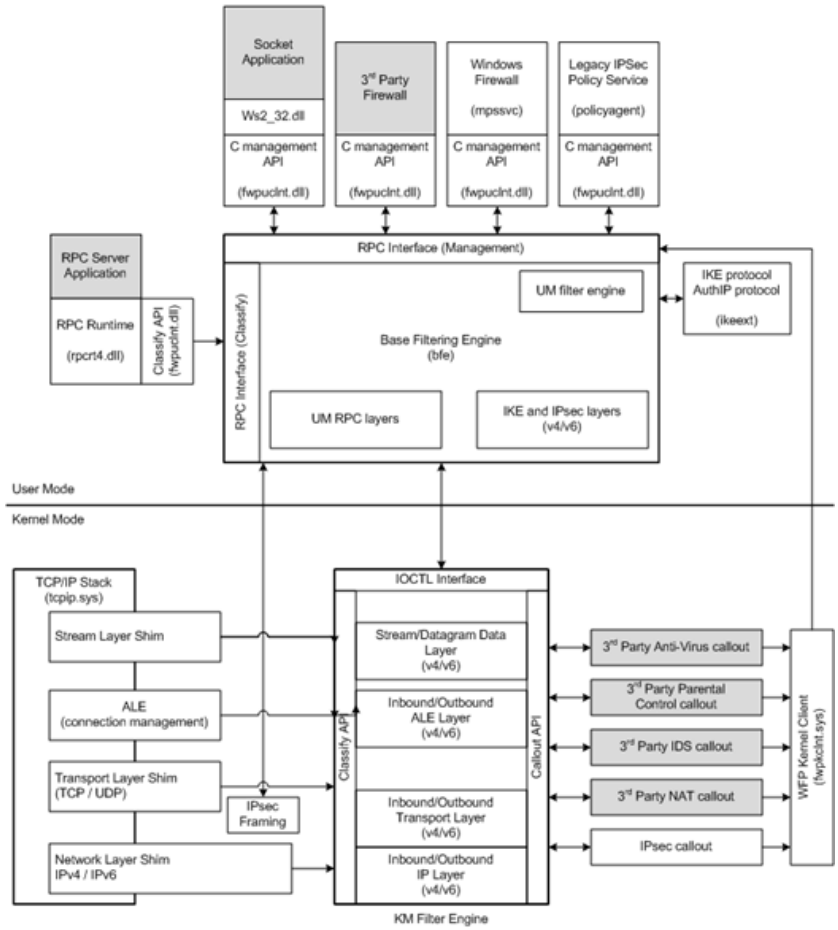
2.2.2. A Windows Filtering Platform

A Windows Filtering Platform megjelenése 2006-ban, a Windows Vista rendszerrel együtt egy új korszak kezdetét jelentette.

A WFP nem egy tűzfal, hanem egy afféle keretrendszer csomagszűrési és -módosítási feladatok támogatására. A korábbi megoldásokat teljesen újragondolták, azzal a célkitűzéssel, hogy kialakítsanak egy moduláris és fejlesztőbarát, jövőben is jól működő *(future-proof)* rendszert. Úgy tűnik, hogy ez sikerült is, hiszen a mai napig ezt használjuk.

A Windows Filtering Platform támogatja a kimenő forgalom szűrését is,

valamint nyilvántartja a kapcsolatok állapotát is, azaz stateful módon viselkedik. Felépítése (ld.: 3. ábra) némileg hasonlít a Linux netfilteréhez. Két fő komponensből áll: a *Base Filtering Engine (BFE)* és a *Kernel Mode Filter Engine*. A BFE egy user módú komponens, amely felelős többek között a modulok betöltéséért és eltávolításáért és a beállítások kezeléséért. A kernel módú komponens először az úgynevezett *shim* modulok segítségével információt gyűjt a csomagról, majd ezen információk alapján továbbengedi vagy eldobja azt, vagy odaadja a megfelelő, tipikusan harmadik féltől származó *callout* modul(ok)nak.



3. ábra. A Windows Filtering Platform felépítése.

A calloutok hasonlóan dönthetnek továbbengedés és eldobás mellett, vagy

kérhetnek további csomagokat az adatfolyamból (például víruskeresés céljából megvárnak egy teljes fájlletöltést), de léteznek olyan calloutok is, amelyek ugyan minden forgalmat elkérnek, de nem szűrést végeznek, hanem például naplózást. Előfordulhat, hogy ugyanazt a csomagot, adatfolyamot több modul is megvizsgálja, és eltérő ítéletet mondanak: van, aki azt mondja, mehet, és van, aki azt, hogy nem. Ilyenkor egy relatíve bonyolult szabályrendszer (*filter arbitration*) alapján dől el, hogy mi történjen. Üzemeltetőként erre a folyamatra nem igazán van ráhatásunk, így ennek részleteibe nem megyünk bele mélyebben. A lényeg: a tiltás erősebb mint az engedélyezés. Ebből pedig az is következik, hogy a netfilterrel ellentétben **a Windows Filtering Platform nem *first match* alapú**.

2.2.3. Windows (Defender) Firewall with Advanced Security

A Windows Filtering Platform önmagában nem használható tűzfalként, de megkapjuk mellé a Windows Firewall with Advanced Security komponenst (újabb nevén Windows Defender Firewall with Advanced Security, a továbbiakban csak Windows Firewall) is.

Ahogyan egy tűzfaltól elvárható, a Windows Firewall segítségével szűrhetünk forrás- és cél-IP, forrás- és célport, illetve protokoll szerint is, és az is megadható, hogy egy szabály csak egy adott (helyi) programra vonatkozzon. A szabályok tiltók (Block) vagy engedélyezők (Allow) lehetnek[2]. Alapértelmezés szerint befelé irányban mindent tilos, amit nem szabad (whitelisting), míg kifelé irányban mindent szabad, amit nem tilos (blacklisting). Ez a viselkedés megváltoztatható.

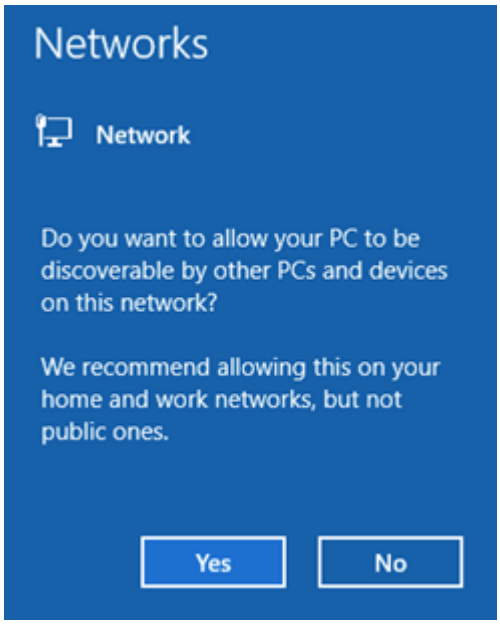
Minden szabály hozzárendelhető egy vagy több profilhoz:

- *Domain* – nagyvállalati környezet, céges hálózat Active Directoryvel
- *Private* – magánjellegű (pl. otthoni) hálózat, jellemzően csak ismert eszközökkel; biztonságosabbnak mondható egy nyilvános hálózatnál
- *Public* – nyilvános hálózat (pl. internetkávézó), többnyire ismeretlen eszközökkel; a legkevésbé biztonságosnak mondható hálózattípus

A szabályok csak azon interfészekre (kicsit félreérthető terminológiával: hálózati kapcsolatokra) érvényesek, amelyek olyan profilhoz tartoznak, mint amilyen profilokra a szabályok engedélyezve lettek. Így például ha engedélyezve van a fájlmegosztás csak a privát hálózatokra, de én most egy publikus

[2]Létezik egy *Allow connection if it is secure* beállítási lehetőség is, amely az IPSec témaköréhez kapcsolódik, de ez túlmutat a mostani labor keretein.

hálózaton vagyok az egyetemen, akkor a szabály nem lesz aktív, a fájlmegosztási szolgáltatás nem lesz elérhető kívülről. Mivel egy számítógép egyszerre több hálózatnak is tagja lehet, így egyszerre több különböző profilú szabály is aktív lehet, ekkor ezek külön-külön a megfelelő hálózatokra érvényesülnek. Hogy egy hálózat melyik kategóriába esik, azt az operációs rendszeren futó *Network Location Awareness (NlaSvc)* szolgáltatás határozza meg. Egy hálózat profilját kitalálni domain hálózatok esetén könnyű feladat, míg az egyéb esetekben nem, ilyenkor a felhasználó által, az ehhez a hálózathoz történő első csatlakozáskor megadott információk alapján dönt a rendszer (ezért is fontos a 4. ábrán látható képernyőnél mindig jól átgondolni, mit is választunk).

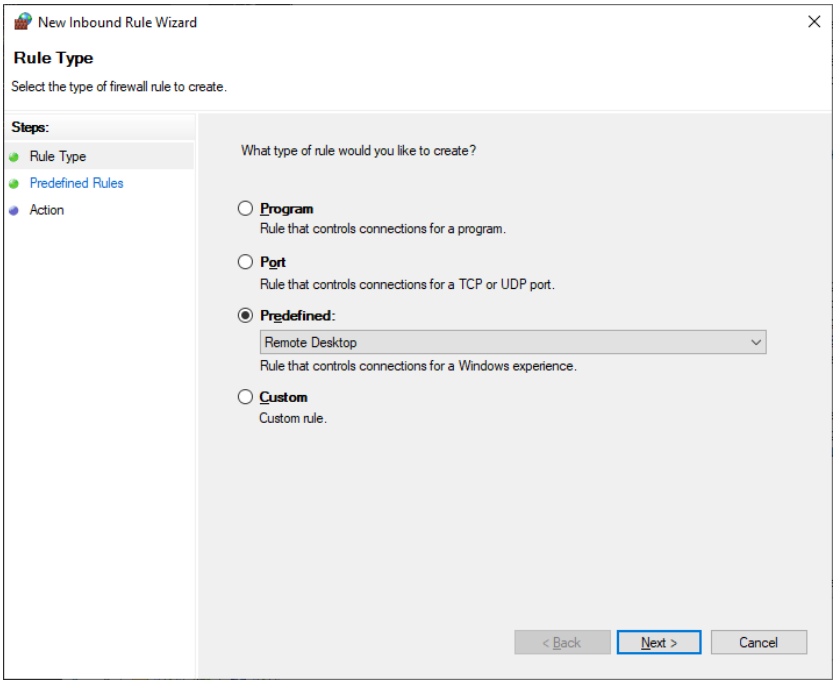


4. ábra. A felhasználó éppen választ, hogy az újonnan csatlakoztatott hálózat privát vagy publikus hálózatnak számítson.

A könnyebb felhasználás érdekében a rendszerrel együtt készen kapunk jónéhány előre definiált szabálykészletet (5. ábra). Ezek a szabálykészletek több, ugyanazon szolgáltatáshoz kapcsolódó szabályból állnak, melyek együtt vagy külön-külön aktiválhatók, a szokásos paraméterek (IP-cím, portszám, stb.) megadása nélkül. Egy friss telepítés esetén ezek egy része alapból be is van kapcsolva, így javasolt a listát átnézni és szükség esetén szűkíteni.

A Windows Firewall menedzselhető a grafikus felületen keresztül (wf.msc),

PowerShellből a *NetFirewall* parancsokkal a *NetSecurity* modulból (például: *Get-NetfirewallRule*), vagy a *netsh* parancssoros eszközzel az *advfirewall* kontextusból. Ezekkel érdemes lehet egy kicsit ismerkedni a labor előtt.



5. ábra. A rendszergazda a távoli asztalt készül engedélyezni az előre definiált szabálykészletek segítségével, a grafikus felületen.

3. Néhány lehetséges beugrókérdés

A felkészülés sikerességének önálló mérését elősegítendően néhány lehetséges beugró kérdést előre megadtam. Ha ezekre tudsz válaszolni, várhatóan nem lesznek komoly nehézségeid sem a labor elején megírandó beugrón, sem pedig később, a feladatok megoldása során. **Fontos: a lista nem teljes, azaz a beugrón előfordulhatnak itt nem szereplő kérdések is!**

3.1. netfilter (Linux)

1. Néhány mondatban foglald össze a netfilter (iptables) működését!
2. Mi az a *table*, és mi a *chain*? Mindegyikre adj 2-2 példát!
3. Mi az a *rule*, és mi a *policy*? Írj 1-1 példát!
4. Milyen műveleteket lehet végezni csomagokkal? Adj legalább 3 példát!
5. Mi a különbség a *REJECT* és a *DROP* között?
6. Adott egy Linux tűzfal, melyen több szabály is be van állítva. Beérkezik egy csomag, melyre ránézésre 5 szabályunk is illeszkedik. Mi fog történni, melyik szabály fog érvényesülni?
7. Milyen célt szolgálhat a következő parancs: *iptables -A INPUT -p tcp --dport 8080 -j ACCEPT*?
8. Írd le, milyen paranccsal tennéd elérhetővé a külvilág számára a szervereden futó *nginx*et! (Segítség: az nginx egy webszerver szoftver.)
9. Frissen telepítettem egy Linux rendszert, most léptem be rá először. Hogyan néz ki most a tűzfal?

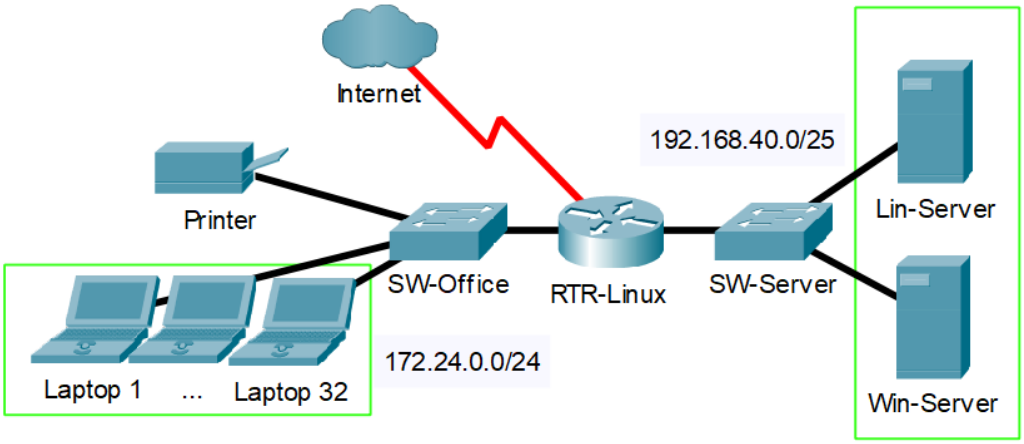
3.2. Windows Filtering Platform, Windows Firewall

1. Néhány mondatban foglald össze a Windows Filtering Platform működését!
2. Adott egy Windows tűzfal, melyen több szabály is be van állítva. Beérkezik egy csomag, melyre ránézésre 4 szabályunk is illeszkedik. Mi fog történni, melyik szabály fog érvényesülni?
3. Frissen telepítettem egy Windows rendszert, most léptem be rá először. Hogyan néz ki most a tűzfal?
4. Mi az, hogy *profile*? Mi célt szolgálnak, hányféle profil létezik?
5. Telepítettem egy programot, amely szeretne bejövő kapcsolatokat fogadni a hálózatról. Hányféleképpen tudom ezt engedélyezni a tűzfalban? Az egyik megoldást részletezd!

4. Feladatok

Az utóbbi idők nagy port kavaró hackertámadásai után a Contoso kft. (egy képzeletbeli cég) úgy gondolta, ideje nagyobb hangsúlyt fektetni a biztonságra, és megbíztak különböző más cégeket azzal, hogy elemezzék rendszereik biztonságát, tegyenek javaslatokat a fejlesztésre. Az egyik ilyen audit során kiderült, hogy van két olyan szerver, amelyen a helyi tűzfal nem megfelelően van (sőt, egyáltalán nincs is) beállítva. Ezek után felkértek téged, hogy állítsd be a tűzfalat ezen a két szerveren.

A cég informatikai infrastruktúrájának egyszerűsített váza a 6. ábrán látható.



6. ábra. A mérési elrendezés.

Az irodában 32 laptop és egy nyomtató található, melyek a 172.24.0.0/24-es tartományból kapnak IP-címet. A mérés során az egyik ilyen laptop előtt ülve fogsz dolgozni, és bizonyos feladatoknál szükség lesz a laptop IP-címére, így egy jó első lépés lehet az IP-cím megnézése és jegyzőkönyvezése.

Az eszközök egy Linux alapú routeren (*RTR-Linux*) keresztül érik el az internetet és a céges szervereket. A routernek a mérés során üzemelnie kell, viszont arra belépni, azon konfigurációt módosítani nem szükséges. A router IP-címe minden mérőpár esetén más, az eszközt bekapcsolva körülbelül 1 perc után jelenik meg vCenterben. **Fontos, hogy mindenki a legnagyobb odafigyeléssel nézze meg, jegyzőkönyvezze és használja ezt**

az IP-címet, ellenkező esetben előfordulhat, hogy több mérőpár is ugyanazon eszközökre lép be, és egymás ellen fogtok dolgozni!

Az eszközök IP-címe és fontosabb adatai a táblázatban olvashatók:

Név	**IP**	**Megjegyzés**
Laptop	ld. megjegyzés	A mérőhely számától függ. Derítsd ki és jegyzőkönyvezd!
RTR-Linux (Office)	DHCP	Mindenkinek más! Derítsd ki és jegyzőkönyvezd! Bekapcsolás után kb. 1 perccel jelenik meg vCenterben.
RTR-Linux (Server)	192.168.40.126/25	Ezt nem kell felhasználod semmihez sem.
Win-Server	192.168.40.120/25	HTTP, HTTPS kiszolgáló fut rajta. RDP-n elérhető. Felhasználónév: Administrator, jelszó: Admin1
Lin-Server	192.168.40.121/25	HTTP kiszolgáló fut rajta a 8080-as TCP porton. SSH-n elérhető. Felhasználónév: administrator, jelszó: Admin1 root user jelszava: toor

A korábban említett két szerver a *Win-Server* és a *Lin-Server*, ezeken kell a tűzfalat a leírtaknak megfelelően felkonfigurálni a mérés során. A szervereket közvetlenül is el tudod érni a laptopodról, amennyiben azon rendszergazdaként kiadod a

```
route add 192.168.40.0 mask 255.255.255.128 ROUTER_IP
```

parancsot, a *ROUTER_IP* helyére behelyettesítve a routered "feléd néző" (Office lábának) IP-címét.

Megjegyzés: Amennyiben a fenti parancs – például rendszergazdai jogok hiányában – nem működne, fallback megoldásként a két szerveren futó szolgáltatásokat a router IP-címén keresztül, NAT-olva is el lehet érni. Ebben az esetben a portszámok változatlanok, viszont mindkét szerver IP-címe helyett a router IP-címét kell használni a laptopról való csatlakozásnál és tesztelésnél. (De a feladatok megoldásánál ennek nincs jelentősége!)

Amennyiben valamelyik feladat megoldása során kizárnád (kitűzfalaznád) magadat valamely VM-ről, a vCenterbe belépve és egy webes vagy VMRC konzolt elindítva, majd a hibás beállításokat visszavonva vissza tudod magad

engedni.

Minden feladatnál dokumentáld:

- a kiinduló állapotot (ahol van értelme)
- mit csináltál (és ahol van értelme: miért azt, miért úgy?)
- mi lett az eredmény?
- az elvárt eredményt kaptad-e? (ahol van értelme)

Nem feltétlenül szükséges minden egyes lépést képekkel dokumentálni (szövegesen is megfelelő, ha elég részletes), ugyanakkor ha valahol hasznosnak találod, akár minden gondolatpontra beszúrhatsz egy képet válaszként. Fontos, hogy minden képhez írj valamilyen magyarázatot, hogy mi látható a képen; **a magyarázat nélküli, odadobott kép értéktelen, értékelhetetlen**.

4.1. A Win-Server tűzfalazása

A feladatok tetszőlegesen megoldhatók a grafikus felületről, PowerShellből, valamint a *netsh* parancs használatával is. Javaslom, hogy a tapasztalattal még nem rendelkezők használják a grafikus felületet, kivéve ahol a feladat külön azt írja, hogy mindenképp parancssorból kell megoldani.

4.1.1. A tűzfal bekapcsolása I.

A szerveren a tűzfal jelenleg ki van kapcsolva, és egyetlen szabály sincs felvéve. Szeretnéd a tűzfalat bekapcsolni, azonban ezt most még nem teheted meg, hiszen ezzel kizárnád magad. Ezért előtte fel kell venned egy tűzfalszabályt, amely:

- a méréshez használt laptopod IP-céméről
- a Remote Desktop protokolljának
- megfelelő portjára érkező forgalmat
- beengedi.

(A szabálytípusok leírásainak elolvasása után válaszd a leginkább megfelelőt.)

4.1.2. A tűzfal bekapcsolása II.

A Win-Serveren indíts egy végtelenített pinget (segítség: *-t* kapcsoló) egy külsős IP irányába (például 152.66.249.158), majd kapcsold be a tűzfalat mindhárom profilra. Ha utána pár másodperccel megszakad a távoli asztali kapcsolat, akkor az előző feladatot nem jól oldottad meg. Szerezd vissza a hozzáférést, javítsd a hibát, majd próbáld újra.

Bár a bejövő forgalom most tiltott, mégis megjönnek a válaszok a pingekre. Miért lehet ez? (Közben hagyd futni a pingelést.) Ha egyelőre nincs ötleted, folytasd a következő két feladattal.

4.1.3. Alapértelmezett viselkedés megváltoztatása

Állítsd át a tűzfalat, hogy kifelé irányban is csak azt engedje, amit kifejezetten szabad. Megszakadt az RDP kapcsolatod? Miért (nem)? Mi történik most a pingekkel?

4.1.4. Alapvető szabályok beállítása I.

Vegyél fel egy szabályt, amely kiengedi a gépről származó ICMP forgalmat. Mi történik most a pingekkel?

4.1.5. Alapvető szabályok beállítása II.

A rendszer alapvető üzemeltetéséhez és működéséhez (tanúsítványok frissítése és ellenőrzése, Windows Update futtatása, stb.) tudnunk kell DNS, NTP, HTTP és HTTPS forgalmat bonyolítani kifelé irányban az alábbiak szerint:

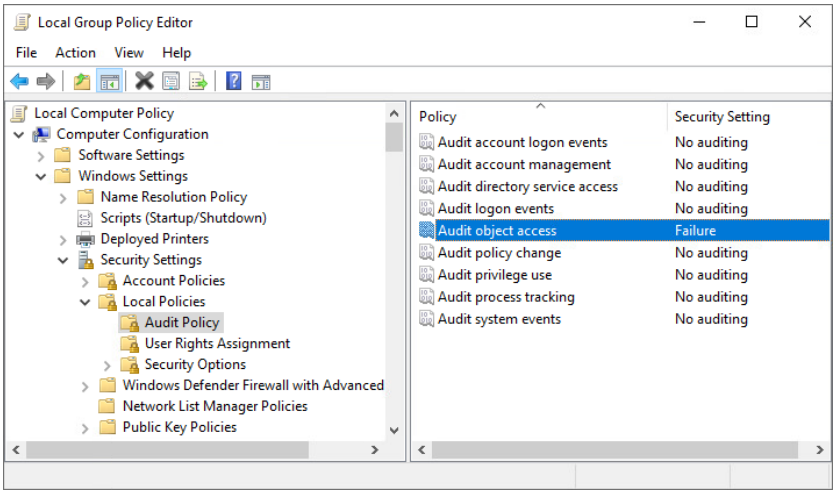
- HTTP, HTTPS: bárhova
- DNS: csak a DNS szerver felé
- NTP: csak a clk.hit.bme.hu felé

Vegyél fel egy-egy szabályt, amely ezt lehetővé teszi. Milyen IP-címeket, szállítási rétegbeli protokollokat, portokat engedélyeztél?

4.1.6. Sikertelen kapcsolatok naplózása

Szeretnéd elérni, hogy a sikertelen csatlakozási kísérletekről készüljön bejegyzés a rendszernaplóba. Ehhez először be kell kapcsolnod az események

naplózását. Indíts egy *Local Group Policy Editor*t (*gpedit.msc*), majd a 7. ábrán látható módon találd meg a *Computer Configuration* → *Windows Settings* → *Security Settings* → *Local Policies* → *Audit Policy* elemet a fában, és kattints is rá. A jobb oldalon megjelenlő listában válaszd ki az *Audit object access* szabályt, majd állítsd be, hogy a sikertelen események (*Failure*) kerüljenek naplózásra[3].



7. ábra. Objektumhozzáférések naplózásának bekapcsolása.

Ellenőrizd, hogy fut-e a webszerver. A laptopodról próbálj meg csatlakozni ehhez, például böngésző használatával. Sikerül-e? Miért (nem)?

Nézd meg a biztonsági rendszernaplót (a *Win-Server* gépen), látni-e bármit ezzel kapcsolatban.

4.1.7. Alapvető szabályok beállítása III.

Engedélyezd a bejövő HTTP forgalmat egy port alapú szabály felvételével, majd ismét próbálj meg csatlakozni a laptopodról.

[3]Az objektumhozzáférések naplózásának bekapcsolásával nemcsak a hálózati hozzáférések lesznek naplózva, hanem például a fájlokhoz való hozzáférések is. Ez most minket ne zavarjon.

4.1.8. Alapvető szabályok beállítása IIII.

Engedélyezd a bejövő HTTPS forgalmat, ezúttal az előre elkészített szabálykészletek (*Predefined rules*) segítségével.

4.1.9. Alapvető szabályok beállítása IIIII.

Kiderült, hogy egyelőre ismeretlen támadók feltörték a *172.24.0.242*-es címen található hálózati nyomtatót, majd innen próbáltak terjeszkedni a hálózaton található többi eszközhöz is. Amíg a nyomtatót alaposan nem ellenőrzi az erre a feladatra felállított csapat, akadályozd meg, hogy erről a címről bármilyen forgalmat is megkaphasson a szerver.

4.1.10. Szabályok kezelése parancssorból I.

A **jövőben** szeretnénk majd feltelepíteni és beállítani a Winlogbeat szoftvert, hogy továbbítsa a rendszernapló eseményeit a – szintén a jövőben kiépítendő – monitorozó rendszer felé. Ehhez előkészítésként fel kellene venned egy szabályt, hogy a Winlogbeat tudjon majd csatlakozni az Elasticsearch szolgáltatáshoz a *192.168.40.119*-es IP-című szerveren. (Segítség: az Elasticsearch portszáma: TCP 9200.)
A feladatot PowerShell paranccsal oldd meg!

4.1.11. Szabályok kezelése parancssorból II.

Szeretnénk majd egy Kibana dashboardot is a Winlogbeat adataiból, így el kellene érnie a Kibana szolgáltatást is. Ez szintén a *192.168.40.119*-es szerveren fog futni, az 5601-es TCP porton.
A feladatot a *netsh advfirewall* parancs segítségével oldd meg!

4.1.12. A tűzfal állapotának mentése, visszatöltése

Végeztél a tűzfal beállításával. Készíts mentést a tűzfal konfigurációjának jelenlegi állapotáról *firewall.wfw* néven.

Véletlenül töröld valamelyik kimenő szabályt, majd töltsd vissza a mentett állapotot.

4.2. A Lin-Server tűzfalazása

A feladatok megoldása során a *netfiltert* kell konfigurálnod *iptables* parancsok kiadásával.

Tipp: bár nem kötelező, ajánlom, hogy nulladik lépésként készíts egy bash scriptet, amelyet, ha lefuttatsz:

- visszaállítja minden lánc policyjét megengedőre
- törli az összes felvett tűzfalszabályt
- egymás után végrehajtja a feladatokat megoldó (*iptables*) parancsokat

Így egy elrontott parancs esetén elég a scriptet javítani és újból lefuttatni, és nem kell kézzel törölni az esetlegesen elrontott szabályokat, megszüntetni azok hatását. A feladatok megoldását sorban írd a fájlba. **Ahol fontos volna a szabályok (parancsok) sorrendje, ott olyan parancsot írj, ami a feladatok sorrendjében történő lefuttatás esetén is az elvárt működést eredményezi.**

4.2.1. A tűzfal bekapcsolása I.

A netfilter jelenleg is aktív, de mivel a policyk megengedőre vannak állítva, így minden forgalmat átenged. Szeretnéd ezt megváltoztatni, viszont ezt egyelőre (a *Win-Server* első feladataihoz hasonlóan) itt sem teheted meg, hiszen kizárnád magad.

Vegyél fel tehát egy szabályt, amely:

- a méréshez használt laptopod IP-céméről
- az SSH protokolljának
- megfelelő portjára érkező forgalmat
- beengedi.

4.2.2. A tűzfal bekapcsolása II.

A Windows tűzfalával ellentétben a netfilter nem engedi ki automatikusan a beengedett forgalomhoz tartozó, kimenő válaszüzeneteket (és fordítva, a kiengedett forgalomhoz tartozó visszajövő válaszokat sem). Így ezen a ponton még mindig nem kapcsolhatod be a tűzfalat, hiszen továbbra is kizárnád

magad.

Vegyél fel egy olyan szabályt, amely minden olyan forgalmat kienged, amely olyan kapcsolathoz tartozik, amely korábban be lett engedve.

4.2.3. A tűzfal bekapcsolása III.

Ellenőrizd, hogy valóban működnek az előző feladatban beállított szabályok. Ha mindent jól csináltál, akkor a két szabályhoz tartozó számlálók folyamatosan növekednek az SSH forgalom hatására. Ha nem ezt látod, valamit elrontottál. Javítsd a hibát, ne menj tovább, mert ki fogod magad zárni!

4.2.4. A tűzfal bekapcsolása IV.

Most már átállíthatod az alapértelmezett policyket megengedőről tiltóra. Tedd is ezt meg.

4.2.5. Alapvető szabályok beállítása I.

Próbálj csatlakozni a szerveren futó Apache-hoz a laptopról. **Az ezen a szerveren futó Apache a 8080-as porton várja a kapcsolatokat!** Sikerül? Miért (nem)?

Vegyél fel egy szabályt, amely bárhonnan beengedi a forgalmat erre a portra. Próbálj meg ismét csatlakozni az Apache-hoz. Most sikerül?

4.2.6. Alapvető szabályok beállítása II.

A rendszer megfelelő működéséhez és működtetéséhez itt is szükségünk lenne, hogy az NTP, DNS, HTTP és HTTPS forgalmak kifelé engedélyezve legyenek az alábbiak szerint:

- HTTP, HTTPS: bárhova
- DNS: csak a DNS szerver felé
- NTP: csak a clk.hit.bme.hu felé.

Vegyél fel szabályokat, amelyek mindezt lehetővé teszik.

Ügyelj arra is, hogy a válaszüzenetek is be legyenek engedve! Ez utóbbi célra elegendő egy általánosabb, minden válaszüzenetet beengedő szabályt felvenni. Ezt a szabályt célszerű a szabálylista tetejére beszúrni.

4.2.7. Alapvető szabályok beállítása III.

Linuxok esetében a loopback címek közötti kommunikáció sincs automatikusan engedélyezve, márpedig erre sok esetben szükség van.

Vegyél fel egy szabályt, amely beenged minden olyan forgalmat, ami a loopback interfészről érkezik a loopback interfészre.

4.2.8. Alapvető szabályok beállítása IV.

Később erről a szerverről is szeretnénk adatokat gyűjteni, majd a gyűjtött adatokat elküldeni az ELK stack felé. (Emlékeztetőül: az Elasticsearch a 9200-as, a Kibana az 5601-es TCP porton figyel, a szoftverek pedig a *192.168.40.119*-es IP-című szerverre lesznek telepítve.)

4.2.9. Alapvető szabályok beállítása V.

Ezen a gépen is szűrni kellene a feltört nyomtatótól érkező forgalmat. Vegyél fel egy szabályt, amely eldob minden, a nyomtatótól (*172.24.0.242*) érkező csomagot. (Megoldásodat gondold át!)

4.2.10. Naplózás

Vegyél fel egy szabályt, melynek hatására az összes bejövő, az Apache HTTP szerver felé irányuló kapcsolódási kísérletről *syslog* bejegyzés keletkezzen.

Töltsd be újra a laptopról a *Lin-Server* főoldalát. Mit látni a rendszernaplóban?

4.2.11. Haladó konfiguráció I.

Kezdd el pingelni a *Lin-Server*t a *Win-Server*ről. Mit tapasztalsz?

Vegyél fel egy szabályt, amely beengedi **a szerveres hálózaton belülről érkező** ping üzeneteket. Milyen protokollt engedélyeztél?

4.2.12. Haladó konfiguráció II.

A főnök panaszkodik, hogy lassú a webszerver. Szerinte az a baj, hogy túl sok a feldolgozott ping üzenet, és nyomatékosan kéri, hogy korlátozd az adott

időegységen belül érkező pingek számát (*rate-limiting*). Szerinted ez full baromság, de azért nekilátsz a feladatnak... Oldd meg, hogy percenként csak körülbelül 20 ping üzenet érkezzen meg, a többi dobódjon el.

Állítsd meg a pingelést, várj 5-10 másodpercet, majd kezdd el ismét. Mit tapasztalsz?

4.2.13. Haladó konfiguráció III.

Már megint a főnök hív. Szerinte a szerver által kiküldött válaszcsomagok TTL-jéből megállapítható, hogy Linux operációs rendszer fut rajta. Igaza lehet-e?

Kéri, hogy oldd meg, hogy a szerver inkább Windowsra hasonlítson a TTL-ek szempontjából. (Segítség: Windows esetén a TTL mező kezdőértéke 128.)

Listáztasd ki a tűzfalszabályokat. Hová lett a most felvett szabály?

Mit látni most a *Win-Server* konzolján (ahonnan a pingelést indítottad)?

4.2.14. A tűzfal állapotának mentése, visszatöltése I.

Listáztasd ki a tűzfalszabályokat. Hány szabályod van?

Indítsd újra a gépet. Változott-e bármi a tűzfal állapotát, beállításait illetően? Ha igen, mi? Ha vesztek el szabályok, futtasd újra a scripted, hogy visszaálljon az újraindítás előtti állapot.

4.2.15. A tűzfal állapotának mentése, visszatöltése II.

Telepítsd fel az *iptables-persistent* csomagot. Telepítés közben válaszd azt, hogy mentse az aktuális szabályokat.

Ismét indítsd újra a gépet. Változott-e bármi a tűzfal állapotát, beállításait illetően? Ha igen, mi, miért?

4.2.16. A tűzfal állapotának mentése, visszatöltése III.

Időközben a feltört nyomtatót sikerült helyreállítani, most már ártalmatlan. Töröld az erre vonatkozó tűzfalszabályt. Listáztasd ki a tűzfalszabályokat.

Indítsd újra a gépet. Változott-e bármi a tűzfal állapotát, beállításait illetően? Ha igen, mi, miért?

4.2.17. A tűzfal állapotának mentése, visszatöltése IV.

Ismét töröld a nyomtatóra vonatkozó tűzfalszabályt. Mentsd el a mostani állapotot, hogy a következő induláskor ne kerüljön vissza a törölt szabály.

Indítsd újra a gépet. Változott-e bármi a tűzfal állapotát, beállításait illetően? Ha igen, mi, miért?