Redundáns tűzfal konfiguráció OpenBSD/PF alapon

Csillag Tamás
cstamas@itk.ppke.hu
Pásztor Miklós
pasztor@ppke.hu

Pázmány Péter Katolikus Egyetem
Információs Technológiai Kar

Miről lesz szó?

- Motiváció
- Eszközök
  - OpenBSD
  - PF
  - Carp
  - Pfsync
- Probléma – megoldás párok

Motiváció

- Csomagszűrő tűzfalat akarunk
  - Biztonságos alapon
  - Egyszerűen konfigurálhatóan
  - Redundáns kiépítésben
    - Lehessen hardvert bővíteni, upgradelni üzemidőben is!

OpenBSD

- Egyike a szabad operációs rendszereknek
  - Nem csak ingyenes
- Születése
  - 1995, Theo de Raadt, Kanada
- Nemes unixos hagyományok folytatója
  - egyszerű, célratörő, sőt szikár és nyers
  - egységes, áttekinthető, nem szószátyár

OpenBSD fejlesztés

- Évente két új változat
- Évente fejlesztői összejövetel: *hackaton*
- Nem csak KISS elv (Keep it Small and Simple), hanem
- *Shut up and hack*
- *A biztonság a legfőbb szempont*

Az OpenBSD erős hálózati alkalmazásokban

- *Az OpenSSH ,,hazája"*
  - *Legtöbben innen ismerik a nevet: OpenBSD*
- *Routing protokoll implementációk*
  - *OSPF, BGP*
- *Tűzfal: PF*

PF, Packet Filter

- *Az IPFilter kiváltására*
- *Daniel Hartmaier (Svájc) munkája*
- *A linuxos iptables megfelelője*
- *Szűrés*
- *NAT*
- *Naplózás*
- *Átirányítás*
- *...*

Szűrési feltételek

- *IP cím*
  - *forrás*
  - *cél*
- *protokoll*
- *port*
  - *forrás*
  - *cél*
- *interfész*
- *operációs rendszer (!)*

A PF vezérlése

- */etc/pf.conf fájl*
- *pfctl parancs*

PF listák

- *Egy-egy szabályban felsorolásokat tehetünk portokra, IP címekre*
- *Például:*

```
pass out proto udp from any to
10.20.30.40 port {domain, ntp}
```

PF makrók

- *PF-ben nevekat adhatunk, változókat definiálhatunk*
- *Példák:*

```
dmz_if = "em1"
dns_server ="10.20.30.40"
```

- `pass out log $dmz_if proto udp from any to $dns_server port {domain, ntp}`

PF stateful filtering

- *A PF nem csak csomagokat, hanem kapcsolatokat lát*
- *Ha egy kapcsolat első csomagját átengedtük, rendelkezhetünk róla, hogy kapcsolat többi csomagja is átmenjen*
- *Az élő kapcsolatokról a PF egy belső táblázatot tart fenn*
  - *Ez teszi lehetővé, hogy pl. a window-n kívüli TCP csomagokat kiszűrje*

PF táblázatok

- *IPv4 vagy IPv6 címek egy halmaza*
- *Nevet rendelünk hozzá*
- *Szűrési, átirányítási, NAT szabályoknál hivatkozhatunk táblázatokra*
- *A táblázatokat manipulálhatjuk `pfctl` parancsokkal*
- *Példa*

```
table <goodguys> { 192.0.2.0/24,
!192.0.2.5 }
pass  in on fxp0 from <goodguys> to any
```

Scrub

- *A Scrub a csomagok normalizálására szolgáló eszköz. Megtehetjük, hogy:*
  - *IP fragmentumokat csak egyben továbbítunk*
  - *IP csomagok ID mezőjét randomizálunk*
  - *IP csomagok TTL mezőjét normalizáljuk*

Anchor-ok

- *Szabályok egy halmazát külön kezelhetjük*
- *Struktúrált, rugalmas konfigurálás*
  - *Később user space-ből manipulálhatjuk az egyes anchor-okat*
- *Pl:*

```
anchor horgony
load anchor horgony from
"/etc/horgony.0"
```

- *Aztán: #pfctl -a horgony pass in quick on $dmz_if proto tcp from 11.12.3.4 to $ns_server port domain keep state*

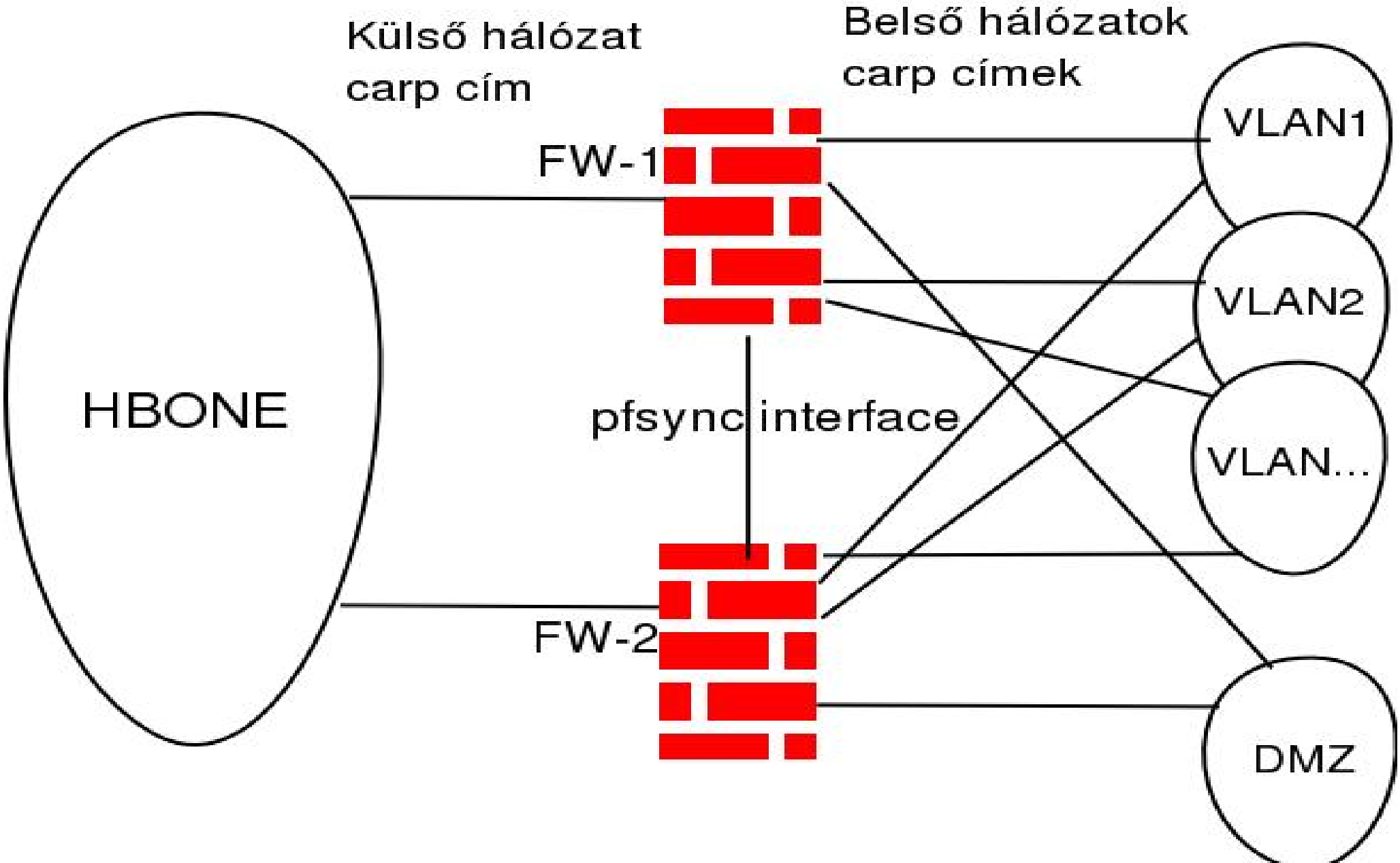
CARP - Common Address Redundancy Protocol

- *Két vagy több számítógép közös IP címen nyújt szolgáltatást*
- *Új, közös ethernet címen válaszolnak az ARP kérésekre*
- *Ha az egyik kiesik, zökkenőmentes marad a szolgáltatás*
- *Cisco VRRP-hez hasonló, annál többet tudó protokoll*
- *112-es IP protokoll, multicast csomagok*

pfsync

- *A tűzfal állapotváltozásainak szinkronizálására szolgáló protokoll*
- *A carp-hoz hasonlóan virtuális interfész*
- *Külön ethernet portot használunk a két tűzfal közt erre a célra*

Konfiguráció a PPKE ITK-n



Problémák és megoldások I.
Változó IP című gép beengedése

- Egy belső gépre egy változó IP című ADSL végpontról SSH elérést akarunk biztosítani
- Megoldás
  - A változó IP címhez DDNS címet rendelünk
  - Egy PF table elemeire engedjük meg az SSH elérést:

```
pass out  quick proto tcp from <ad_sl> to
$in_here port ssh $syn keep state
```

  - Egy cron job a tűzfalon a változó IP címet beteszi a táblázatba

Problémák és megoldások II.
Tantermi internet forgalom korlátozása

- A számítógépes tantermekben csökkenteni kell a hallgatók kísértését a hálózaton való bolyongásra :-)
- Megoldás
  - authpf shell
    - Elsősorban kapu nyitásra szolgálhat a PF kiegészítéseként
    - Ha `pista` felhasználó bejelentkezik, az `/etc/authpf/users/pista/authpf.rules`-ben levő szabályok betöltődnek az authpf anchor-ba

Problémák és megoldások II.
Tantermi internet forgalom korlátozása (folyt.)

- Itt éppen nem engedélyezésre, tiltásra van szükség
- Létrehozunk egy `nemnet` nevű felhasználót, authpf shell-lel, ilyen szabállyal:
  ```
  block return in  log quick  proto tcp
  from $user_ip/24
  ```
- Az egyes tantermek egy-egy /24-es tartományból kapnak címeket
- A szabály hatására az egész teremben „megszűnik az internet"
- Néhány szolgáltatást külön szabállyal engedélyezünk

Problémák és megoldások III.
Neptun túlterhelés

- Windows terminál szerverekre bejelentkezve használják a hallgatói információs rendszert
- Vizsgaidőszak kezdetén ezrével jönnek kérések a 3389-es tcp portra
- A terminál szerverre felépül a kapcsolat, de egy határon túl azonnal bomlik
  - Okosabb lenne, ha nem FIN-nel, hanem azonnal RESET-tel bontana...
- A PF állapottábláját ilyenkor több tízezer kapcsolat terheli

Problémák és megoldások III.
Neptun túlterhelés (folyt.)

- Megoldás
  - A PF-ben korlátozzuk az állapottáblázat méretét:
    ```
    set limit states 20000
    ```
  - Szabályozzuk az egyes bejegyzések elévülésének idejét:
    ```
    set timeout { adaptive.start 8000,
    adaptive.end 40000 }
    ```
    - Ennek hatására lineáris skála szerint változik az elévülési idő:
      ```
      (adaptive.end - number of states) /
      (adaptive.end - adaptive.start)
      ```
  - Egy-egy szolgáltatásra vonatkozó korlátozás:
    ```
    pass out log quick on $neptun_if proto tcp to
    $neptun port $neptuns_server $syn keep state
    (max 4000)
    ```

Problémák és megoldások IV.
FTP gondok

- A kliens gépekről minden kifele menő TCP kapcsolat megengedett, de befele semmi
- Aktív ftp-nél a szerver építene fel a saját 20-as portjáról a kliens valamelyik tcp portrjára kapcsolatot
- A tűzfalon ezt át kell engedni
- Erre való iptables-ben a *connection tracking*

Problémák és megoldások IV.

- *PF-nél a pftpx nevű külső kiegészítő használható*
- *FTP proxy, amihez átirányítjuk a kéréseket:*

```
rdr pass on $int_if proto tcp from $lan
to any port 21 -> 127.0.0.1 port 8021
```

- *Bevezetünk anchor-okat:*

```
anchor "pftpx/*"
nat-anchor "pftpx/*"
```

- *A külső szerverek a tűzfalról látják érkezni az ftp kapcsolatot*
- *A pftpx processz röptében beteszi a megfelelő szabályokat az anchor-ba*

Problémák és megoldások V.
Flow információk gyűjtése

- *Netflow: Cisco által fejlesztett nyílt protokoll*
  - *Adatgyűjtésre szolgál*
  - *Külső kollektor gépre UDP csomagokban mennek a flow-król összesítések*
- *A hálózatunk forgalmának jelentős része (belső vlan-ok, DMZ) csak a tűzfalunkon megy át*
- *Megoldás:*
  - *pfflowd, a PF kiegészítése*
  - *netflow formában exportál forgalmi adatokat*
  - *a pfsync mechanizmusát használja*

Problémák és megoldások VI.
Spam források fékezése

- *A hálózati sávszélesség növelése a spam küldőknek is kedvez*
- *Megoldás: a spamd nevű PF kiegészítő smtp szerverként funkcionál*
  - *Laaassaaaaan válaszol*
  - *Végül visszautasítja a levelet*
- *A PF-ben egy táblázat segítségével a spamd-hez tereljük a spam forrásokat:*

```
rdr pass on $ext_if inet proto tcp from <spamd>
to any port smtp tag SPAMD -> 127.0.0.1 port
8825
```

- *A spamd-setup program*
  - *Kezeli a spamd PF táblát*
    - *Veheti fájlból, és/vagy a hálózatról*
  - *Cron-ból indul*

Problémák és megoldások VII.
Spam és féreg elhanyagolt gépekről

- *Rosszindulatú kód által generált levelek*
- *Windows9x gépekről jövők nagy valószínűséggel ilyenek*
- *A PF forrás operációs rendszer szerint is képes szűrni*
  - *Passive fingerprinting: p0f*
    - *Michal Zalewski (Lenegyelo.) munkáján alapul*
    - *http://lcamtuf.coredump.cx/p0f-help*
- *Ez a sor átirányítja a vindoze9x-ről jövő leveleket spamd-hez:*

```
rdr pass on $ext_if inet proto tcp from any os
{"Windows 95", "Windows 98"} to any port smtp
tag SPAMD_WIN -> 127.0.0.1 port 8825
```

Az OpenBSD/PF páros

- *a redundáns tűzfal kialakításra alkalmas*
- *egyszerűen és hatékonyan kezelhető*
- *kellemes meglepetésekkel szolgál*
- *QoS kezelésre is alkalmas*
  - *prioritásos sorokat adhatunk meg*
  - *sávszélességet rendelhetünk a sorokhoz*
  - *nincs vele tapasztalatunk*

Olvasnivaló

- *http://www.benzedrine.cx/pf-paper.html*
- *http://www.openbsd.org/faq/pf/*
- *Jacek Artymiak: Building Firewalls with OpenBSD and PF ISBN 83-916651-1-9*
- *http://www.bgnett.no/~peter/pf/en/index.html*
- *http://www.mindrot.org/pfflowd.html*
- *http://www.stearns.org/p0f/*
  és
- *http://www.ppke.hu/~pasztor/openbsd-pf.html*