# **6.a** Mutassa be a forgalomirányítókon megvalósítható csomagszűrés elvét és megvalósításait! Ismertesse a konfiguráció jellemzőit és lépéseit?

# Access Control List – ACL

## ACL rendeltetése

* Hozzáférés szabályozás vagy hálózati forgalom szűrésére szolgál.
* Forgalomirányító csomagszűrőként viselkedik, amikor továbbítja vagy eldobja a csomagokat.
* Egy ACL permit és deny állítások sorrendezett listája.

## ACL elhelyezése a hálózatban

* Minden ACL-t oda kell helyezni, ahol a legnagyobb hatékonysággal képes szűrni.
* **Extended ACL:** Lehető legközelebbi forgalom forráshoz kell rakni.
* **Standard ACL:** Nem határozza meg a célcímet, ezért a célállomáshoz közel kell rakni.

## ACL típusok

* **Számozott:** Szűrés csak forráscím alapján
  + **Standard ACL (1-99-ig és 1300-1999-ig)**
    - Harmadik rétegbeli szűrés
  + **Extended ACL (100-199-ig és 2000-2699-ig)**
    - Harmadik és negyedik rétegbeli szűrés
* **Nevesített:** Szűrés protokoll, port, forrás/cél alapján
  + Csak betűk és számokból állhat a neve. (Javasolt nagybetűs)

## ACL konfigurálás lépései

* Szabályok:
  + **One ACL per protocol:**
    - Az interfészen értelmezett minden protokollhoz külön lista kell.
  + **One ACL per direction:**
    - ACL-ek egyszerre csak egy irányban vizsgálnak forgalmat.
    - Egy interfészen a kimenő/bemenő irányba két külön ACL kell.
  + **One ACL per interface:**
    - ACL-ek egy adott interfészen értelmezendők.

1. ACL céljának meghatározása
2. ACL létrehozása
3. ACL alkalmazása
4. ACL tesztelése például tesztcsomagok küldésével.

## Helyettesítő maszk hatása – Wildcard mask

* Arra szolgál, hogy megkönnyítse az IP-címek vagy más fejlécmezők tartományának megadását egy ACL-szabályban.
* **0-s bit**: A cím megfelelő bitjeit hasonlítja.
* **1-es bit**: A cím megfelelő bitjeit figyelmen kívül hagyja.
* **Speciális wild card maszkok**: 0.0.0.0; 255.255.255.255

# Context-Based Access Control – CBAC

## CBAC fő funkciói

* **Állapottartó szűrés – Stateful Packet Filtering**
  + Nem csak hálózati és szállítási réteg információk alapján vizsgálja a viszonyok állapotát, hanem alkalmazási réteg információkat is.
* **Forgalom figyelés – Traffic Inspection**
  + SYN flood támadások, TCP sorszámozást figyel és gyanúsakat eldobja.
* **Behatolás érzékelés – Intrusion Detection**
  + A syslog üzenetek átvizsgálásával ki lehet szűrni az smtp támadások és SYN flood támadások sajátosságait, ezeket a kapcsolatokat eldobja és riasztást, értesítést küld a rendszernek.

## CBAC működése

* TCP, UDP és ICMP kapcsolatokról információt tárol az állapot táblában. (state table)
* Állapot tábla alapján dinamikusan ACL-t hoz létre a visszajövő csomagok számára.
* CBAC ideiglenes nyílásokat hoz létre megadott kapcsolathoz, amik beengedik a blokkolt forgalmat.
* Az állapottábla automatikusan frissül a forgalom áramlásának megfelelően.
* CISCO IOS tűzfal 3 küszöbértéket is figyel a TCP DoS támadások kivédésére:
  + Félig megnyitott TCP kapcsolatok száma.
  + Félig megnyitott TCP kapcsolatok száma adott intervallumban.
  + Félig megnyitott TCP kapcsolatok száma egy adott host-tól.

## CBAC konfigurálása

1. **Interfész kiválasztása**
   1. Belső interfész ahonnan indulhat egy viszony felépítés.
2. **ACL konfigurálás az interfészen**
   1. Milyen típusú forgalmat engedélyezünk az interfészen
      1. Alap konfiguráció, hogy a belső hálózattól a külső hálózatig mindent, de a külső hálózattól a belső hálózating semmit.
      2. Engedélyezzük azt a forgalmat, amit meg kell vizsgálni a CBAC-nak.
      3. Implicit deny-t tegyük explicitté a naplózás miatt.
3. **Inspection rule megfogalmazása** a vizsgált forgalomra
4. **Alkalmazás a megfelelő interfészen**

# Zone-Based Policy Firewall – ZPF

* ACL-től független
* Mindent tiltunk, amíg külön nem engedjük
* Házirend minden forgalom hatással van, így nem kell több ACL/ellenőrzési művelet.

## ZPF funkciói

* **Inspect**
  + Automatikusan beengedi a válasz forgalmat.
  + Támogatja azokat a protokollokat, amik több párhuzamos kapcsolat felépítését igénylik.
* **Pass**
  + Hasonló az ACL permit-hez.
  + Nem követi a kapcsolat állapotát
  + Csak egy irányban engedi át a forgalmat
  + Megfelelő szabványt kell alkalmazni a válaszforgalom beengedésére
* **Drop**
  + Hasonló egy ACL deny-hoz
  + Blokkolt csomagok naplózása

## Tervezési szabályok:

* Zónát konfigurálni kell, mielőtt egy interfészt hozzárendelhetünk.
* Egy interfész egy biztonsági zónához rendelhető!
  + Közöttük forgalom engedélyezett (impliciten)
* Különböző zónák közötti forgalom engedélyezéséhez policy-t kell konfigurálni
  + Egy zónabeli és nem zónabeli interfész között a forgalom nem engedélyezett
* Zónák közötti események megadása: pass, inspect és drop
* Nem zónához tartozó interfészen CBAC-ot lehet konfigurálni.
* Ha egy interfészt nem akarunk zónához rendelni.
  + Mindent átenged, policy-vel konfigurált zónába tehetjük.

## Konfiguráció:

1. Tűzfal zónák létrehozása – zone security
2. Forgalmi osztályok definiálása - class-map type inspect
3. Tűzfal policy meghatározása - policy-map type inspect
   1. alkalmazása zónapárok között - zone-pair
4. Interfészek zónákhoz rendelése - zone-member security

## Zónák

* Self Zóna: Ha a router, a forrása vagy a célállomása egy forgalomnak
* DMZ (perimeter hálózat)
  + **Szolgáltatás**okat nyújt a külső hálózat irányába
  + **Biztonság**: DMZ és a belső hálózat között lesz még egy tűzfal
* Privát/Publikus zóna (Internet)

## A „self” zóna célja, feladata és jellemzői

* Alapértelmezetten a router interfészei a SELF zóna tagjai
* Forgalomszabályok definiálhatók akkor is, ha a zónapár egyik tagja a SELF zóna
  + Self zóna egy rendszer által definiált zóna = maga a router.
  + Nem kell konfigurálni az interfészeket, hogy tagja legyenek.
  + Az egyetlen kivétel az alapértelmezett deny all policy alól.