HÁLÓZATI ISMERETEK

5. forduló

A kategória támogatója: Deutsche Telekom IT Solutions

RENDELKEZÉSRE ÁLLÓ IDŐ:

30:00

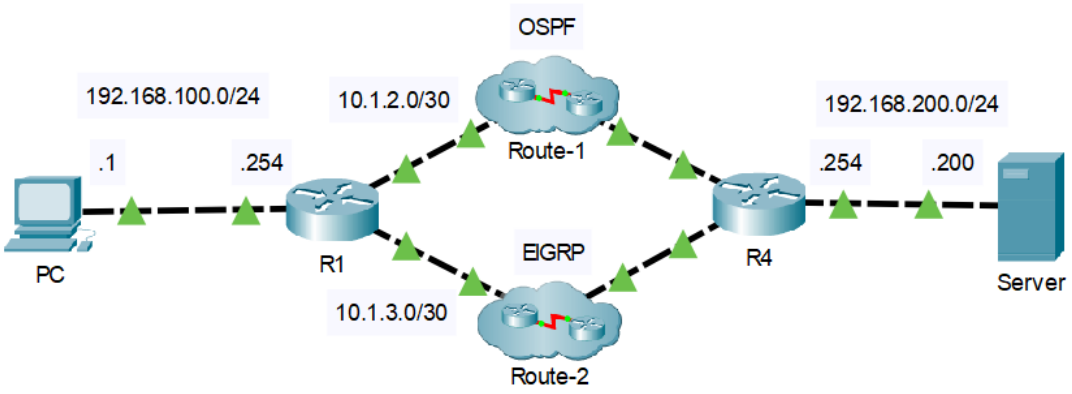
Ismertető a feladathoz

Az informatikai terület népszerűsítésére nyáron útjára indítottuk az **ITmap** platformot, ahol több mint **80 IT munkakört mutatunk be - fizetési adatokkal** együtt. Te mennyire értesz egyet a munkaköröknél szereplő információkkal? Kíváncsiak vagyunk tapasztalataidra: véleményezd Te is a hálózati területhez kapcsolódó pozíciók tartalmát!
https://itmap.hu/szakterulet/halozatok/

Felhasznált idő: 02:07/30:00 **Elért pontszám:** 0/40

1. feladat 0/8 pont

Egy cég hálózatában a 192.168.100.0/24-es és 192.168.200.0/24-es belső hálózatokat két redundáns útvonal köti össze az alábbi ábrán látható módon.



A Route-1-es hálózaton minden link sebessége 1 Gbps, OSPF routing protokollt használnak, míg a Route-2-es hálózaton EIGRP protokollt használnak, a linkek sebessége végig 100 Mbps. Útvonalösszegzés (summary) sehol nincs alkalmazva. Feltételezhetjük, hogy mindkét útvonal megfelelően működik, valamint hogy a topológiában szereplő összes eszköz a minimálisan szükséges, helyes konfigurációval rendelkezik. A hálózat már legalább 5 perce stabilan üzemel, a PC-ről a Server irányába indított ping teszt sikeres volt.

R1 konfigurációjának releváns részletei a következők:

```
R1# sh run | section eigrp
router eigrp 666
 network 192.168.100.0 0.0.0.255
 network 10.1.3.0 0.0.0.3
 maximum-paths 2
```

```
R1# sh run | section ospf
router ospf 42
 log-adjacency-changes
 network 192.168.100.0 0.0.0.255 area 0
 network 10.1.2.0 0.0.0.3 area 0
 maximum-paths 2
```

A PC-ről szeretnénk letölteni több kisebb fájlt a szerverről, FTP-n keresztül. Merre fog haladni a hálózatban a PC kimenő forgalma?

Válasz

- A kimenő forgalom átlagosan körülbelül 9%-a Route-1-en keresztül, míg körülbelül 91%-a Route-2-n keresztül fog haladni.
- A kimenő forgalom átlagosan körülbelül 91%-a Route-1-en keresztül, míg körülbelül 9%-a Route-2-n keresztül fog haladni.
- A kimenő forgalom átlagosan fele Route-1-en keresztül, a másik fele Route-2-n keresztül fog haladni.
- A teljes kimenő forgalom Route-1-en keresztül fog haladni.
- A teljes kimenő forgalom Route-2-n keresztül fog haladni.
- A teljes kimenő forgalom eldobásra kerül R1-en konfigurációs hiba miatt.

Magyarázat

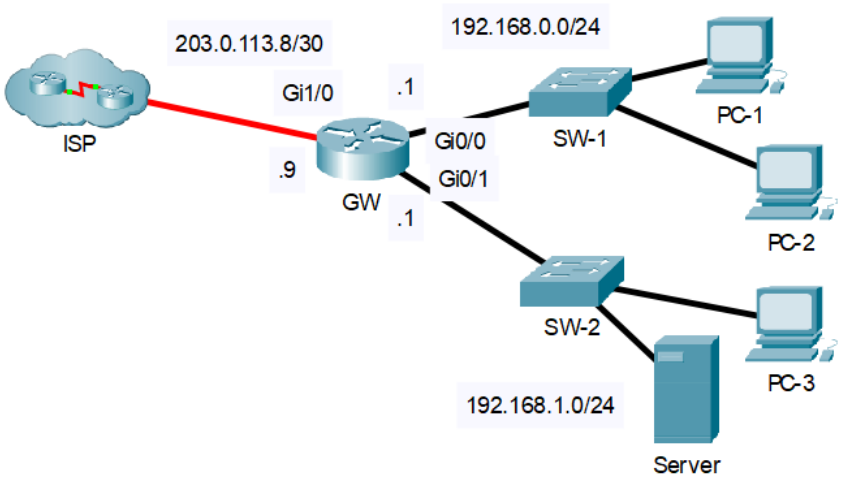
A szerverhez két útvonalunk van, egy gyors a Route-1-en át, amelyet OSPF-en keresztül ismertünk meg, és egy lassú a Route-2-n át, amelyet EIGRP-n keresztül. Mivel két különböző protokollról van szó, első körben az adminisztratív

távolság (administrative distance, AD) dönt, hogy melyik útvonalat használjuk. Az AD (internal) EIGRP esetén 90, OSPF esetén 110. Mivel a kisebb nyer, kizárólag az EIGRP-n megismert útvonalat használjuk, hiába az csak 100 Mbites.

A fele erre, fele arra megoldás akkor működne, ha mindkét útvonalon ugyanazt az Equal-Cost Multi-Path (ECMP) routingra képes protokollt használnánk (az OSPF és az EIGRP is jó lenne), továbbá mindkét útvonal költsége pontosan ugyanannyi lenne. Ehhez viszont jelentősen fel kellene újítani a hálózatot (vagy kézzel átírni az egyes interfészek jellemzőit). A nem egyenlő arányban történő elosztáshoz már UCMP (Unequal-Cost Multi-Path) routing kellene. Ilyet EIGRP-vel lehetne megvalósítani, szintén jelentős átszámolások, módosítások mellett.

2. feladat 0/8 pont

Egy új cég kis hálózatában szükség lenne arra, hogy belülről el lehessen érni az internetet. A szolgáltató összesen 1 fix publikus IP-címet biztosított a cégnek.



Mely konfiguráció(k) oldja/oldják meg a feladatot? Feltételezhetjük, hogy az IP-címek már helyesen be vannak állítva, a hálózat minden egyéb értelemben megfelelően működik.

Válaszok

- [x]

```
GW(config)# interface GigabitEthernet0/0
GW(config-if)# ip nat inside
GW(config-if)# exit

GW(config)# interface GigabitEthernet0/1
GW(config-if)# ip nat inside
GW(config-if)# exit

GW(config)# interface GigabitEthernet1/0
GW(config-if)# ip nat outside
GW(config-if)# exit

GW(config)# access-list 1 permit 192.168.0.0 0.0.0.255
GW(config)# access-list 1 permit 192.168.1.0 0.0.0.255

GW(config)# ip nat inside source list 1 interface GigabitEthernet1/0 overload
```

```
GW(config)# interface GigabitEthernet0/0
GW(config-if)# ip nat outside
GW(config-if)# exit

GW(config)# interface GigabitEthernet0/1
GW(config-if)# ip nat outside
GW(config-if)# exit

GW(config)# interface GigabitEthernet1/0
GW(config-if)# ip nat inside
GW(config-if)# exit

GW(config)# access-list 1 permit 192.168.0.0 0.0.0.255
GW(config)# access-list 1 permit 192.168.1.0 0.0.0.255

GW(config)# ip nat inside source list 1 interface GigabitEthernet1/0 overload
```

```
GW(config)# interface GigabitEthernet0/0
GW(config-if)# ip nat inside
GW(config-if)# exit

GW(config)# interface GigabitEthernet0/1
GW(config-if)# ip nat inside
GW(config-if)# exit

GW(config)# interface GigabitEthernet1/0
GW(config-if)# ip nat outside
GW(config-if)# exit

GW(config)# access-list 1 permit 203.0.113.8 0.0.0.252

GW(config)# ip nat inside source list 1 interface GigabitEthernet1/0 overload
```

```
GW(config)# interface GigabitEthernet0/0
GW(config-if)# ip nat outside
GW(config-if)# exit

GW(config)# interface GigabitEthernet0/1
GW(config-if)# ip nat outside
GW(config-if)# exit

GW(config)# interface GigabitEthernet1/0
GW(config-if)# ip nat inside
GW(config-if)# exit

GW(config)# access-list 1 permit 203.0.113.8 0.0.0.252

GW(config)# ip nat pool NAT_POOL 192.168.0.1 192.168.1.254 prefix-length 23
GW(config)# ip nat inside source list 1 pool NAT_POOL overload
```

```
GW(config)# interface GigabitEthernet0/0
GW(config-if)# ip nat inside
GW(config-if)# exit

GW(config)# interface GigabitEthernet0/1
GW(config-if)# ip nat inside
GW(config-if)# exit

GW(config)# interface GigabitEthernet1/0
GW(config-if)# ip nat outside
GW(config-if)# exit

GW(config)# access-list 1 permit 203.0.113.8 0.0.0.252

GW(config)# ip nat pool NAT_POOL 192.168.0.1 192.168.1.254 prefix-length 23
GW(config)# ip nat inside source list 1 pool NAT_POOL overload
```

```
GW(config)# interface GigabitEthernet0/0
GW(config-if)# ip nat inside
GW(config-if)# exit

GW(config)# interface GigabitEthernet0/1
GW(config-if)# ip nat inside
GW(config-if)# exit

GW(config)# interface GigabitEthernet1/0
GW(config-if)# ip nat outside
GW(config-if)# exit

GW(config)# access-list 1 permit 192.168.0.0 0.0.0.255
GW(config)# access-list 1 permit 192.168.1.0 0.0.0.255

GW(config)# ip nat pool NAT_POOL 203.0.113.9 203.0.113.9 prefix-length 30
GW(config)# ip nat inside source list 1 pool NAT_POOL overload
```

```
GW(config)# interface GigabitEthernet0/0
GW(config-if)# ip nat outside
GW(config-if)# exit

GW(config)# interface GigabitEthernet0/1
GW(config-if)# ip nat outside
GW(config-if)# exit

GW(config)# interface GigabitEthernet1/0
GW(config-if)# ip nat inside
GW(config-if)# exit

GW(config)# access-list 1 permit 192.168.0.0 0.0.0.255
GW(config)# access-list 1 permit 192.168.1.0 0.0.0.255
```

```
GW(config)# ip nat pool NAT_POOL 203.0.113.9 203.0.113.9 prefix-length 30
GW(config)# ip nat inside source list 1 pool NAT_POOL overload
```

```
GW(config)# interface GigabitEthernet0/0
GW(config-if)# ip nat outside
GW(config-if)# exit

GW(config)# interface GigabitEthernet0/1
GW(config-if)# ip nat outside
GW(config-if)# exit

GW(config)# interface GigabitEthernet1/0
GW(config-if)# ip nat inside
GW(config-if)# exit

GW(config)# access-list 1 permit 203.0.113.8 0.0.0.252

GW(config)# ip nat inside source list 1 interface GigabitEthernet1/0 overload
```

Magyarázat

A feladat leírásából (és persze a lehetséges válaszokból) látszik, hogy hálózati címfordításra (Network Address Translation, NAT) van szükség. A belső hálózat interfészeit (GigabitEthernet0/0 és GigabitEthernet0/1) belsőnek, a külső hálózat felé néző interfészt (GigabitEthernet1/0) külsőnek kell jelölni (ip nat inside/outside). Szükségünk van egy listára, ami a NAT-olandó (belső) címeket tartalmazza, ezek a 192.168.0.0/24 és 192.168.1.0/24. Mivel egy külső címünk van, innentől két megoldási irány létezik. Az egyszerűbb, ha megadjuk, hogy a listán található belső címeket NAT-olja a külső interfészen (az interfészt név szerint megadva). A másik, kicsit bonyolultabb megoldás, ha definiálunk egy poolt, amely a külső címeket tartalmazza (ilyenből ugye jelenleg egy van), majd megadjuk, hogy a belsős címlistán szereplő címeket fordítsa a poolban található cím(ek)re. Mivel több belső címünk van mint külső, az overload kulcsszó mindkét esetben kell, hogy a portok is fordításra kerüljenek (NAPT).

Így tehát a jó megoldások:

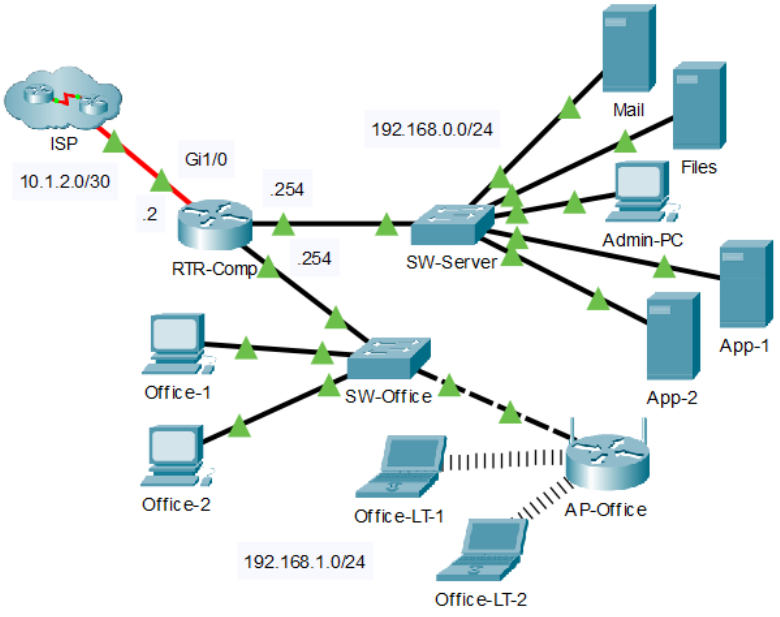
```
GW(config)# interface GigabitEthernet0/0
GW(config-if)# ip nat inside
GW(config-if)# exit

GW(config)# interface GigabitEthernet0/1
GW(config-if)# ip nat inside
GW(config-if)# exit

GW(config)# interface GigabitEthernet1/0
GW(config-if)# ip nat outside
GW(config-if)# exit

GW(config)# access-list 1 permit 192.168.0.0 0.0.0.255
GW(config)# access-list 1 permit 192.168.1.0 0.0.0.255

GW(config)# ip nat inside source list 1 interface GigabitEthernet1/0 overload
```

```
GW(config)# interface GigabitEthernet0/0
GW(config-if)# ip nat inside
GW(config-if)# exit

GW(config)# interface GigabitEthernet0/1
GW(config-if)# ip nat inside
GW(config-if)# exit

GW(config)# interface GigabitEthernet1/0
GW(config-if)# ip nat outside
GW(config-if)# exit

GW(config)# access-list 1 permit 192.168.0.0 0.0.0.255
GW(config)# access-list 1 permit 192.168.1.0 0.0.0.255

GW(config)# ip nat pool NAT_POOL 203.0.113.9 203.0.113.9 prefix-length 30
GW(config)# ip nat inside source list 1 pool NAT_POOL overload
```

3. feladat 0/8 pont

Egy kisebb cég IT-mindenese vagyok. Olvastam a hírekben, hogy terjedőben van egy új malware, amely cégeket fertőz meg. Először hazaküld minden dokumentumot, amit talál, majd letitkosítja az összes rendszert, váltságdíjat követelve a titkosítás feloldásáért. A főnökség leginkább a titkos dokumentumok kiszivárgásától tart, így megkérnek, hogy gátoljam meg, hogy a céges hálózatból kiszivárogtathasson a "vírus" bármit is, ha esetleg megfertőződnétek. Szerencsére a malware készítői meglehetősen amatőrök voltak, ugyanis a neten utánaolvasva kiderítettem, hogy minden rendszerük a 198.51.100.0/24-es IP-tartományban található.

Tanulmányozva a cég hálózatát, a tervem az, hogy készítek egy ACL-t, amely a kérdéses IP-tartomány felé menő forgalmat tiltja.



Egy terminált nyitva belépek az RTR-Comp eszközre, és kiadom a következő parancsokat:

```
RTR-Comp(config)# ip access-list extended BLOCK_EXFILTRATION
RTR-Comp(config-ext-nacl)# deny ip any 198.51.100.0 0.0.0.255
RTR-Comp(config-ext-nacl)# permit ip any any
```

Ezek után még mielőtt bármilyen más parancsot kiadnék, nyitok egy másik terminált, belépek még egyszer, majd addig próbálok pingelni különböző címeket a tiltandó tartományból, amíg találok egyet, ami válaszol.

Ezt követően beállítom az ACL-t az interfészre:

```
RTR-Comp(config)# interface GigabitEthernet1/0
RTR-Comp(config-if)# ip access-group BLOCK_EXFILTRATION out
RTR-Comp(config-if)#
```

Hogy kipróbáljam, minden rendben van-e, a másik terminálból ismét megpingelem a távoli gépet, amelyet korábban sikerült.

```
RTR-Comp> ping 198.51.100.25

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 198.51.100.25, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/3/6 ms

RTR-Comp>
```

Meglepődve nézem az eredményt. **Mi történt, mi a probléma?**

Válasz

- ○ Az *ip access-group* parancs kiadása után nem léptem ki az interfész konfigurációs módjából, így a parancs hatása még nem érvényesült.
- ○ Az *ip access-group* parancs érvényre juttatásához ki kell adni egy *shutdown*, majd egy *no shutdown* utasítást.
- ○ Minden megfelelően működik, írja is, hogy *Success rate is 100 percent*.
- ○ A feladat csak standard ACL segítségével oldható meg, én viszont extended ACL-t hoztam létre.
- ○ A feladat csak extended ACL segítségével oldható meg, én viszont standard ACL-t hoztam létre.
- ○ Az ACL-ben megadott maszk helytelen.
- ○ Az ACL csak IP forgalmat tilt, a ping viszont ICMP-t használ.
- ○ A gond az, hogy a második terminálon már be voltam lépve, mielőtt az ACL-t érvényesítettem, így arra a munkamenetre még nem volt érvényes. Ha belépnék újra, már jó lenne.
- ○ A forgalom nem került tiltásra, mert pár perccel korábban megpingeltem a másik oldalt. Próbáljam meg 5 perc múlva újra.
- ○ Az ACL-ben a tiltásnál felcseréltem a forrást és a célt.
- ○ A problémát az ACL *permit ip any any* sora okozza, tévesen vettem fel.
- ○ Az *ip access-group* parancs helyett az *ip access-class* parancsot kellett volna használnom.
- ○ A többi felsorolt válaszlehetőség közül egyik sem helyes.

Magyarázat

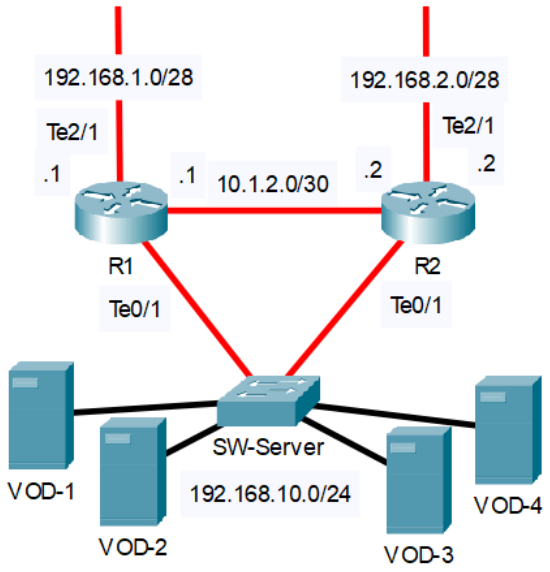
A felsorolt válaszlehetőségek közül egyik sem helyes. A hibát ott követtem el, hogy a routerről próbáltam meg pingelni a távoli gépet. A saját maga által keletkeztetett kimenő forgalomra nem érvényesek a kimenő ACL-ek, így a ping kérés kiment, a válasz visszajött. Ha bármely más céges eszközről próbáltam volna pingelni, az már nem sikerült volna. Érdemes lett volna egy bejövő ACL-t is felvennem, az megfogta volna a router pingjeire érkező válaszokat is.

Helytelen válaszok:

- *Az ip access-group parancs kiadása után nem léptem ki az interfész konfigurációs módjából, így a parancs hatása még nem érvényesült.* A parancs hatása azonnal érvényesül, nem kell kilépni.
- *Az ip access-group parancs érvényre juttatásához ki kell adni egy shutdown, majd egy no shutdown utasítást.* Lásd az előző pontot.
- *A gond az, hogy a második terminálon már be voltam lépve, mielőtt az ACL-t érvényesítettem, így arra a munkamenetre még nem volt érvényes. Ha belépnék újra, már jó lenne.* Nem ettől függ az ACL-ek működése.
- *Minden megfelelően működik, írja is, hogy 'Success rate is 100 percent'.* Ez pont azt jelenti, hogy minden válasz megérkezett, tehát sikeresen kimentek a kérések is. Ez az ellenkezője annak, amit szerettem volna.
- *A feladat csak standard ACL segítségével oldható meg, én viszont extended ACL-t hoztam létre.* Standard ACL-ben csak a forrást tudom megadni, de én a célt szeretném.
- *A feladat csak extended ACL segítségével oldható meg, én viszont standard ACL-t hoztam létre.* A kiadott parancsból látható, hogy de, extended ACL-t hoztam létre.
- *Az ACL-ben megadott maszk helytelen.* Nem helytelen. Wildcard maszkot kell megadni, ami /24-es hálózat esetén 0.0.0.255.
- *Az ACL csak IP forgalmat tilt, a ping viszont ICMP-t használ.* Az ICMP IP felett utazik, tehát ha az IP forgalmat tiltom, akkor az ICMP-t is.
- *A forgalom nem került tiltásra, mert pár perccel korábban megpingeltem a másik oldalt. Próbáld meg 5 perc múlva újra.* Bár lehetne ilyesmit csinálni (ld. reflexív ACL-ek), de nem így, és most nem is erről van szó.
- *A problémát az ACL 'permit ip any any' sora okozza, tévesen vettem fel.* Nem, arra szükség van. Ha az nincs ott, az implicit deny ip any any miatt minden forgalmat letiltok.
- *Az ACL-ben a tiltásnál felcseréltem a forrást és a célt.* Nem cseréltem fel, jó a sorrend. Először jön a forrás (nálam any), utána a cél (a tiltandó tartomány). A célom pedig az volt, hogy a kimenő forgalmat tiltsam.
- *Az ip access-group parancs helyett az ip access-class parancsot kellett volna használnom.* Az *access-class* parancsot (*ip* nélkül) a vonalas (line) terminálokhoz történő hozzáférés szabályzására használjuk, interfészre nem adható ki.

4. feladat 0/8 pont

Egy adatközpontba látogatsz, ahol egy szerverfarm tervezésekor az volt az egyik fő cél, hogy egy router meghibásodása ne okozzon kiesést. Miután lelkesen érdeklődtél a megvalósítás mikéntjéről, az ottani adminisztrátor elküldte neked a topológia egyszerűsített rajzát, valamint a routerek konfigurációjának egy-egy releváns részletét. Azt is elmondta, hogy a routerek között iBGP-vel történik az útvonalak szűrése és terjesztése (erről a feladatban feltételezheted, hogy megfelelően működik).



R1:

```
!
track 10 interface TenGigabitEthernet2/1 ip routing
!
!
interface TenGigabitEthernet0/1
 ip address 192.168.10.252 255.255.255.0
 standby 100 ip 192.168.10.253
 standby 100 priority 120
 standby 100 timers msec 500 msec 2000
 standby 100 track 10 decrement 20
 standby 100 authentication oitm
 standby 100 version 2
!
```

R2:

```
!
track 10 interface TenGigabitEthernet2/1 ip routing
!
!
interface TenGigabitEthernet0/1
 ip address 192.168.10.254 255.255.255.0
 standby 100 ip 192.168.10.253
 standby 100 priority 110
 standby 100 timers msec 500 msec 2000
 standby 100 track 10 decrement 20
 standby 100 authentication oitm
 standby 100 version 2
!
```

Értelmezd a konfigurációt, majd jelöld meg az igaz állításokat!

Válaszok

- [ ] A szervereken a 192.168.10.254-es IP-címet kell beállítani alapértelmezett átjárónak a legnagyobb rendelkezésre állás érdekében.

- [x] A szervereken a 192.168.10.253-as IP-címet kell beállítani alapértelmezett átjárónak a legnagyobb rendelkezésre állás érdekében.
- [ ] A szervereken a 192.168.10.252-es IP-címet kell beállítani alapértelmezett átjárónak a legnagyobb rendelkezésre állás érdekében.
- [ ] Hibamentes állapotban mindig az R1 lesz az aktív eszköz.
- [ ] Hibamentes állapotban mindig az R2 lesz az aktív eszköz.
- [ ] Hibamentes állapotban R1 és R2 mindig mindketten aktív eszközök lesznek.
- [x] Nem jelenthető ki egyértelműen, hogy hibamentes állapotban melyik lesz az aktív eszköz.
- [ ] R1 és R2 között hibamentes állapotban terheléselosztás valósul meg.
- [x] Hiba esetén kevesebb mint 3 másodperc alatt megtörténhet az átállás.
- [ ] Hitelesítésre MD5 algoritmust használnak az eszközök.
- [ ] Hitelesítésre SHA1 algoritmust használnak az eszközök.
- [ ] A konfiguráció által nyújtott redundancia csak a 100-as VLAN-ra érvényes.
- [ ] A redundancia VRRP segítségével valósul meg.
- [ ] A redundancia GLBP segítségével valósul meg.
- [x] A redundancia HSRP segítségével valósul meg.

Magyarázat

Az interfészeken kiadott standby parancsokból látszik, hogy HSRP-t (Hot Standby Router Protocol) használunk (és nem VRRP-t vagy GLBP-t, melyeknél vrrp vagy glbp szerepelne a parancsokban).

A két eszköz a saját IP-k mellett egy közös virtuális IP-t is használ, mely egyszerre mindig csak az egyik oldalon aktív, és az adott oldal hibája esetén ezt átveszi magához a másik fél. Ez a virtuális IP a *standby 100 ip 192.168.10.253* sorokban látható, ezt kell beállítani a szervereken alapértelmezett átjáróként a legnagyobb rendelkezésre állás érdekében (ha a két router IP-je közül választjuk valamelyiket, az adott router hibája esetén nem lesz kapcsolatunk kifelé).

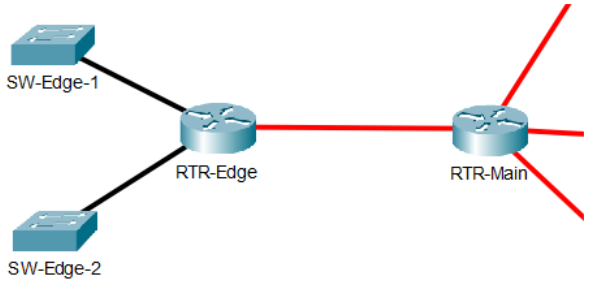
A HSRP nem tud aktív-aktív módban működni, így mindig valamelyik egy eszköz lesz az aktív. Kezdetben az R1 kezd aktívként (priority 120 vs. 110), meghibásodása esetén ezt átveszi az R2. Mivel nincs beállítva preemption, R1 a helyreállása után nem veszi vissza az aktív szerepet magához, így nem jelenthető ki egyértelműen, hogy hibamentes állapotban melyik lesz az aktív eszköz. Mivel egyszerre mindig csak az egyik eszköz aktív, terheléselosztás nem valósul meg.

A *standby 100 timers* sor alapján kb. 2 másodperc után megtörténik az átállás a másik oldal eltűnése esetén. (De ha a másik oldal él és hibát detektál, szólhat a túloldalnak, és akkor sokkal hamarabb is megtörténhet az átállás.)

Hitelesítésre az eszközök plaintext jelszót használnak hashelés nélkül. Az MD5 jobb választás lenne, SHA1-et nem is lehet választani. A konfiguráció által nyújtott redundancia nem csak a 100-as VLAN-ra érvényes. A 100-as szám itt a HSRP csoportazonosítója, nincs köze VLAN-okhoz.

5. feladat 0/8 pont

Egy fióktelepre hívnak, ahol egyelőre ismeretlen hiba miatt órák óta szakadozik az internetelérés. Hibafeltárás közben kiderül, hogy a szolgáltatónál volt a hiba, de addig is találtál egy érdekes konfigurációt, melyet szeretnél megérteni, hogyan működik.



RTR-Edge és RTR-Main EIGRP protokollt futtatnak. RTR-Main az alábbi útvonalakat hirdeti RTR-Edge felé összegzés (summary) nélkül:

- 10.10.0.0/24
- 10.10.1.0/24
- 10.10.2.0/24
- 10.10.3.0/25
- 10.10.3.128/25
- 10.10.4.0/24
- 10.10.5.0/25
- 10.10.5.128/26
- 10.10.5.192/26
- 10.10.6.0/24
- 10.10.7.0/25

RTR-Edge konfigurációjának egy részlete pedig az alábbi:

```
!
ip access-list standard ACL_A
 permit 10.10.0.0 0.0.0.255
 permit 10.10.6.0 0.0.0.127
!
ip access-list standard ACL_B
 permit 10.10.1.0 0.0.0.255
 permit 10.10.2.0 0.0.0.255
!
ip access-list standard ACL_C
 permit 10.10.1.0 0.0.0.255
 permit 10.10.3.0 0.0.0.255
!
ip prefix-list PREFIX_LIST seq 10 permit 10.10.2.0/24
ip prefix-list PREFIX_LIST seq 20 permit 10.10.4.0/23 ge 24 le 25
!
route-map ROUTE_MAP permit 10
 match ip address ACL_A
!
route-map ROUTE_MAP deny 20
 match ip address ACL_B ACL_C
!
route-map ROUTE_MAP permit 30
 match ip address prefix-list PREFIX_LIST
!
route-map ROUTE_MAP permit 40
 set tag 53
!
!
router eigrp 5
 network 172.16.200.0 0.0.0.3
 distribute-list route-map ROUTE_MAP in
 no auto-summary
!
```

A fenti információk ismeretében: **mely útvonalakat fogja RTR-Edge 53-as taggel ellátni?** (Feltételezhetjük, hogy az eszközök megfelelően fel vannak konfigurálva, a hálózat megbízhatóan működik.)

Válaszok

- [ ] 10.10.0.0/24
- [ ] 10.10.1.0/24
- [ ] 10.10.2.0/24
- [ ] 10.10.3.0/25
- [ ] 10.10.3.128/25
- [ ] 10.10.4.0/24
- [ ] 10.10.5.0/25
- [x] 10.10.5.128/26
- [x] 10.10.5.192/26

- [ ] 10.10.6.0/24
- [x] 10.10.7.0/25

- [ ] Egyik útvonalra sem kerül rá a tag

Magyarázat

RTR-Edge konfigurációjában látható, hogy a bejövő útvonalakra érvényesül a *ROUTE_MAP* nevű route map. Ennek utolsó állomásaként feltétel nélkül beállítódik az 53-as tag azokra az útvonalakra, amelyek eljutnak odáig, azaz semmilyen korábbi ellenőrzésre nem illeszkednek (hiszen illeszkedés esetén megáll a feldolgozás, mert sehol sem szerepel *continue* utasítás). A feladat így tehát az, hogy végignézzük a feltételeket, olyan útvonalakat keresve, amelyek egyik ACL-re vagy prefixlistára sem illeszkednek.

> Az útvonalak listájánál használjuk a *Másolás* gombot az útvonalak vágólapra helyezésére, majd innen illesszük be őket tetszőleges szövegszerkesztőbe. Ha egy útvonal illeszkedik valamire, töröljük ki a listából, ami pedig a végén marad, azok lehetnek a jó válaszok.
>
> A route-map 10-es sorában egy feltételt vizsgálunk, az *ACL_A*-t. Ez illeszkedik a 10.10.0.0/24-re, így azt kihúzzuk. ~~Az ACL-ben szerepel még a 10.10.6.0/25 is, de ez nem illeszkedik egyetlen leküldött útvonalra sem (a 10.10.6.0/24 nagyobb, így az marad).~~
>
> A következő (20-as) sorban két ACL (*ACL_B* és *ACL_C*) szerepel egymás mellett, ilyenkor ezek VAGY kapcsolatban vannak, tehát kiesnek azok az útvonalak, amelyek a kettő közül legalább az egyikre illeszkednek. *ACL_B* miatt kiesik a 10.10.1.0/24 és 10.10.2.0/24, *ACL_C* miatt kiesik még a 10.10.3.0/24 is.
>
> A következő (30-as) sorban egy prefixlistát kell értelmezni. A lista első sora kiejtené a 10.10.2.0/24-et, de az már korábban kiesett. A második sora illeszkedik azon alhálózatokra, amelyek a 10.10.4.0/23 alá tartoznak ÉS a prefix hossza legalább 24 de legfeljebb 25. Ezek a 10.10.4.0/24 és 10.10.5.0/25, ők kiestek. (A 10.10.5.128/26 és 10.10.5.192/26 maradnak a 26-os prefixhossz miatt.)
>
> El is jutottunk a taget beállító (40-es) sorhoz. Itt nincs megadva illeszkedési feltétel, így minden (korábban el nem fogadott és el nem utasított) útvonal illeszkedni fog. Ezek a 10.10.5.128/26, a 10.10.5.192/26~~, a 10.10.6.0/24~~ és a 10.10.7.0/25.

Frissítve (2021.11.21. 23:09): érkeztek visszajelzések, miszerint a 10.10.6.0/24-es útvonalra sem kellene rákerülnie a tagnek. Ellenőriztük, az észrevétel helyes. Ha egy standard ACL-t *distribute-list*ben használunk, a benne szereplő wildcard mask nem számít, csak a forráscímben szereplő bitek, így *ACL_A permit 10.10.6.0 0.0.0.127*-es sora illeszkedni fog a 10.10.6.0/24-re is, emiatt pedig nem jut el odáig a kiértékelésben, hogy rákerülhetne a tag. A válaszok helyességét és a pontozást ennek megfelelően javítottuk.