

Budapesti Műszaki Szakképzési Centrum

Neumann János Informatikai Technikum

Szakképesítés neve: Informatikai rendszer- és alkalmazás-üzemeltető technikus

száma: 5-0612-12-02

VIZSGAREMEK

Aranytoll Könyvelés hálózata és szerverei Dokumentáció

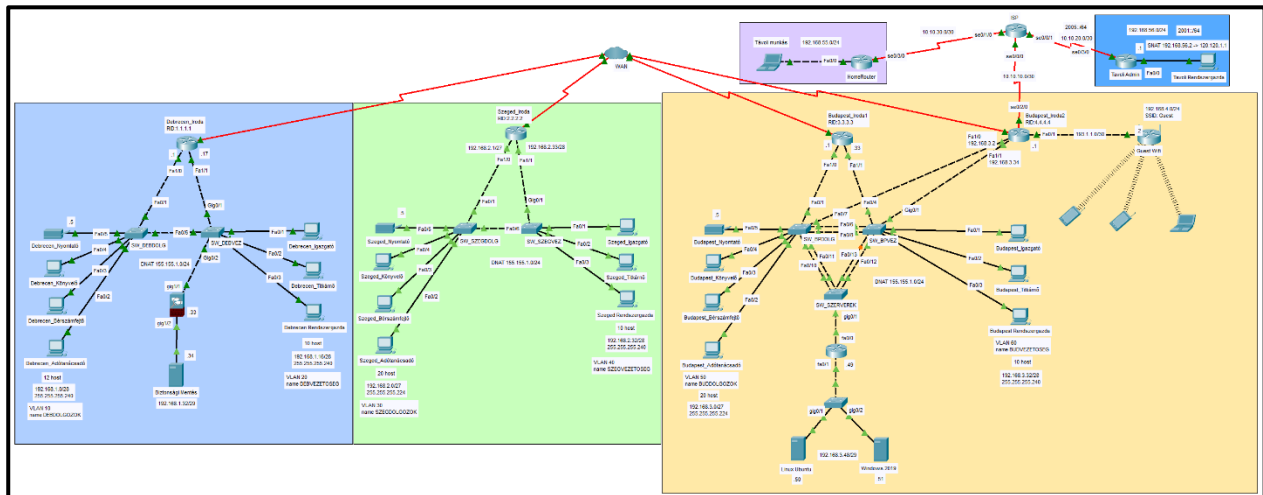
Gyetvai Fanni, Vörös Csanád
2/14.A

Budapest, 2022.

Tartalomjegyzék

Tartalomjegyzék	1
Telephelyek	2
Aranytoll Könyvelő iroda.....	3
Munkamegosztás	3
IP címek kiosztása	4
Statikus IPv4 címek kiosztása	4
Statikus IPv6 címek beállítása.....	4
Kapcsolók konfigurálása	5
VLAN-ok.....	5
VTP	6
Port biztonság.....	7
STP	8
Forgalomirányítás.....	9
IPv4 statikus forgalomirányítás	9
IPv4 dinamikus forgalomirányítás	10
IPv6 dinamikus forgalomirányítás	11
Alkalmazott WAN technológia	12
Frame Relay	12
HSRP	14
Alapvető biztonsági megoldások.....	15
Címfordítás	16
Dinamikus címfordítás	16
Statikus címfordítás.....	17
VPN	18
Hardveres tűzfal.....	19
Szerverek és felhőszolgáltatások.....	20
Linux Ubuntu	20
HTTP/HTTPS.....	20
DHCP	23
Windows 2019.....	24
DNS	24
Active Directory	25
Automatizált szoftver telepítés	26
Fájl- és nyomtató megosztás	28
Automatizált mentés.....	29
Hálózatprogramozás	30

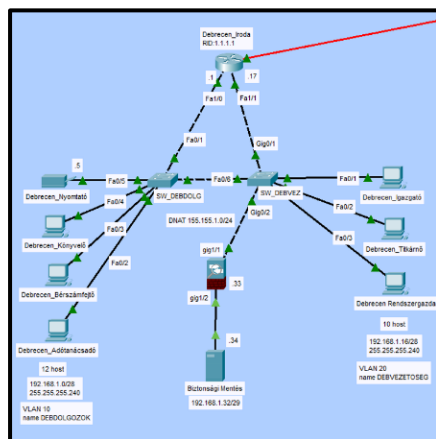
Telephelyek



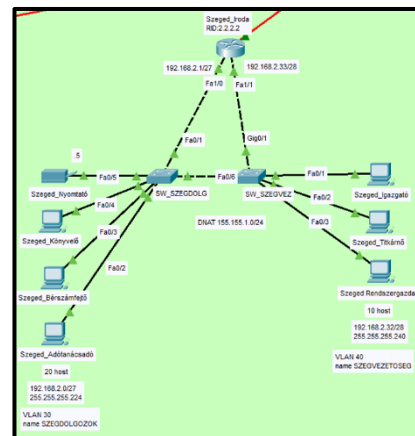
1.ábra

Szeged Iroda

Debrecen Iroda

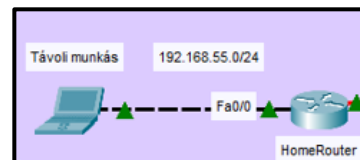


2.ábra



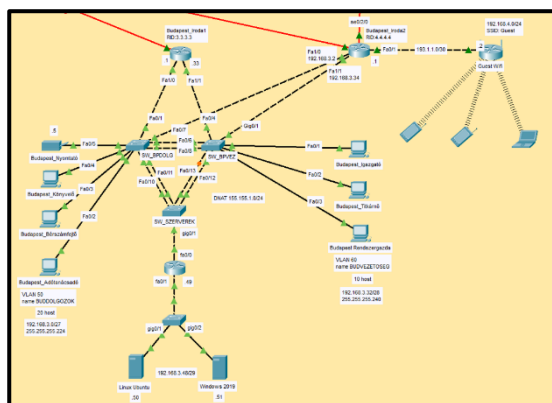
3.ábra

Távoli Rendszergazda



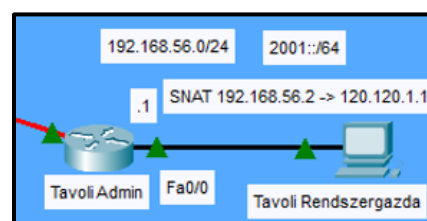
5.ábra

Budapest Iroda



4.ábra

Távoli Munkás



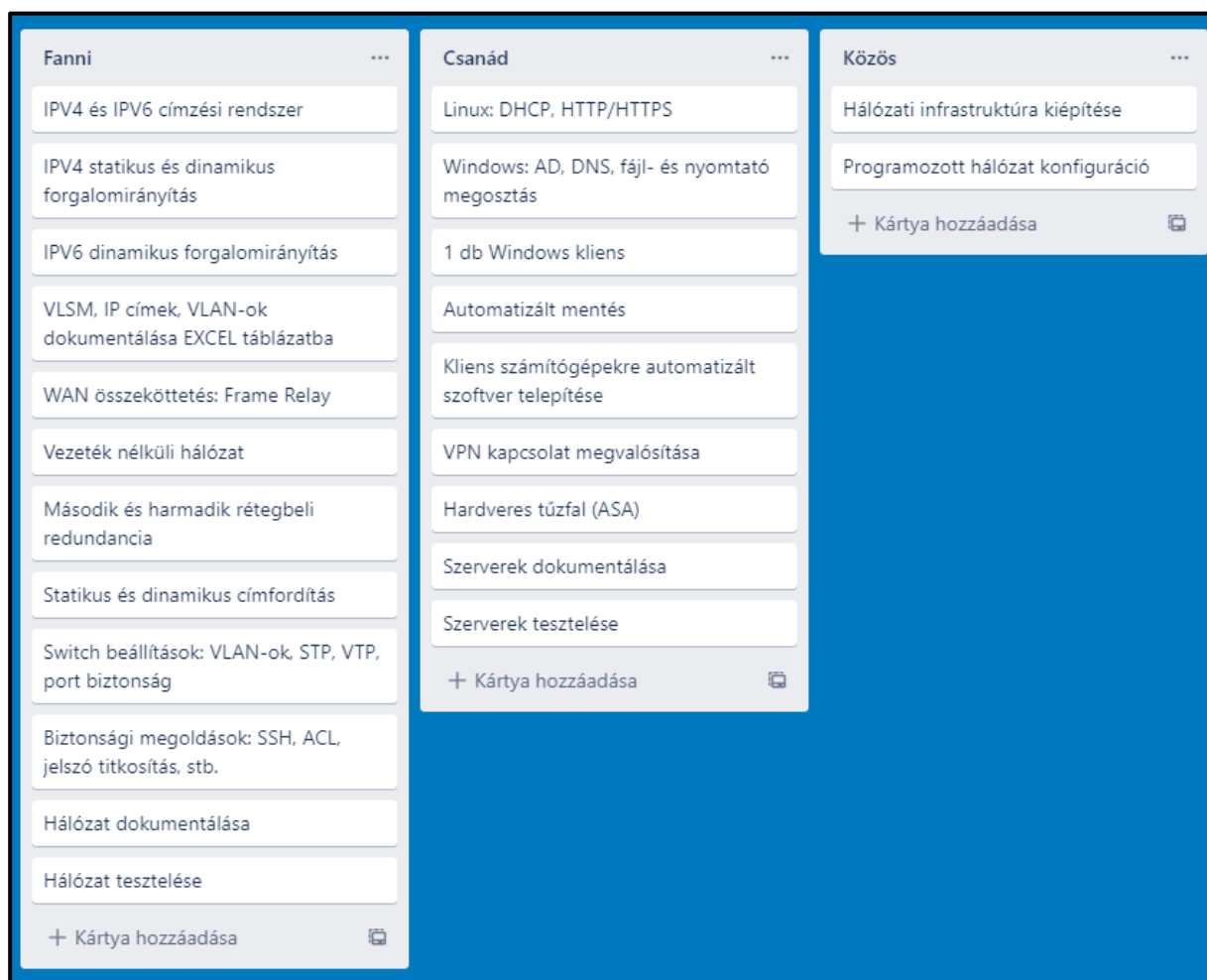
6.ábra

Aranytoll Könyvelő iroda

Az Aranytoll Könyvelő nevű cég megbízást adott számunkra, hogy készítsük el az általuk kívánt hálózatot, mivel kisebb-nagyobb változások voltak a cégüknél. A használt eszközök egy része már a cég tulajdonát képezte a másik részét pedig a rendelkezésünkre álló, szűkös pénzösszezből kellett kivitelezni. Vásárláskor kifejezetten odafigyeltünk, hogy a már meglévő eszközökkel kompatibilisek legyenek, illetve képesek legyenek a rajtuk kívánt technológiák megvalósításához. A szerverek kiválasztásakor legfőbb célkitűzésünk volt, hogy a megfelelő mennyiségű kliens kiszolgálására képesek legyenek.

Munkamegosztás

A kívánt igények elkészítéséhez a Trello-t (7.ábra) és a Google Drive-ot használtuk projektmenedzsmenti eszközként.



7.ábra

IP címek kiosztása

Statikus IPv4 címek kiosztása

DEBRECEN IRODA 192.168.1.0/24						
Részleg neve	Hostok száma	Hálózat címe	Prefix	Alhálózati maszk	Kiosztható címek	Üzenetszórási cím
Dolgozók	12	192.168.1.0	/28	255.255.255.240	192.168.1.1-192.168.1.14	192.168.1.15
Vezetőség	10	192.168.1.16	/28	255.255.255.240	192.168.1.17-192.168.1.30	192.168.1.31
Szerver	1	192.168.1.32	/29	255.255.255.248	192.168.1.33-192.168.1.38	192.168.1.39
SZEGED IRODA 192.168.2.0/24						
Részleg neve	Hostok száma	Hálózat címe	Prefix	Alhálózati maszk	Kiosztható címek	Üzenetszórási cím
Dolgozók	20	192.168.2.0	/27	255.255.255.224	192.168.2.1-192.168.2.30	192.168.2.31
Vezetőség	10	192.168.1.32	/28	255.255.255.240	192.168.1.33-192.168.1.46	192.168.1.47
BUDAPEST IRODA 192.168.3.0/24						
Részleg neve	Hostok száma	Hálózat címe	Prefix	Alhálózati maszk	Kiosztható címek	Üzenetszórási cím
Dolgozók	20	192.168.3.0	/27	255.255.255.224	192.168.3.1-192.168.3.30	192.168.3.31
Vezetőség	10	192.168.3.32	/28	255.255.255.240	192.168.3.33-192.168.3.46	192.168.3.47
Szerverek	2	192.168.3.48	/29	255.255.255.248	192.168.3.49-192.168.3.54	192.168.3.55

8.ábra

Ahogy a táblázatban is látható, a dolgozóknak, vezetésnek, illetve a szervereknek külön-külön alhálózatokat hoztunk létre (8.ábra). Ezáltal a címtereket hatékonyabban tudtuk kihasználni, rugalmasabban lehetett a hálózatot tervezni. Az alhálózatok mérete az egyes irodák alkalmazottai száma szerint lettek elkészítve, így takarékosabban bántunk az IPv4-es címekkel. Az alapértelmezett átjáró minden alhálózatban az első kiosztható IP cím. A nyomtatók statikusan kapták az ötödik kiosztható IP címet mindenhol. A számítógépek dinamikusan kapják a Linux szervertől az IP címeket, DHCP szolgáltatás segítségével. A Linux szerver a hálózata második kiosztható, a Windows szerver a harmadik kiosztható címet kapta statikusan.

Statikus IPv6 címek beállítása

```
Tavoli_Admin(config)#ipv6 unicast-routing
Tavoli_Admin(config)#int fa0/0
Tavoli_Admin(config-if)#ipv6 add 2001::1/64
Tavoli_Admin(config-if)#int se0/3/0
Tavoli_Admin(config-if)#ipv6 add 2005::2/64
```

9.ábra

IPv6-os címeket egyaránt használatban van a hálózatban, pontosabban az ISP és a Távoli adminisztrátor között. Az IPv6-os címekkel megnövekedett a címtartomány is, és a topológiához jobban illeszkedő címzést, illetve útvonalválasztásokat is képesek voltunk kialakítani (9.ábra).

Kapcsolók konfigurálása

VLAN-ok

```
SW_DEBDOLG(config)#vlan 10
SW_DEBDOLG(config-vlan)#name DEBDOLGOZOK
SW_DEBDOLG(config-vlan)#vlan 20
SW_DEBDOLG(config-vlan)#name DEBVEZETOSEG
SW_DEBDOLG(config-if)#interface vlan 10
SW_DEBDOLG(config-if)#interface vlan 20
SW_DEBDOLG(config-if)#int range fa0/2-5
SW_DEBDOLG(config-if-range)#switchport mode access
SW_DEBDOLG(config-if-range)#switchport access vlan 10
SW_DEBDOLG(config-if-range)#int fa0/1
SW_DEBDOLG(config-if)#switchport mode trunk
SW_DEBDOLG(config-if)#int fa0/6
SW_DEBDOLG(config-if)#switchport mode trunk
SW_DEBDOLG(config-if)#int range fa0/7-24
SW_DEBDOLG(config-if-range)#sh
SW_DEBDOLG(config-if-range)#int range g0/1-2
SW_DEBDOLG(config-if-range)#sh
```

10.ábra

Hat darab VLAN található a cég hálózatában. Telephelyenként kettő-kettő darabot valósítottunk meg, mind a dolgozók, mind a vezetés részére (10.ábra). A táblázatban láthatóak szerint a debreceni telephelyen a 10-es, illetve 20-as VLAN-t, a szegedin a 30-as és 40-es, a budapestin pedig az 50-es és 60-as VLAN-t használtuk (11.ábra). Mindezek mellett még megtekinthető a VLAN-ok hozzárendelése a kapcsolók megfelelő portjaihoz. Ezzel a kialakítással biztosítottuk a biztonságot, a terhelésmegosztást, illetve az adatszórás kezelését. Másodsorban minden használt porton lehetővé tettük a kézi beállítás módját, ezzel elkerülve, hogy dinamikus módba legyenek a portok, ahol nem tudjuk azokat személyre szabni. A kapcsolók között, illetve a forgalomirányítók felé trónk üzemmódba állítottuk a kapcsolókat. A nem használt portokat adminisztratíván lekapcsoltuk, ezzel is növelve a hálózat biztonságát.

VLAN id	VLAN név		Eszköz	VLAN 10	VLAN 20	VLAN 30	VLAN 40	VLAN 50	VLAN 60
10	DEBDOLGOZOK		SW_DEBDOLG	Fa0/2-5	-	-	-	-	-
20	DEBVEZETOSEG		SW_DEBVEZ	-	Fa0/1-3	-	-	-	-
30	SZEGDOLGOZOK		SW_SZEGDOLG	-	-	Fa0/2-5	-	-	-
40	SZEGVEZETOSEG		SW_SZEGVEZ	-	-	-	Fa0/1-3	-	-
50	BUDDOLGOZOK		SW_BPDLG	-	-	-	-	Fa0/2-5	-
60	BUDVEZETOSEG		SW_BPVEZ	-	-	-	-	-	Fa0/1-3

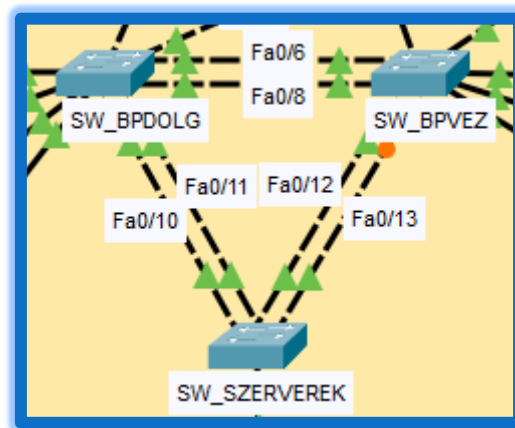
11.ábra

VTP

```
SW_BPDOLG(config)#vtp mode server  
SW_BPDOLG(config)#vtp domain BPVLAN  
SW_BPDOLG(config)#vtp password BPVLAN  
SW_BPDOLG(config)#vtp version 2
```

12.ábra

A hatékonyság növelése érdekében VTP-t konfiguráltunk a budapesti hálózatban (12.ábra, 13.ábra). Mivel ezen a telephelyen található a legtöbb kapcsoló így elengedhetetlennek találtuk a használatát, így a saját munkánkat tudtuk könnyebbé tenni, időt spórolva a hálózat konfigurálása és karbantartása során.



13.ábra

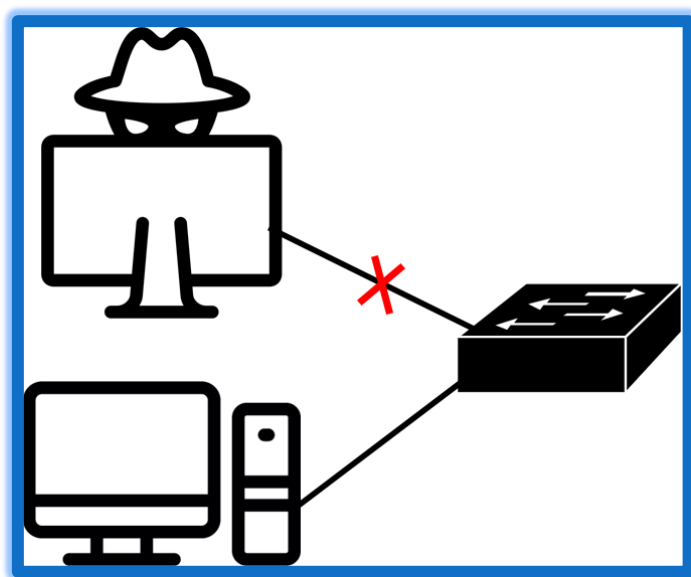
Az SW_BPDOLG kapcsolón konfiguráltuk a szerver üzemmódot, mivel a budapesti hálózatban ez a legkorszerűbb kapcsoló a többivel összevetve. Ebből adódóan ez a kapcsoló hirdeti a VLAN információkat a többi kapcsoló irányába, jelen esetben a SW_BPVEZ-nek és az SW_SZERVEREK-nek. Ezáltal ezen a kapcsolón van egyedül engedélyezve, hogy konfigurálhassuk, valamint módosíthassuk a meglévő VLAN-okat. Elegendő tárhelye van a ROM-ban, hogy ott tárolhassa a VTP információkat, illetve szükség esetén vissza tudja azokat állítani. A SW_BPVEZ, illetve SW_SZERVEREK kliens üzemmódba működnek. Ezeken is engedélyezve van a VTP, illetve a hirdetések tudják fogadni és küldeni is egyaránt, de nem tudják megváltoztatni a VLAN információkat. A VTP tartományban minden eszköz a 2-es verziót problémamentesen futtatja, mivel a használatban lévő VTP kliensek nem képesek a 3-as verzió futtatására.

Port biztonság

```
SW_BPDOLG(config)#int range fa0/2-5  
SW_BPDOLG(config-if-range)#switchport port-security  
SW_BPDOLG(config-if-range)#switchport port-security mac-address sticky  
SW_BPDOLG(config-if-range)#switchport port-security maximum 1  
SW_BPDOLG(config-if-range)#switchport port-security violation restrict
```

14.ábra

Az alapértelmezetten nyitott, védelem nélküli kapcsoló portokra különböző beállításokat konfiguráltunk biztonsági okokból (14.ábra). Mivel bizalmas információk haladnak át a hálózaton így fontosnak találtuk, hogy megfelelő támadáselhárítási megoldásokat biztosítsunk (15.ábra). A VLAN-ok konfigurálása során már hozzáférhetővé tettük a portokat, így könnyedén személyre tudtuk azokat szabni. Ezután a kívánt portokat portbiztonsági üzemmódba léptettük.



15.ábra

Abból adódóan, hogy a hálózatunkat fokozatosan, illetve rendszeresen ellenőrizzük, hogy véletlenül se legyen illetéktelen személyeknek hozzáférése a belső hálózatunkhoz, így a kapcsoló tanuló üzemmódban az elsőként csatlakozó gépnek az IP címét engedélyezi és ennek a MAC címét megjegyzi. Annak érdekében, hogy más címet akaratlanul se jegyezzen meg, kiadtuk, hogy maximálisan azt az egy címet tanulhassa meg. Portsértés esetén Syslog üzenetet küld, eldobja a csomagokat ezáltal nem továbbítja így a forgalmat, illetve növeli a büntetés számlálót, mindezek ellenére nem állítja le az adott portot.

STP

```
SW_BPDOLG(config)#spanning-tree mode rapid-pvst
SW_BPDOLG(config)#spanning-tree vlan 60 root primary
SW_BPDOLG(config)#spanning-tree vlan 50 root secondary
SW_BPDOLG(config)#int range fa0/2-5
SW_BPDOLG(config-if)#spanning-tree portfast
SW_BPDOLG(config-if)#spanning-tree bpduguard enable
SW_BPDOLG(config-if)#sh
SW_BPDOLG(config-if)#no sh
SW_BPDOLG(config-if)#spanning-tree guard root
SW_BPDOLG(config-if)#int fa0/10
SW_BPDOLG(config-if)#spanning-tree guard root
```

16.ábra

A második rétegbeli redundancia kialakításához STP protokollt használunk, ezzel növelve a hálózat megbízhatóságát, stabilitását. Az alternatív útvonalak okozta problémák elkerülése érdekében használjuk a feszítőfa algoritmust, így nem kell attól tartanunk, hogy esetleges hurkok alakuljanak ki a hálózatunkba. A STP konvergálása időigényes, ezért azt figyelembe véve, hogy addig semmilyen információt (köztük az IP címinformációkat sem a Linux szerveren futó DHCP szolgáltatástól) nem kapják meg, így a PortFast üzemmódra esett a választásunk. Amíg a szóban forgó eszközök nem kapják meg a BPDU üzenetet addig ebben az üzemmódban vannak, viszont a BPDU üzenet érkezésével kikapcsolja ezt a funkciót a porton és megkezd a konvergálás folyamatát. Ennek a működéséhez a BPDU Guard funkciót alkalmaztuk. Az SW_BPDOLG lett az elsődleges gyökérponti híd. Abból adódóan, hogy fokozottan odafigyeltünk minden támadási lehetőségre, a root guard funkcióval megakadályoztuk, hogy az általunk kinevezett gyökérponti porton kívül más port ne válhasson azzá. Ez azért volt fontos számunkra, mert egy végberendezési eszköz által használt portra csatlakoztatna valaki egy kapcsolót, akkor van esély rá, hogy átvegye az általunk kinevezett gyökérponti kapcsolónk helyét (16.ábra).

Forgalomirányítás

IPv4 statikus forgalomirányítás

```
ISP(config)#ip route 0.0.0.0 0.0.0.0 10.10.30.2  
ISP(config)#ip route 0.0.0.0 0.0.0.0 10.10.20.2  
ISP(config)#ip route 0.0.0.0 0.0.0.0 10.10.10.2
```

17.ábra

A routerek IP címét statikusan adtuk meg. A megfelelő VLAN-ok az első kiosztható címét kapták meg a hálózatnak, mivel router-on-stick megoldást alkalmaztunk. A távoli munkás, illetve adminisztrátor a hálózata első kiosztható címét kapta. A távoli munkás, a távoli adminisztrátor, illetve a Budapest_Iroda2-nél statikusan lett megoldva a forgalomirányítás, illetve ugyan az ISP konfigurálása nem a mi hatáskörünk, de itt is statikusan oldottuk meg a szimulálás működésének érdekében (17.ábra).

```
Budapest_Iroda2(config)#ip route 0.0.0.0 0.0.0.0 10.10.10.1
```

18.ábra

A Budapest_Iroda2-nél alapértelmezett forgalomirányítást alkalmaztunk az ISP felé, illetve ugyanígy is lát vissza az összes környező hálózatába az ISP (18.ábra).

IPv4 dinamikus forgalomirányítás

```
Debrecen_Iroda(config)#router ospf 1
Debrecen_Iroda(config-router)#router-id 1.1.1.1
Debrecen_Iroda(config-router)#network 192.168.1.0 0.0.0.15 area 0
Debrecen_Iroda(config-router)#network 192.168.1.16 0.0.0.15 area 0
Debrecen_Iroda(config-router)#network 192.168.1.32 0.0.0.7 area 0
Debrecen_Iroda(config-router)#network 172.20.2.0 0.0.0.3 area 0
Debrecen_Iroda(config-router)#network 172.20.3.0 0.0.0.3 area 0
Debrecen_Iroda(config-router)#network 172.20.4.0 0.0.0.3 area 0
Debrecen_Iroda(config-router)#passive-interface fa1/0
Debrecen_Iroda(config-router)#passive-interface fa1/1
```

19.ábra

A telephelyek között dinamikus IPv4-es forgalomirányítást valósítottunk meg a kapcsolatállapot alapú OSPF segítségével (19.ábra). Az ok, amiért ezt választottuk, a gyorsabb konvergencia, illetve támogatja a VLSM-et is, ami számunkra elengedhetetlen feltétel volt. A folyamatazonosítója a forgalomirányításnak 1, minden routernek lett egy-egy router-idja. Egyterületű OSPF-et használtunk. Azokat a portokat, ahol nem volt szükség útvonal hirdetésekre, mivel PC-k, illetve egyéb eszközök vannak, passzív üzemmódba tettük. A forgalomirányításba belevontunk a Frame Relay érdekében elkészített virtuális hálózatokat is.

```
Debrecen_Iroda(config)#int se0/3/0.102
Debrecen_Iroda(config-subif)#ip ospf authentication message-digest
Debrecen_Iroda(config-subif)#ip ospf message-digest-key 1 md5 ospfauth
```

20.ábra

A forgalomirányítás biztonságossá tételéhez MD5-ös hitelesítést alkalmaztunk. Az OSPF MD5-ös hitelesítése biztonságosabb, mint az egyszerű szöveges hitelesítés (20.ábra). Ez a módszer az MD5-ös algoritmust használja, ami egy hash értéket számol ki egy OSPF csomagból és egy jelszóból. Ez a hash érték a csomagban továbbítódik és a fogadó fél, aki tudja ugyanezt a jelszót, képes kiszámolni a saját hash értékével.

```
Budapest_Iroda2(config-router)#default-information originate
```

21.ábra

Budapest_Iroda2 hirdeti az alapértelmezett útvonalat az ISP felé (21.ábra). Amennyiben ismeretlen címre kell küldenie egy forgalomirányítónak a csomagot, mindig ennek a forgalomirányítónak küldi el továbbításra.

IPv6 dinamikus forgalomirányítás

```
Tavoli_Admin(config)#ipv6 router ospf 1
Tavoli_Admin(config-router)#router-id 5.5.5.5
Tavoli_Admin(config-router)#int fa0/0
Tavoli_Admin(config-if)#ipv6 enable
Tavoli_Admin(config-if)#ipv6 ospf 1 area 0
Tavoli_Admin(config-if)#int se0/3/0
Tavoli_Admin(config-if)#ipv6 enable
Tavoli_Admin(config-if)#ipv6 ospf 1 area 0
```

22.ábra

Ahhoz, hogy a forgalomirányítás is működhessen a két Ipv6-os címkészletet használó eszköz között, OSPFv3 protokollt alkalmaztunk (22.ábra). A felhasználási indokunk, mint az IPv4-es címeknél is, a könnyen méretezhetőség, a VLSM támogatása és a gyorsabb konvergencia. A távoli adminisztrátor az első kiosztható címét kapta a hálózatából. Itt szintén az 1-es folyamatazonosítót használtuk és egyterületű dinamikus forgalomirányításról van szó.

Alkalmazott WAN technológia

Frame Relay

```
Debrecen_Iroda(config)#int se0/3/0
Debrecen_Iroda(config-if)#encap frame-relay
Debrecen_Iroda(config-if)#int se0/3/0.102 point-to-point
Debrecen_Iroda(config-subif)#frame-relay interface-dlci 102
Debrecen_Iroda(config-subif)#ip add 172.20.2.1 255.255.255.252
Debrecen_Iroda(config-subif)#int se0/3/0.103 point-to-point
Debrecen_Iroda(config-subif)#frame-relay interface-dlci 103
Debrecen_Iroda(config-subif)#ip add 172.20.3.1 255.255.255.252
Debrecen_Iroda(config-subif)#int se0/3/0.104 point-to-point
Debrecen_Iroda(config-subif)#frame-relay interface-dlci 104
Debrecen_Iroda(config-subif)#ip add 172.20.4.1 255.255.255.252
```

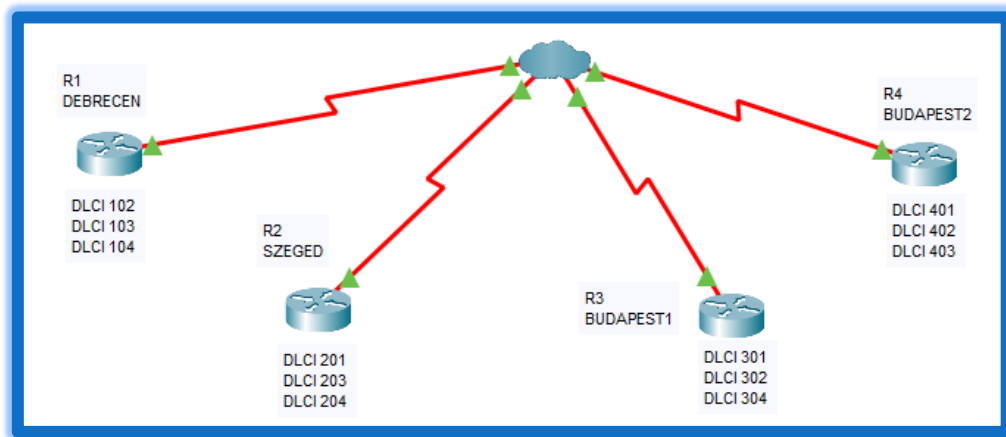
23.ábra

A Frame Relay technológia már évtizedekkel ezelőtt ki lett építve a cég telephelyeinek számára és nem lett visszabontva. A vezetőség kifejezetten ragaszkodott, hogy megmaradjon ez a kapcsolat így ehhez alkalmazkodva készítettük el a WAN kapcsolatokat. A kommunikációhoz a már meglévő virtuális áramköröket alkalmaztuk (23.ábra). Ahhoz, hogy a forgalomirányító több másik hasonló eszközt elérhessen, címeket hoztunk létre, melyek a globálisan ugyan nem, de lokálisan egyedi DLCI-k. Számozásukkor kifejezetten odafigyeltünk, hogy leegyszerűsített, könnyen érthető számokat hozzunk létre, így az első számjegy mindig a forrás router számát, az öt követő két szám (esetünkben csak az utolsó, mivel nincsen 9-nél több routerünk) a cél routerünk számát jelöli.

	From Port	Sublink	To Port	Sublink
1	Serial0	R1-R3	Serial2	R3-R1
2	Serial0	R1-R4	Serial3	R4-R1
3	Serial1	R2-R3	Serial2	R3-R2
4	Serial1	R2-R4	Serial3	R4-R2
5	Serial2	R3-R4	Serial3	R4-R3
6	Serial0	R1-R2	Serial1	R2-R1

24.ábra

A fenti példán így tökéletesen és egyszerűen kiolvasható, hogy ez az egyes számú routerből a kettes számú routerhez vezető virtuális utat jelöli (24.ábra). Ugyanez az útvonal a kettes számú routeren a „201”-es DLCI-t jelenti. A virtuális áramkörökhöz természetesen egyenként rendeltünk 1-1 IP címet, hogy létrejöhessen a kettejük között a kommunikáció, mint olyan.



25.ábra

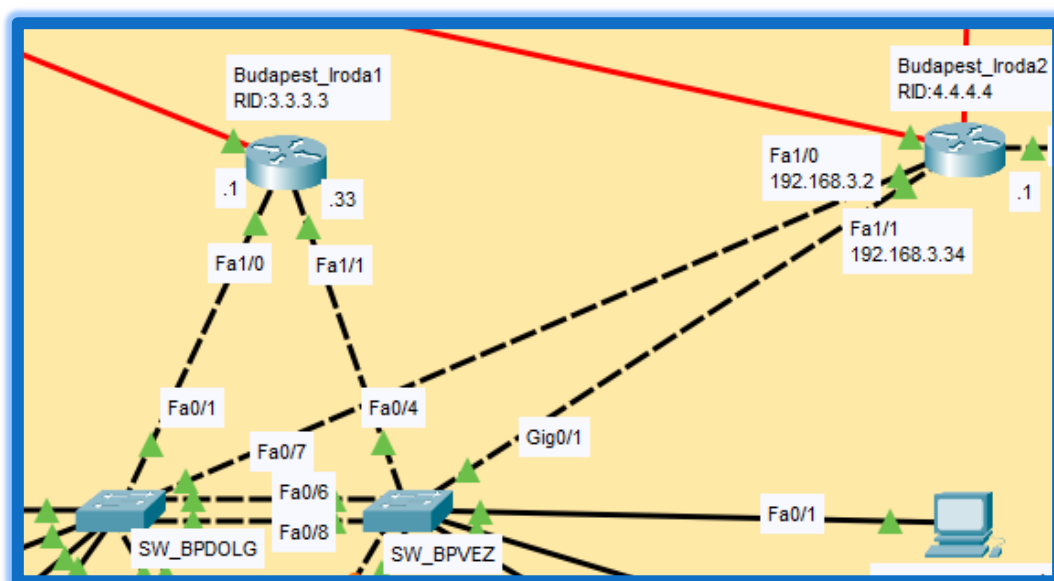
A fenti példán részletesebben megtekinthető az egyes virtuális áramkörök DLCI-nek a számozása (25.ábra).

HSRP

```
Budapest_Iroda(config)#int fa1/0.50  
Budapest_Iroda(config-subif)#standby 1 ip 192.168.3.3  
Budapest_Iroda(config-subif)#standby 1 priority 150  
Budapest_Iroda(config-subif)#standby 1 preempt
```

26.ábra

A harmadik rétegbeli redundanciát a HSRP segítségével valósítottuk meg, ezzel is egy biztonságosabb, hibátűrő IP hálózatot voltunk képesek létre hozni a cég számára (27.ábra). A működése érdekében létrehoztunk egy virtuális routert a fizikai routerek konfigurálása felett. Ezután a megfelelő VLAN interfészekre kiadtuk a parancsokat. Elsősorban engedélyeztük a HSRP működését, majd a HSRP csoport azonosítóját megadva, a virtuális routernek adunk egy IP címet. Ezután a prioritási szintjét állítottuk be, aminek adott esetben az értéke 150, az alapértelmezett 100-tól eltérően. A használt ábrán az aktív routert konfiguráltuk, szóval a HSRP csoportban ennek lesz a legnagyobb prioritása. Utolsó lépésként engedélyeztük, hogy az adott router aktív legyen, ha nagyobb a prioritása, mint a HSRP csoporton belül a többi routeré (26.ábra). Amennyiben kiesne az aktív routerünk, akkor alternatív útvonalat biztosítva egy másik router válik aktívvá. A mi esetünkben, eszközök szempontjából azt jelenti, hogy a Budapest_Iroda1 az aktív router és Budapest_Iroda2 lesz aktív, ha kiesne az első számú routerünk valamilyen hiba hatására. A másodlagos HSRP routerünk a 100-as prioritást kapta meg értékül. Több eszköz nem tartozik bele az általunk létrehozott HSRP csoportba.



27.ábra

Alapvető biztonsági megoldások

```
Debrecen_Iroda(config)#enable password debpass.  
Debrecen_Iroda(config)#no ip domain-lookup  
Debrecen_Iroda(config)#login block-for 600 attempts 5 within 60  
Debrecen_Iroda(config)#security passwords min-length 8  
Debrecen_Iroda(config)#username debadmin secret Admin123deb  
Debrecen_Iroda(config)#banner motd "Jogosulatlan belepés tilos!"  
Debrecen_Iroda(config)#service password-encryption  
Debrecen_Iroda(config)#ip ssh version 2  
Debrecen_Iroda(config)#ip domain-name konyveles.hu  
Debrecen_Iroda(config)#crypto key generate rsa (1024)  
Debrecen_Iroda(config)#line vty 0 15  
Debrecen_Iroda(config-line)#exec-timeout 5  
Debrecen_Iroda(config-line)#transport input ssh  
Debrecen_Iroda(config-line)#login local  
Debrecen_Iroda(config-line)#line con 0  
Debrecen_Iroda(config-line)#password debcons  
Debrecen_Iroda(config-line)#login
```

28.ábra

Az elsődleges megoldások például a domain név feloldásának tiltása, a jelszavak titkosítása minden konfigurációs módban, illetve a már kiadott jelszavakra vonatkozóan is, a MOTD banner megadása, hogy felhívhassuk az illetéktelen behatolókat a cselekedetük jogi következményeire, a minimum jelszó hosszának beállítása, ezzel is nehezebben feltörhetővé téve azokat. Felkészültünk arra is, hogy többszörösen rossz jelszóval próbálják meg támadni az adott routernket. Amennyiben 1 percen belül ötször próbálnak bejelentkezni hibásan, abban az esetben 10 percre lesz letiltva az adott eszközhöz való csatlakozás. Amennyiben tévesnek feltételezzük ezt a lehetőséget, úgy könnyedén vissza tudjuk állítani az alapértelmezett helyzetébe a forgalomirányítónkat. Létrehoztunk egy admin felhasználót, titkosított jelszóval párosítva. SSH kapcsolatot alakítottunk ki, a megbízható, titkosított távoli elérés érdekében. A Cisco IOS-on használható legfrissebb 2-es verziót futtatjuk rajta. Csak ezzel a verzióval érhető el távolról, ezzel kapcsolódhatunk rá. Ahhoz, hogy működjön, beállítottuk a tartománynevet, amely számunkra az aranytollkonyveles.hu jelenti. Ezután elkészítettük a 1024 bites RSA kulcsot. Végző lépésként, a sávos elérés biztonságossá tételéhez beállítottuk, hogy 5 perc inaktivitás után automatikusan szakítsa meg a kapcsolatot. A konzoljelszó beállításra került (28.ábra).

Címfordítás

A legfontosabb biztonsági beállításnak gondoljuk, hogy a belső privát címek ne kerüljenek ki a hálózatunkból. Ennek érdekében hasznosítottuk a NAT-ot, hogy elvégezhessük a címfordításokat, így az internetszolgáltatótól kapott privát címek helyett publikus címekkel kommunikálhatnak az eszközök más hálózatbeli eszközökkel.

Dinamikus címfordítás

```
Budapest_Iroda2(config)#access-list 1 permit 172.20.4.0 0.0.0.255
Budapest_Iroda2(config)#access-list 1 permit 172.20.11.0 0.0.0.255
Budapest_Iroda2(config)#access-list 1 permit 172.20.15.0 0.0.0.255
Budapest_Iroda2(config)#access-list 1 permit 192.168.1.0 0.0.0.255
Budapest_Iroda2(config)#access-list 1 permit 192.168.2.0 0.0.0.255
Budapest_Iroda2(config)#access-list 1 permit 192.168.3.0 0.0.0.255
Budapest_Iroda2(config)#access-list 1 permit 192.168.4.0 0.0.0.255
Budapest_Iroda2(config-subif)#int serial 0/3/0.401
Budapest_Iroda2(config-subif)#ip nat inside
Budapest_Iroda2(config-subif)#int serial 0/3/0.402
Budapest_Iroda2(config-subif)#ip nat inside
Budapest_Iroda2(config-subif)#int serial 0/3/0.403
Budapest_Iroda2(config-subif)#ip nat inside
Budapest_Iroda2(config-subif)#int se0/2/0
Budapest_Iroda2(config-if)#ip nat outside
Budapest_Iroda2(config)#ip nat pool KONYVELESPool 155.155.1.1 155.155.1.253 netmask 255.255.255.0
Budapest_Iroda2(config)#ip nat inside source list 1 pool KONYVELESPool
Budapest_Iroda2(config)#access-list 2 permit 10.10.10.0 0.0.0.3
Budapest_Iroda2(config)#access-list 2 permit 10.10.20.0 0.0.0.3
Budapest_Iroda2(config)#access-list 2 permit 10.10.30.0 0.0.0.3
Budapest_Iroda2(config)#access-list 2 permit host 120.120.1.1
Budapest_Iroda2(config)#access-list 2 permit 192.168.55.0 0.0.0.255
Budapest_Iroda2(config)#int se0/2/0
Budapest_Iroda2(config-if)#ip access-group 2 in
```

29.ábra

Elsősorban létrehoztuk az ACL listákat, hogy meghatározzuk mely hálózat címeit szeretnénk majd lefordítani. Második lépésként meghatároztuk a belső, illetve külső oldalhoz tartozó interfészeket. Ezután létrehoztunk egy pool-t, KONYVELESPool néven és megadtuk, hogy melyik IP cím tartományból használjon címet, fordítás során. Ebből a tartományból kerül ki a publikus cím, ezt fogják a külvilágról látni majd. Miután ezzel végeztünk, összerendeljük az ACL listát és a pool-t (29.ábra).

Statikus címfordítás

```
Tavoli_Admin(config)#int se0/3/0  
Tavoli_Admin(config-if)#ip nat outside  
Tavoli_Admin(config-if)#int fa0/0  
Tavoli_Admin(config-if)#ip nat inside  
Tavoli_Admin(config)#ip nat inside source static 192.168.56.2 120.120.1.1
```

30.ábra

Statikus címfordítás során is elengedhetetlen első lépésnek számít, hogy megjelöljük a belső, illetve külső oldalhoz tartozó interfészeket, viszont itt nem szükséges létrehozni sem hozzáférési listát, sem pedig pool-t. Egyszerűen, manuálisan begépeljük, hogy melyik címet, milyen címre szeretnénk fordítani (30.ábra).

VPN

```
Home_Router(config)#crypto isakmp policy 1
Home_Router(config-isakmp)#authentication pre-share
Home_Router(config-isakmp)#exit
Home_Router(config)#crypto isakmp key KONYVKULCS address 10.10.10.2
Home_Router(config)#crypto ipsec transform-set KONYVPNSET1 esp-aes esp-sha-hmac
Home_Router(config)#ip access-list extended HOMEVPN
Home_Router(config-ext-nacl)#permit ip 192.168.55.0 0.0.0.255 192.168.3.0 0.0.0.31
Home_Router(config-ext-nacl)#exit
Home_Router(config)#crypto map 1BP2 1 ipsec-isakmp
Home_Router(config-crypto-map)#set peer 10.10.10.2
Home_Router(config-crypto-map)#set transform-set KONYVPNSET1
Home_Router(config-crypto-map)#match address HOMEVPN
Home_Router(config-crypto-map)#exit
Home_Router(config)#int se0/3/0
Home_Router(config-if)#crypto map 1BP2
```

31.ábra

A home office-ből dolgozó alkalmazottaknak is biztosítottunk egy biztonságosabb kapcsolatot. Ezért egy virtuális magánhálózatot hoztunk létre, hogy az internetszolgáltató által kiosztott eredeti privát IP címeik ne kerüljenek ki a hálózaton kívülre, elrejtethjük azokat az internet használata közben (31.ábra). Elsősorban a VPN hozzáféréseken keresztül szeretnénk elérni egy zárt hálózatot, de ahhoz, hogy ez tökéletesen és biztonságosan működhessen, konfiguráltunk egy isakmp házirendet, amelynek az azonosítója 1-esre lett állítva. A hitelesítése előre megosztott kulccsal zajlik, amiben a KONYVKULCS értéket adtuk meg. Következő lépésként beállítottuk az IPSec transzformációt, az esp-aes titkosítási módszerrel és az esp-sha-hmac hitelesítéssel. Létrehoztuk a HOMEVPN kiterjesztett ACL listát, amelyben beállítottuk, hogy csak a VPN-en keresztül lehessen elérni a belső hálózatot. Miután létrehoztuk a BP2 nevezetű crypto map-et, beállítottuk a kapcsolat másik végét és hozzárendeltük a KONYVPNSET nevezetű transzformációs készletünket. A kimenő interfészhez hozzárendeltük az elkészített crypto-map beállításainkat.

Hardveres tűzfal

```
ciscoasa(config)#hostname BACKUP-ASA
BACKUP-ASA(config)#enable password backuppass.
BACKUP-ASA(config)#interface GigabitEthernet1/1
BACKUP-ASA(config-if)#nameif outside
BACKUP-ASA(config-if)#security-level 0
BACKUP-ASA(config-if)#ip address 192.168.1.19 255.255.255.240
BACKUP-ASA(config-if)#interface GigabitEthernet1/2
BACKUP-ASA(config-if)#nameif inside
BACKUP-ASA(config-if)#security-level 100
BACKUP-ASA(config-if)#ip address 192.168.1.33 255.255.255.248
BACKUP-ASA(config)#route outside 0.0.0.0 0.0.0.0 192.168.1.18
BACKUP-ASA(config)#access-list 100 permit tcp any any eq ftp
BACKUP-ASA(config)#access-list 100 permit ip host 192.168.1.25 any
BACKUP-ASA(config)#access-list 100 permit ip host 192.168.2.40 any
BACKUP-ASA(config)#access-list 100 permit ip host 192.168.3.40 any
BACKUP-ASA(config)#access-list 100 permit ip host 120.120.1.1 any
BACKUP-ASA(config)#int gig1/1
BACKUP-ASA(config-if)#access-group 100 in interface outside
```

32.ábra

A debreceni telephelyünkön található biztonsági mentésekkel rendelkező szerverünket egy ASA-val védtünk le. Ezen a hardveres tűzfalon csak és kizárólag a rendszergazdáknak van teljes jogosultsága áthaladni, ezáltal is fokoztuk a biztonság tényezőjét. A telephelyek között az egyetlen szolgáltatás, amit elérhetnek az alkalmazottak, az nem más, mint az FTP. Természetesen ennek a szolgáltatásnak az engedélyezésére azért volt szükség, hogy a dolgozók képesek legyenek fájlokat feltölteni, illetve letölteni (32.ábra).

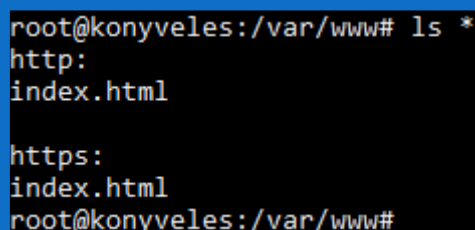
Szerverek és felhőszolgáltatások

A cég tulajdonosai a szerverekkel kapcsolatosan előírták, hogy legyen egy Linux, illetve egy Windows szerver a hálózatukban, viszont azt nem szabták meg, hogy melyik szerveren milyen szolgáltatást konfiguráljunk.

Linux Ubuntu

Kényelmi szempontokból arra esett a választásunk, hogy a Linux operációs rendszer futtató szerverünkön HTTP/HTTPS, illetve DHCP szolgáltatást működtessünk.

HTTP/HTTPS

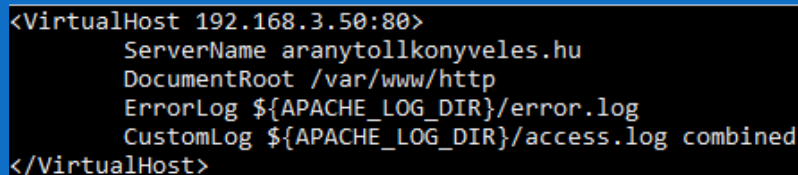


```
root@konyveles:/var/www# ls *
http:
index.html

https:
index.html
root@konyveles:/var/www#
```

33.ábra

A webes kommunikáció létrehozásához HTTP protokollt használtunk. A már meglévő kommunikáció titkosítása és hitelesítése érdekében HTTPS kapcsolatot is létesítettünk. A HTTP/HTTPS weboldalak különböző elérési útjait a fent látható módon oldottuk meg (33.ábra). Annak érdekében, hogy képesek legyünk probléma esetén azonnal reagálni és elhárítani azokat, az oldalak mappákban való elhelyezését egyértelmű, könnyen kezelhető és értelmezhető módon rendszereztük.



```
<VirtualHost 192.168.3.50:80>
    ServerName aranytollkonyveles.hu
    DocumentRoot /var/www/http
    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

34.ábra

A titkosítatlan HTTP kérések céljából készült alapvető konfigurációkat a fenti ábrán látható módon valósítottuk meg (34.ábra).

```
<VirtualHost 192.168.3.50:443>
  ServerName aranytollkonyveles.hu
  DocumentRoot /var/www/https

  ErrorLog ${APACHE_LOG_DIR}/error.log
  CustomLog ${APACHE_LOG_DIR}/access.log combined
  SSLEngine on
  SSLCertificateFile /home/konyvadmin/cert/aranytollkonyveles.hu.crt
  SSLCertificateKeyFile /home/konyvadmin/cert/aranytollkonyveles.hu.key
</VirtualHost>
```

35.ábra

A titkosított weboldal lekérések konfigurációja a fent látható ábra alapján lettek kivitelezve (35.ábra). Fontosnak tartottuk, hogy a webes kommunikáció titkosítva és hitelesítve legyen.

```
sudo openssl genrsa -des3 -out aranytollkonyveles.hu.key 2048
```

36.ábra

A titkosítást egy SSL tanúsítvány segítségével vittük végbe, ezáltal egy saját aláírással ellátott tanúsítvány tudtunk létrehozni és használni. A privát kulcsunkat egy 2048 bites, jelszóval védett formában valósítottuk meg (36.ábra).

```
sudo openssl req -key aranytollkonyveles.hu.key -new -out aranytollkonyveles.hu.csr
```

37.ábra

Mindezek után, szükségünk volt egy tanúsítvány aláírás kérelemre, amit a fent látható paranccsal hajtottuk végre (37.ábra). A létrehozott privát kulcsunkból készítettünk egy kérelmet, amit később felhasználtunk a teljes aláíráshoz. A parancs kiadása után néhány információt kért tőlünk a szerver, amire az alábbi módon válaszoltunk (38.ábra).

```
Country Name (2 letter code) [AU]:HU
State or Province Name (full name) [Some-State]:Pest
Locality Name (eg, city) []:Budapest
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Könyvelő iroda
Organizational Unit Name (eg, section) []:Aranytoll Könyvelés
Common Name (e.g. server FQDN or YOUR name) []:aranytollkonyveles.hu
Email Address []:info@aranytollkonyveles.hu
```

38.ábra

A kért információk megadása után létrehoztuk az SSL tanúsítványunkat. Az alábbi ábrán látható, hogy a lejáratí ideje 365 napra lett állítva (39.ábra).

```
sudo openssl req -key aranytollkonyveles.hu.key -new -x509 -days 365 -out aranytollkonyveles.hu.crt
```

39.ábra

Az elkészített kulcsunk az alábbi formátumban jelenik meg. Mint látható, ez egy emberi szem számára nem kiolvasható formátum, mert ezzel is növekszik a privát kulcs készítésének biztonsága (40.ábra).

```
Certificate:
Data:
  Version: 3 (0x2)
  Serial Number:
    78:a4:27:f1:a4:a0:85:04:54:6e:00:3b:c3:ab:28:19:86:6f:0b:85
  Signature Algorithm: sha256WithRSAEncryption
  Issuer: C = HU, ST = Pest, L = Budapest, O = K\C3\83\C2\B6nyvel\C3\85\C2\91 iroda, OU = Aranytoll K\C3\83\C2\B6nyvel\C3\83\C2\A9s, CN = aranytollkonyveles.hu, emailAddress = info@aranytollkonyveles.hu
  Validity
    Not Before: Apr 23 11:04:30 2022 GMT
    Not After : Apr 23 11:04:30 2023 GMT
  Subject: C = HU, ST = Pest, L = Budapest, O = K\C3\83\C2\B6nyvel\C3\85\C2\91 iroda, OU = Aranytoll K\C3\83\C2\B6nyvel\C3\83\C2\A9s, CN = aranytollkonyveles.hu, emailAddress = info@aranytollkonyveles.hu
  Subject Public Key Info:
    Public Key Algorithm: rsaEncryption
    RSA Public-Key: (2048 bit)
    Modulus:
      00:b3:4c:3c:79:fa:cd:4f:49:a2:cd:3f:95:52:42:
      3d:71:5d:f2:cc:fd:10:cc:f7:6e:99:2e:6f:33:db:
      cd:ea:4b:5f:e3:7d:4a:64:0f:47:fe:96:01:5a:2f:
      9f:dc:5f:29:f3:00:8a:f2:db:9f:db:db:07:e3:a1:
      9a:30:74:98:7b:c1:57:d0:fa:10:52:f3:f4:ee:e5:
      00:fb:2c:e9:1d:23:eb:28:de:9d:de:58:ed:8d:fe:
      ba:b6:e9:c8:23:9b:54:a5:39:51:f4:e6:fb:48:4d:
      77:97:85:b2:a5:8e:62:01:f4:dd:06:0c:57:62:70:
      c3:71:75:f4:8f:56:b3:81:c2:b5:f5:f4:0f:5f:72:
      51:e3:9d:59:48:ec:5f:2f:76:58:74:b4:80:71:0b:
      03:36:24:0e:62:ce:5b:03:94:e6:54:a4:9e:4e:67:
      8b:cd:46:37:82:0d:a0:ee:a0:d0:0c:0f:6c:d3:62:
      ed:f9:3c:cc:4b:c9:68:20:64:2c:6f:4a:11:ee:aa:
      89:11:d6:94:64:91:23:02:e9:46:1a:ef:6c:40:51:
      78:96:e2:88:e5:bf:c3:9b:47:ff:5c:9a:d6:6b:16:
      e4:90:65:9e:c2:2b:bd:46:da:d9:19:ce:86:9e:c8:
      57:32:70:b5:2b:91:84:57:e0:a6:5c:62:0f:dd:38:
      d7:a3
    Exponent: 65537 (0x10001)
  X509v3 extensions:
    X509v3 Subject Key Identifier:
      A1:A2:E1:E1:60:F5:AF:11:9E:8B:2C:54:AC:BA:01:82:81:87:BC:A9
    X509v3 Authority Key Identifier:
      keyid:A1:A2:E1:E1:60:F5:AF:11:9E:8B:2C:54:AC:BA:01:82:81:87:BC:A9
    X509v3 Basic Constraints: critical
      CA:TRUE
  Signature Algorithm: sha256WithRSAEncryption
    6f:a1:78:94:75:c7:6d:5f:76:33:81:10:a1:26:ea:46:4f:1b:
    c3:56:f0:f3:7d:7a:e6:3d:cb:d7:61:b0:02:02:de:fe:37:e1:
    cb:51:e4:fa:c6:be:d9:a1:2d:47:75:4c:3a:f2:a5:68:51:49:
    32:78:40:9d:26:38:78:56:c6:fa:eb:b2:bc:08:0a:61:bd:21:
    06:81:4a:3d:d1:ce:91:17:24:ab:ac:f0:7d:b4:09:e8:4e:96:
    aa:68:58:2d:83:9f:4e:12:1e:47:bb:59:33:af:37:5f:93:d2:
    57:1c:f0:23:04:9a:8e:d8:7b:79:b8:ff:11:90:83:1f:e8:b6:
    da:37:1f:03:23:14:23:65:c5:a3:bb:03:df:ec:0b:e4:17:a9:
    f6:76:dc:d5:04:ec:c2:bc:a5:b0:90:46:bc:39:05:c0:d3:16:
    50:71:5e:96:a8:b4:66:94:38:d5:ff:f7:3a:67:a1:51:7e:a4:
    2d:b5:f5:15:1d:93:76:9f:da:aa:da:d5:85:be:e5:3b:5f:aa:
    29:f5:04:19:76:82:24:66:fe:a1:d0:3e:02:c4:04:0a:b1:4e:
    85:c3:cc:66:28:b6:cb:12:62:dc:83:d3:85:ec:7d:b1:d4:fc:
    6d:0a:a2:2c:f2:14:c2:f1:f6:08:da:b7:24:3c:f4:de:62:07:
    fb:a0:58:ea
```

40.ábra

DHCP

Ahogy az „IP címek kiosztása” résznél már említettük, a Linux szerver felel a hálózati eszközök dinamikus címezéséért. A DHCP szolgáltatást az egész cég igénybe veszi, ezért több alhálózatot is konfiguráltunk, hogy a címtereket hatékonyabban, illetve takarékosabban tudjuk kihasználni és nem utolsósorban, elkerülhessük a IP cím duplikációkat. Az ábra tanulmányozása során kivehető, hogy a budapesti irodák DHCP konfigurációját jelenítettük meg, ahol megvalósítottuk a harmadik rétegbeli redundancia (HSRP) jelenlétét. Ennek okán lett mindkettő telephelyen a virtuális router IPv4 címe megadva, amely az adott irodáknál a harmadik kiosztható IP cím. A kiosztott DNS server címe a Windows operációs rendszert futtató szerverünknél található meg, ugyanis ott valósítottuk meg ezt a szolgáltatást. A kliensek a névfeloldásért felelős szervert, a 192.168.3.51-es címen érhetik el (41.ábra).

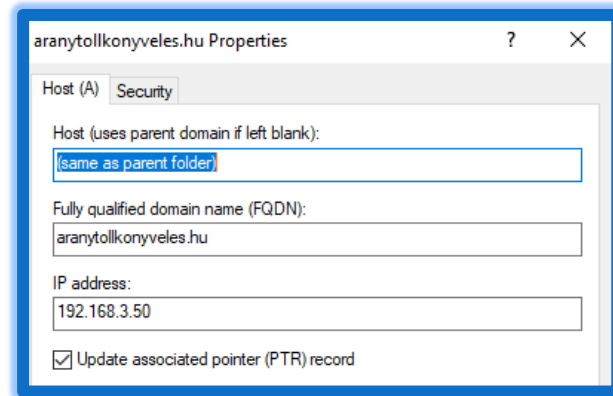
```
subnet 192.168.3.0 netmask 255.255.255.224 {
    option routers                192.168.3.3;
    option subnet-mask            255.255.255.224;
    option domain-search          "aranytollkonyveles.hu";
    option domain-name-servers    192.168.3.51;
    range 192.168.3.4 192.168.3.30;
}
subnet 192.168.3.32 netmask 255.255.255.240 {
    option routers                192.168.3.35;
    option subnet-mask            255.255.255.240;
    option domain-search          "aranytollkonyveles.hu";
    option domain-name-servers    192.168.3.51;
    range 192.168.3.36 192.168.3.46;
}
```

41.ábra

Windows 2019

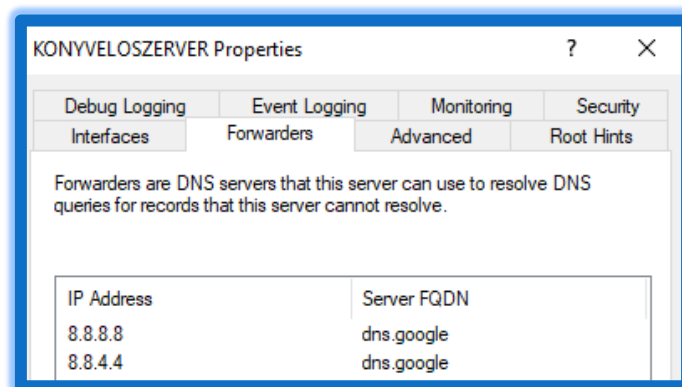
A Windows operációs rendszert futtató szerverünkön a DNS-t, az AD-t, az automatizált mentést, illetve a fájl- és nyomtató megosztást valósítottuk meg.

DNS



42.ábra

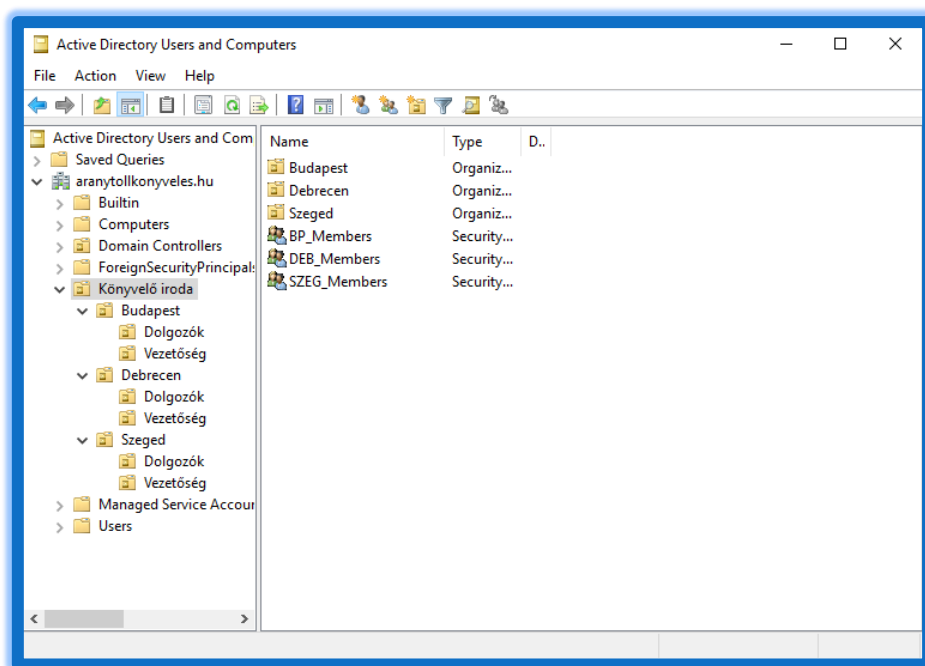
Annak érdekében, hogy a weboldalunk könnyen, a szerverünk IP címének birtoklása nélkül is elérhető és könnyen megjegyezhető legyen, DNS szolgáltatást használtunk. A DNS segítségünkre van, amennyiben begépeli az egyik kliens a weboldalunk domain címét, ami a mi esetünkbe az „aranytollkonyveles.hu”, akkor lefordítja ezt a szerver IP címére és az elérése során ezt használja. A fenti ábrán látható, hogy a 192.168.3.50-es IPv4-es cím van hozzárendelve az általunk használt domain névhez (42.ábra).



43.ábra

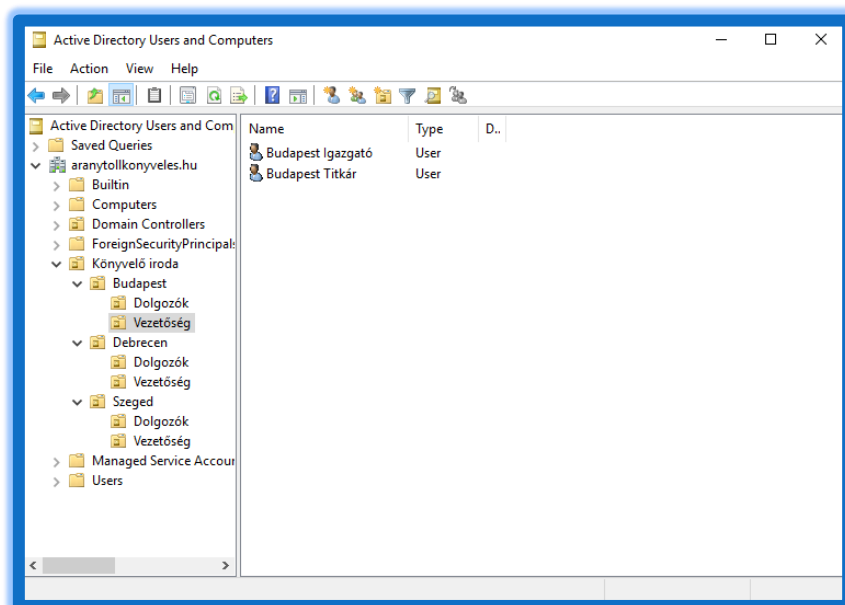
Természetesen a mi szerverünk sem tudja alapértelmezetten a világ összes domainjét lefordítani, ennek okán felvettünk két Forwarder-t, egészen pontosan a Google szolgáltatásait vettük igénybe, hisz világszerte ők biztosítják a legnagyobb DNS szolgáltatást (43.ábra).

Active Directory



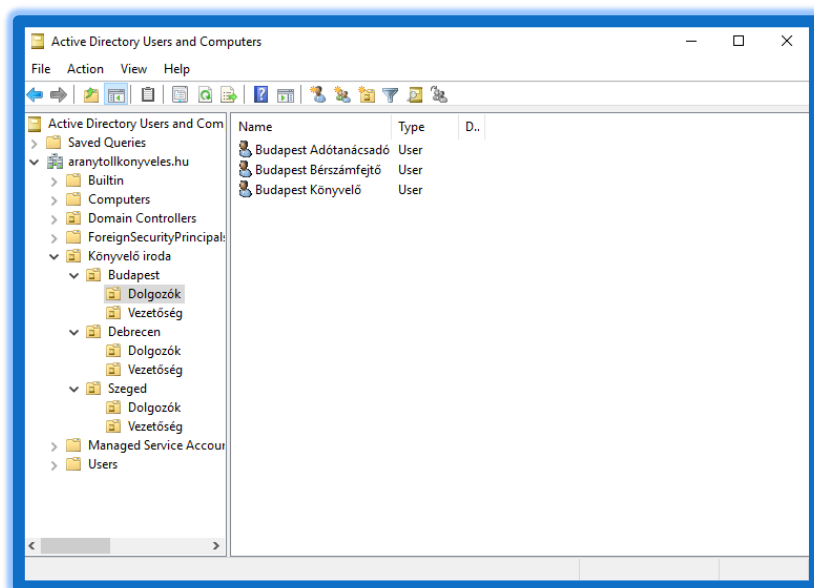
44.ábra

Az Active Directory szolgáltatásunkat „aranytollkonyveles.hu” néven hoztuk létre (44.ábra). Az AD lehetővé teszi nekünk, rendszergazdáknak, hogy könnyedén és átláthatóan kezelhessük az egyes felhasználók számára bizonyos lehetőségek szabályozását, menedzselését. A forest-ünkön belül létrehoztunk egy fő szervezeti egységet, melyet Könyvelő iroda néven találhatunk meg (45.ábra).



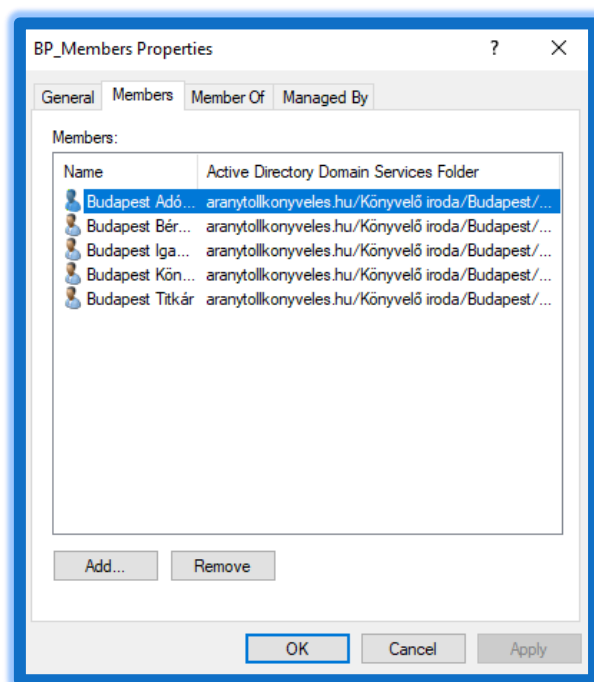
45.ábra

Ugyanakkor elkészítettünk három alszervezeti egységet is, a fő egység mellett, melyeket telephelyekre bontottunk le. Ezekbe belépve található még kettő egység, hogy elkülönítsük a dolgozókat és a vezetőséget egymástól (46.ábra).



46.ábra

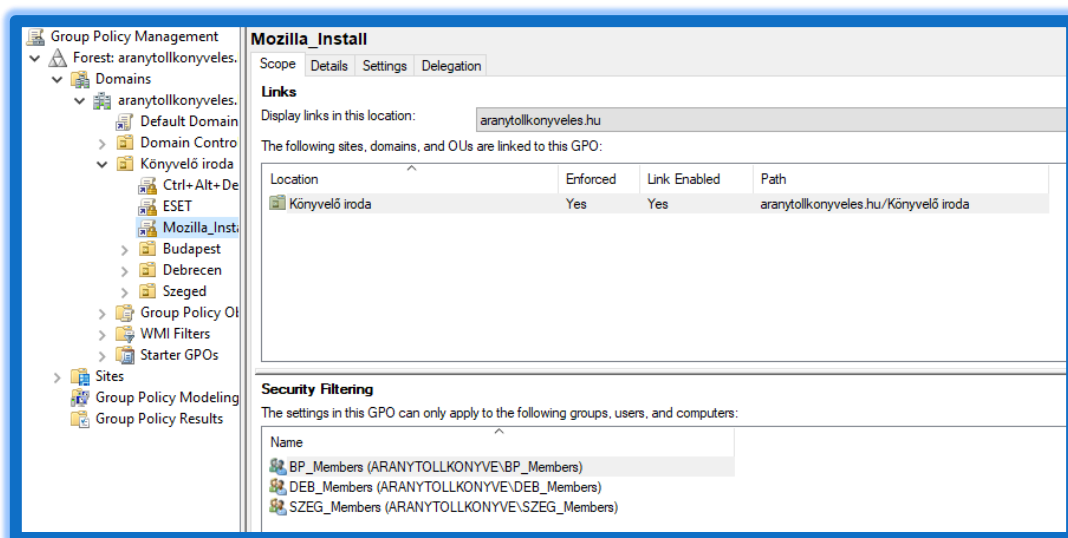
A következő ábrán közelebbről megtekinthető, hogy egyes csoportokat telephelyre bontottunk és hozzárendeltünk felhasználókat, hogy könnyedén tudjuk azokat csoportszabályzatokhoz hozzárendelni (47.ábra). Ennek köszönhetően nem kell egyesével elvégezni ezeket a folyamatokat minden felhasználóval, időtakarékos megoldásnak bizonyult a használata.



47.ábra

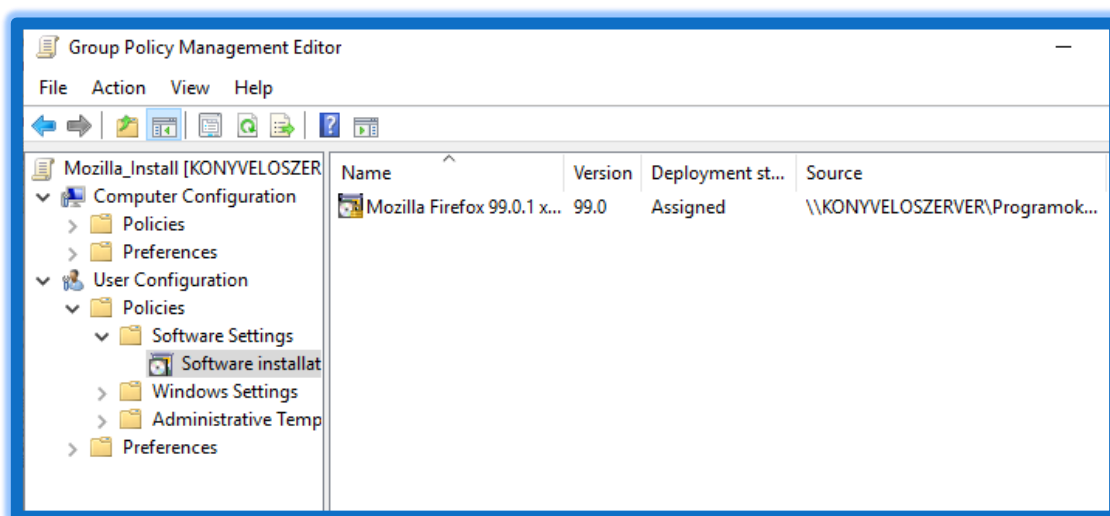
Automatizált szoftver telepítés

Az alapértelmezett Windows szolgáltatások mellett elengedhetetlennek éreztünk még két nagyon fontos és hasznos alkalmazást telepíteni a szerver kliensekre. Az egyik ilyen nélkülözhetetlen szoftver a Mozilla Firefox (48.ábra). A telephelyeken dolgozó alkalmazottak így könnyedén el tudják érni a munkájukhoz kellő információkat és forrásokat, valamint tudnak kommunikálni az ügyfelekkel. Amint belép a alkalmazott az újonnan létrehozott fiókjába, telepítésre kerülnek ezek az alkalmazások.



48.ábra

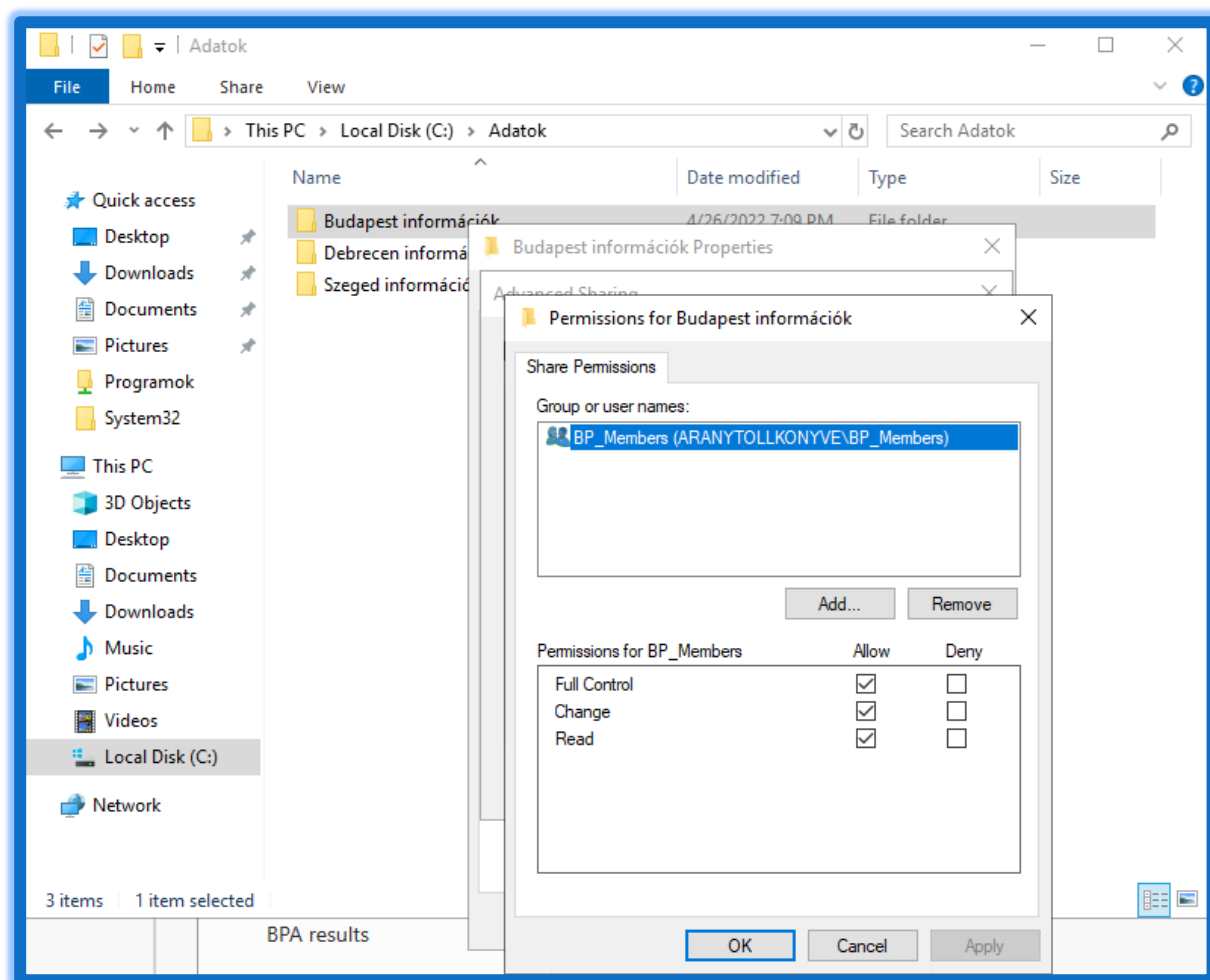
A másik létfontosságú alkalmazás pedig egy ESET Endpoint Antivirus nevezetű vírusírtó. Mindkettőhöz létrehoztunk egy-egy csoport szabályzatot, az utóbbit ESET névvel, a használt böngészőnek pedig a Mozilla_Install névvel (49.ábra).



49.ábra

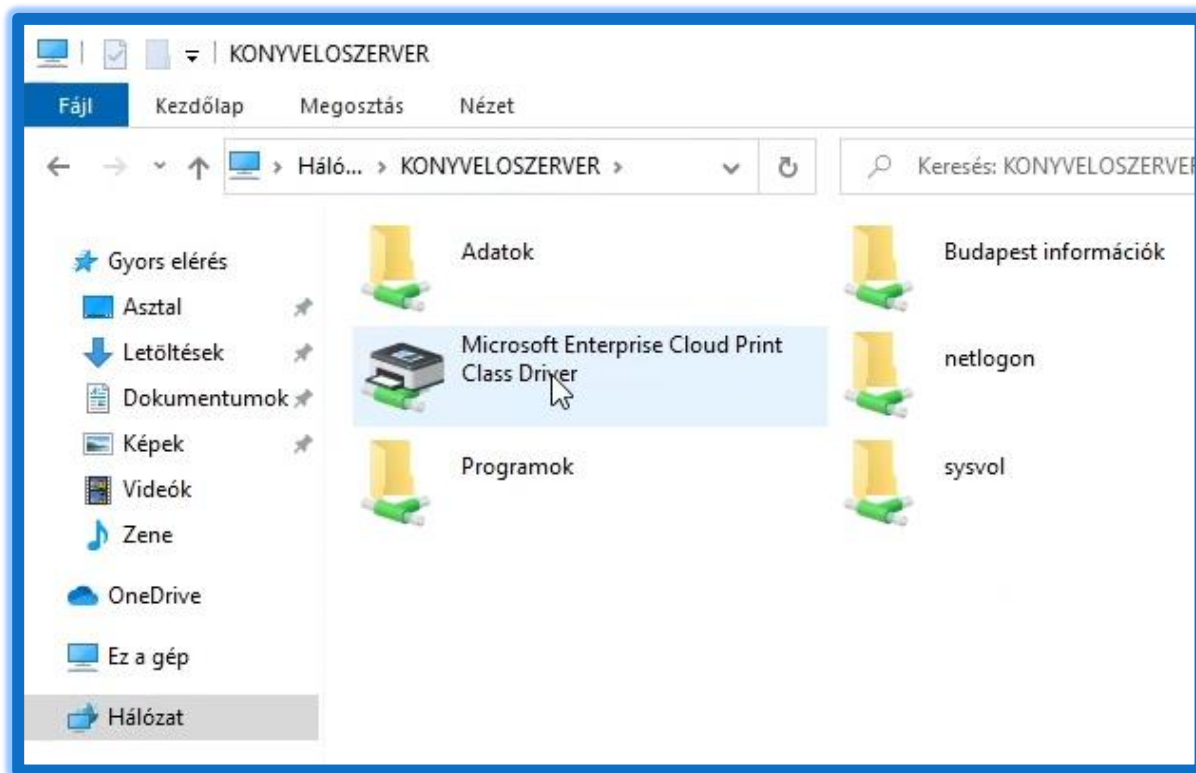
Fájl- és nyomtató megosztás

A fájlmegosztás megvalósításának eredményeképp létrehoztunk egy fő mappát, melyet „Adatok”-nak neveztünk el. Ezen a mappán belül készítettünk három almappát, a telephelyük neveivel ellátva. Az almappákhoz csak a mappák neveivel azonos telephelyen dolgozó alkalmazottaknak van hozzáférésük (50.ábra). Ez a gyakorlatban úgy néz ki, hogy egy budapesti telephelyen dolgozó alkalmazott csak a „Budapest információk” nevű mappához fér hozzá. Ezekbe a mappákba könnyedén tudnak fel- és letölteni fájlokat a dolgozók, ezzel megkönnyítve egymás dolgát és időt tudnak egymásnak, illetve maguknak megtakarítani.



50.ábra

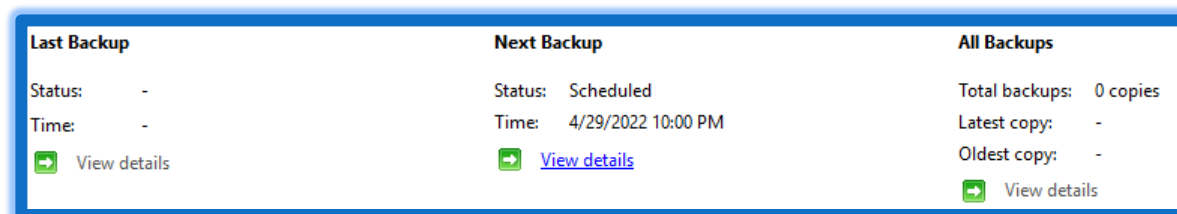
A nyomtató megosztása megkönnyítette a dolgozók életét, hiszen az egy telephelyen lévő alkalmazottak bármikor tudnak hivatalos dokumentumokat nyomtatni a közösen használható hálózati nyomtató segítségével (51.ábra).



51.ábra

Automatizált mentés

Az automatizált mentés, mint ahogyan az ábrán is látható, mindennap este 10 órakor készít egy biztonsági mentést a felhasználókról, a hálózati eszközökről, a tartománykezelőkről, a csoportszabályzatokról, a fájlmegosztásról stb. (52.ábra). Ez a lehetőség akkor válik kifejezetten fontossá és hasznossá a hálózaton belül, amikor esetleges meghibásodástól bármilyen adatvesztés történik. Abban az esetben az előző esti állapotra vissza lehet mindig állítani a szervert.



52.ábra

Hálózatprogramozás

A hálózatprogramozás megvalósításához a Netmiko modult alkalmaztuk. Ezáltal megkönnyítettük önmagunknak, hogy elmentsük a szerverre a különböző hálózati információkat, az eszközök konfigurációját. A program elindítása során nem kell bajlódniuk a felhasználónevek, illetve a hozzájuk tartozó jelszavak beírásával, egyszerűen csak futtatjuk a programot és a hálózati eszközök konfigurációját az eszköz nevével ellátva, egy .txt állományba elmenti a képen is látható a „/home/konyvadmin/backups” mappába (53.ábra).

```
from netmiko import ConnectHandler
CSR = {
    'device_type': 'cisco_ios',
    'ip': '192.168.2.1',
    'username': 'szegadmin',
    'password': 'Admin123szeg'
}
net_connect = ConnectHandler(**CSR)
hostname = net_connect.send_command('show run | i host')
device = hostname.split(" ")
print ("Biztonsági mentes keszítése a kovetkezo eszkozon: " + device[1])
filename = '/home/konyvadmin/backups/' + device[1] + '.txt'
showrun = net_connect.send_command('show run')
showvlan = net_connect.send_command('show vlan')
showver = net_connect.send_command('show ver')
log_file = open(filename, "a")
log_file.write(showrun)
log_file.write("\n")
log_file.write(showvlan)
log_file.write("\n")
log_file.write(showver)
log_file.write("\n")
log_file.close()
net_connect.disconnect()
```

53.ábra