

Scenario Configure WAN Connection with ACL filtering.

Configure access control for LAN1 and LAN2 user according to the company security policy. Verify that the configured filtering meets the given requirements.

Task 1 Basic Configuration.

- Cable the network topology.**
- Perform basic router configuration.**
 - Configure router hostnames (command *hostname*) and disable DNS lookup (*no ip domain-lookup*).
 - Create a user with name *admin* and password *cisco123* on each router. Set privilege level to 15.
router(conf)#username admin privilege 15 password cisco123
- Configure network interfaces and IP addressing**
 - Configure IP addresses on the routers. The interface on ISP towards Internet receives the address from the DHCP server (use command *ip address dhcp*).
 - Manually configure IP address 10.1.1.10 on the local web server.
 - Create loopback interfaces on London and Vienna.
- Configure DHCP server on London and Vienna**
 - Exclude the first 10 addresses. For the DNS server use address 10.10.10.1.
- Enable SSH on each router. Configure ssh access to vty 1-4 using the following commands:**
router(conf)# ip domain-name # e.g., london.com; new IOS: ip domain name (no dash)
router(conf)# crypto key generate rsa modulus 1024 # if possible choose the length 4096
router(conf-if)# line vty 0 4
router(conf-line)# login local
router(conf-line)# transport input ssh
- Configure OSPF routing.**
 - Exchange routing information about all networks connected network.
 - Do not send routing updates towards the Internet, LAN1, LAN2 and loopbacks.
- Configure NAT translation towards the Internet on ISP for all local networks (incl. loopbacks).**

Task 2 Configure Web server

- If not installed, download the HTTP File Server (HFS) from <https://rejetto.com/hfs>.
- Launch the HTF server on host 10.1.1.10.
- Check the web connection from PC1 to the web server at 10.1.1.10.

Self-check 1 Verify connectivity and routing tables.

Display the routing table on your router (London or Vienna): _____

List all non-directly connected entries in form <type> <destination-network> <metrics><next-hop><intf>

Perform connection tests for PC (name): _____

Ping PC1 -> PC2: YES – NO
ssh PC -> London: YES – NO # use Putty to test the ssh connection
ssh PC -> Vienna: YES – NO
Web access from PC -> 10.1.1.10: YES – NO
Web access from PC -> www.fit.vut.cz: YES – NO

Hints:

configuring IP address on the interface and verification

```
router(conf)# interface <name>
router(conf-if)# ip address <IP address> <mask>
router(conf-if)# no shutdown
router# show ip interface brief
```

configuring DHCP server and verification

```
router(conf)# ip dhcp exclude-addresses <starting IP> <ending IP>
router(conf)# ip dhcp pool <name>
router(dhcp-config)# network <ip address> <mask>
router(dhcp-config)# default-router <ip address>
router(dhcp-config)# dns-server <ip address>
router# show ip dhcp pool
router# show ip dhcp bindings
```

configuring PAT translation and verification

```
router(conf)# ip nat inside source list <acl-number> interface <if-name> overload
router(conf-if)# ip nat inside
router(conf-if)# ip nat outside
router# show ip nat translations
```

configuring OSPF routing and verification

```
router(conf)# router ospf <process ID>
router(config-router)# network <ip address> <wildcard mask>
router(config-router)# passive-interface default
router(config-router)# no passive-interface <if-name>
router(config-router)# default-information originate
router# show ip route
```

Self-check 2 Verify the connectivity.

If the telnet is not configured on your Windows, set the telnet service using Control Panel -> Turn windows features on.

On LAN1:

ping PC1 -> PC2:	YES – NO
ping PC1 -> 10.10.10.1:	YES – NO
telnet PC1 -> London:	YES – NO
ssh PC1 -> London:	YES – NO
ssh PC1 -> Vienna:	YES – NO
web access from PC1 -> 10.1.1.10:	YES – NO
web access from PC1 -> www.fit.vut.cz:	YES – NO

use Putty to test the telnet
use Putty to test the ssh

On LAN2:

ping PC2 -> PC1:	YES – NO
ping PC2 -> 10.10.10.1:	YES – NO
telnet PC2 -> London:	YES – NO
ssh PC2 -> London:	YES – NO
ssh PC2 -> Vienna:	YES – NO
web access from PC2 -> 10.1.1.10:	YES – NO
web access from PC2 -> www.fit.vut.cz:	YES – NO

Task 4 Modify ACLs.

Management has decided to allow DNS resolution and ICMP communication for all local networks on London and Vienna. Modify ACL rules so that ICMP and DNS communication is enabled.

Self-check 3 Verify the connectivity.

On LAN1:

ping PC1 -> PC2:	YES – NO
ping PC1 -> www.fit.vut.cz:	YES – NO
telnet PC1 -> London:	YES – NO
ssh PC1 -> London:	YES – NO
ssh PC1 -> Vienna:	YES – NO
web access from PC1 -> 10.1.1.10:	YES – NO
web access from PC1 -> www.fit.vut.cz:	YES – NO

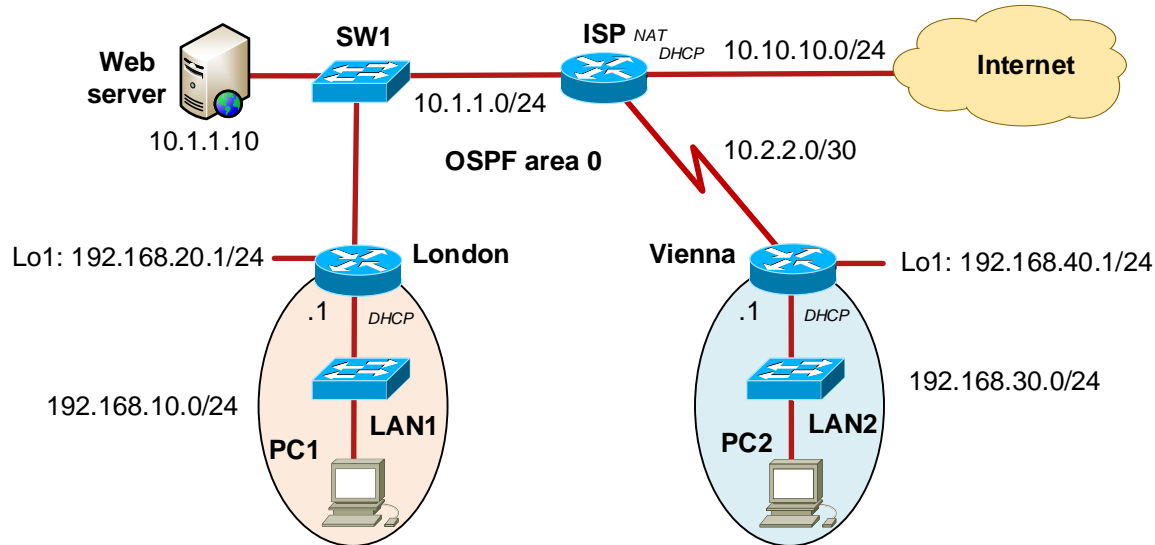
On LAN2:

ping PC2 -> PC1:	YES – NO
ping PC2 -> www.fit.vut.cz:	YES – NO
telnet PC2 -> London:	YES – NO
ssh PC2 -> London:	YES – NO
ssh PC2 -> Vienna:	YES – NO
web access from PC2 -> 10.1.1.10:	YES – NO
web access from PC2 -> www.fit.vut.cz:	YES – NO

Task 5 Evaluation, finishing the lab.

Fill the lab form and demonstrate your result to the instructor.

Switch off the PCs, remove cabling.



Task 3 Configure Traffic Filtering

Implement the following security policy on router London:

1. Allow web access from the 192.168.10.0/24 network to any network.
2. Allow an SSH connection to the Vienna router from PC1.
3. Allow users on 192.168.10.0/24 network to access 192.168.20.0/24 network (any traffic).
4. Other traffic from 192.168.10.0/24 will be blocked.

Implement the following security policy on router Vienna:

1. Network 192.168.30.0/24 is allowed to communicate with network 192.168.40.0/24 (all traffic).
2. Network 192.168.30.0/24 can access web services at the web server at 10.1.1.10.
3. No other traffic is allowed to originate from this network.

Implementation notes:

Configure one numbered extended ACL on London and one named extended ACL on Vienna only. Place the ACLs as close to the source as possible. Add a comment to each ACL to explain its function.

Hints:

```
router(conf)# access-list <num> remark <description>      # create the ACL
router(conf)# access-list <num> permit/deny <proto> <source> <dest> [eq <port>]
router(conf-if)# ip access-group <num> in/out              # place the ACL on the interface
```

```
router# show access-lists                                  # show the ACL
router# clear access-list counters <list>                  # reset the counter;
router# clear ip access-list counters <list>               # for new IOS
```

```
router(conf)# ip access-list standard/extended <name>     # adding a new rule to the existing ACL
router(conf-ext-nacl)# <entry-no> permit/deny <proto> <source> <dest> [eq <port>]
```

```
router(conf)# ip access list standard <number>            # modifying the ACL
router(conf-std-nacl)# no <entry-no>
router(conf-std-nacl)# <entry-no> permit <IP address><wildcard mask>
```