Problem 1

The packets received at each interface are coming from IPs inside the corresponding domain. Since the assignment did not require anti-spoofing such rules are not included. Also communication between machines in the same domain does not go through the firewall interfaces.

Interface	Action	Src IP	Dest IP	Dest Port	Flag	Comments
Internet/Out	Allow	*	192.168.42.17	25		(incoming traffic for mail-server)
Internet/Out	Allow	*	10.0.37.47	22		(for ssh logins)
Internet/Out	Allow	*	192.168.42.12	80		(incoming traffic for web-server)
Internet/Out	Allow	*	192.168.42.12	443		(incoming traffic for web-server)
Internet/Out	Allow	*	*	*	ACK	(established connections)
Internet/Out	Block	*	*	*		(block everything else)
Dmz	Allow	192.168.42.17	10.0.0.17	25		(forwarded traffic to mail-server)
Dmz	Allow	*	*	*	ACK	(established connections)
Dmz	Block	*	10.0.0.0/8	*		(nobody from DMZ-net can send to
Dmz	Allow	*	*	*		(allow outbound connections to any other IP address)
In	Block	*	172.16.0.0/16	*		(block outbound connections to addresses 172.16.0.0/16)

In	Allow	10.47.0.0/16	172.25.33.0/24	*	(allow outbound connections from 10.47.0.0/16 to hosts 172.25.33.0/24)
In	Block	10.47.0.0/16	*	*	(block outbound to any other IP)
In	Allow	*	*	*	(outbound connections)

Problem 2

NFS is based on the RPC technology. Main reasons that make NFS and firewall symbiosis hard are:

- Because NFS is based on RPC, it is going to use random port numbers. (Portmapper is contacted in order to discover the port number of each service on a machine). Random port selection makes the task of writing matching rules impossible.
- Different versions of NFS use different transport protocols (TCP or UDP). Therefore filtering cannot rely on the protocol.

For the above 2 reasons there is no easy way to include NFS on a rule-set of a typical packet filter (stateless firewall), which filters traffic based on IPs- and static ports rules. Stateful firewalls are not useful either because of the random ports used. Also NFS has no good user level authentication that can be used.

Solution: Application-Level Firewall, targeting NFS/RPC packet inspection. In this case the allow/deny decision is based on the contents of each packet.

Problem 3

Packet filters make the filtering decision based on source/destination IPs and source/destination ports, as found in the packets IP headers. Packets protected under ESP encryption hide the actual port number in the encrypted payload. Therefore, packet filters in the middle, different from the 2 entities that have established the IPsec SA, cannot perform filtering based on the actual port numbers, as they only see an IP packet carrying an encrypted payload. Only the host that performed the outer encapsulation (source IP address is not encrypted) and the destination of the packet (but this may not be the final one) can be identified.