Unit 5 - Applied Firewalls

Rich Macfarlane

5.1 Firewall Policies

Before any decisions can be made on firewalls and their individual policies, an overall network security policy should be created. Putting security products in place before the policy has been defined, means technology may drive the policy rather than the other way around. The policy should instead be developed in the planning stage and well before the implementation stage. "Security is a process, not a product" Bruce Schneier (11). The network security policy is used to specify firewall policy guidelines and the firewall policy procedures as shown below. Procedures will define the traffic firewalls allow to flow through them, and the guidelines define how the firewall should be implemented in general.

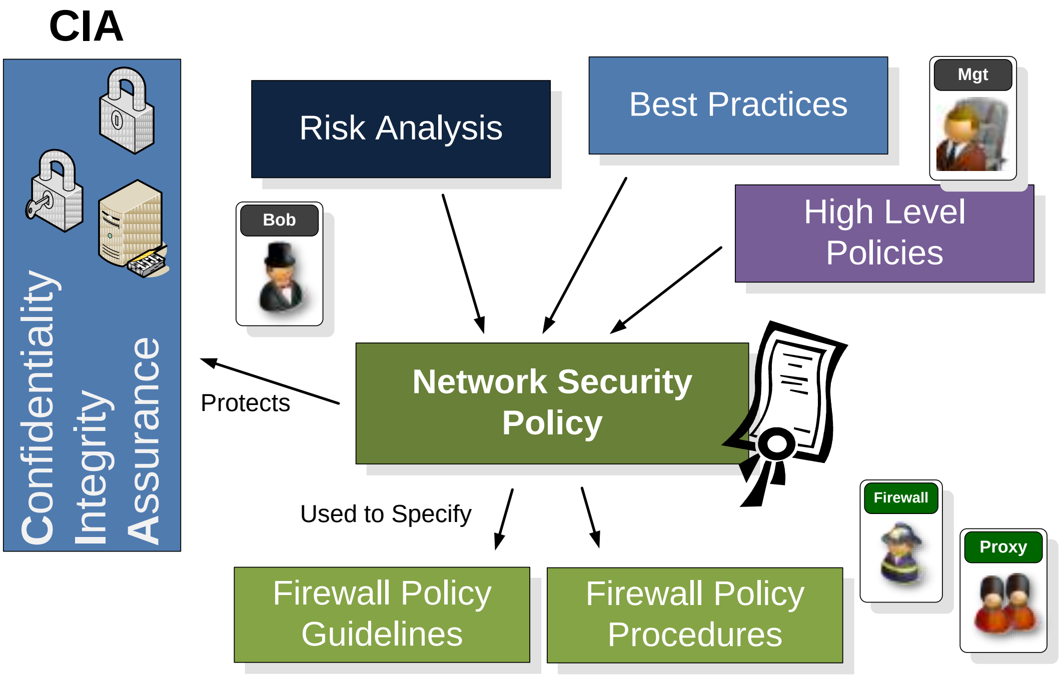


Figure 1 Firewall Policy Creation

Guidelines for firewalls could include:

- The firewall should provide logging of types of traffic dropped by the firewall.
- The perimeter firewall should be the only point of entry from the untrusted Internet (or between networks the firewall is segregating).
- The firewall should have a closed security posture.
- The firewall should provide stateful traffic filtering based on the rule sets defined in the Policy Procedures documents.
- The firewall should provide some HTTP URL filtering.

Procedures for Bobs network perimeter firewall could include the following:

- Traffic allowed through the firewall, from the trusted private network to the untrusted public Internet:
  - HTTP – web traffic - filtered on a URL blacklist
  - HTTPS – encrypted web traffic
- Traffic allowed though the firewall, from the untrusted public Internet to the trusted inside network:
  - No traffic directly

    - VPN traffic terminated on the firewall or parallel VPN termination device
- Traffic allowed though the firewall, from the untrusted public Internet to the DMZ network:
    - POP3 only to the mail server - Mail Traffic
    - HTTP only to the web server – web traffic

Firewall Policy Creation

A firewall policy should specify how firewalls handle network traffic for specific Source and Destination IP Address ranges, Protocols, Applications and Application Content based on the overall network security policy. The firewall policy should list the types of traffic that should be passing through the firewall and under which circumstances, based on the risk analysis process which would have evaluated the threats and vulnerabilities, and current security procedures.

General good practice is to block all traffic through the firewall, which is not specified in the firewall policy. i.e. A closed firewall. The following sections describe other important firewall practices.

Policies for IP Traffic

Firewall policies should only allow the IP Protocols, which the security policy defines, through the firewall. IP Protocols to be considered include ICMP, TCP, UDP, and IPSec protocols – ESP and AH, and routing protocols. These should be confined to hosts and network segments which need to use these protocols. Pass only the protocols which are needed, through the firewall, and drop all the rest by default.

Traffic with invalid source IP Addresses should be dropped at the firewall in both inbound and outbound directions. Examples of invalid addresses would be 127.0.0.1 – the local host (loopback) address, or 0.0.0.0.

Traffic with invalid source IP Addresses for inbound traffic, and with invalid destination IP Addresses for outbound traffic, should also be dropped at the perimeter firewall. This could be caused by IP Spoofing attacks, DoS attacks, malicious software, or simply faulty software or devices somewhere. Inbound source addresses which should be blocked include:

- **RFC 1918 filtering – Address Allocation for Private Internets.** (12) Addresses reserved for use in private networks. For example 192.168.0.0-192.168.255.255 should not be inbound from external networks.
- **Internal Network Addresses.** Addresses from the internal private network address space, should not be inbound from the external network. This is an IP Address spoofing attack.

Inbound traffic addressed to the firewall, should be dropped, unless the firewall is providing services such as an application proxy.

Traffic containing broadcast addresses which are from a different network segment than the broadcast is for. This is called a directed broadcast, and is a way of creating DoS attacks, such as a Smurf attack.

Block all inbound traffic to network segments and hosts that are not to be accessible from the external network. Similarly all outbound traffic that should not be accessing external networks, should be blocked.

Policies for TCP and UDP Traffic

The TCP and UDP protocols are used by applications to create communication sessions across the network. An application server will use a well known TCP or UDP port number (such as 80 for a web server), which a client will connect to, and dynamic port assignment is used at the client end of the connection. Again, a closed policy, of only allowing the TCP and UDP traffic that is needed, and dropping the rest by default is the best policy. To create the firewall policy specification a traffic matrix can be created from the network security policy/policy procedures. An example for part of Bobs policy is shown below. This is the first step towards the firewall ruleset.

Table 1 Firewall Traffic Ruleset Matrix – Internet to DMZ Ingress

Service	Source	Destination	Action
HTTP	Internet	DMZ – Public Web Server	Permit
HTTPS	Internet	DMZ – Public Web Server	Permit
Mail	Internet	DMZ –Mail Server	Permit
FTP	Internet	DMZ – FTP Server	Permit
VPN Tunnels	Home/mobile Users	VPN Terminator	Permit
Other Traffic	Any	Any	Drop

Policies for ICMP Traffic

ICMP can be used for various reconnaissance attacks, and to manipulate the flow of traffic, so some firewalls block all ICMP traffic. This can lead to problems with diagnostics for some applications, so some ICMP types should be permitted through the firewall, for examples type 3 packets – destination unreachable – for diagnostics perhaps should not be blocked outbound. ICMP types and codes are defined at (13), and detailed recommendations for filtering ICMP can be found at (14).

Policies for Application Traffic

Application proxies can be used to protect application servers, and the client hosts using them, in both inbound and outbound directions. If an application proxy exists for the application, and its performance is good enough for the type of application, and it provides better filtering than the server running the application, then a proxy should be used.

Outbound application proxies can also be used to filter the connections made from the trusted inside network. Typically outbound proxies are used for web traffic (HTTP) filtering. The proxy can filter malicious and policy violating web content, and also perform web page caching, for quicker response times, and to save bandwidth. Around 50% web page caching is not an uncommon, so saving significant bandwidth.

Logging

The firewalls logging functionality should be turned on. Use it to log blocked packets to a logging server, such as syslog server. This can then be monitored for problem traffic or malicious activities

General Good Practice

This is a list of good practices, for Firewall Policy creation and management (from Cisco):

- “Position firewalls at key security boundaries.
- Firewalls are the primary security device, but it is unwise to rely exclusively on a firewall for security.
- Deny all traffic by default, and permit only services that are needed.
- Ensure that physical access to the firewall is controlled.
- Regularly monitor firewall logs. Cisco Security Monitoring, Analysis, and Response System (MARS) is especially useful in monitoring firewall logs.
- Practice change management for firewall configuration changes.
- Firewalls primarily protect from technical attacks originating from the outside. Inside attacks tend to be nontechnical in nature.” (17).

Some other possible additions to this list could be:

- Deploy personal firewalls on all laptops and portable devices, that may be used outside the trusted network.
- Coordinate Policies of all firewalls in the trusted network, so they do not conflict with each other.

5.2 Firewall Deployment

Firewall Implementation includes the following three stages:

- **Planning** From the network security policy and the firewall procedures which have been created, requirements for which type firewalls should be decided.
- **Configuration** Implement firewall solutions, hardware and software, and configure rulesets for the firewalls.
- **Management** After deployment of the firewall solutions, monitoring and maintenance must be carried out on the systems.

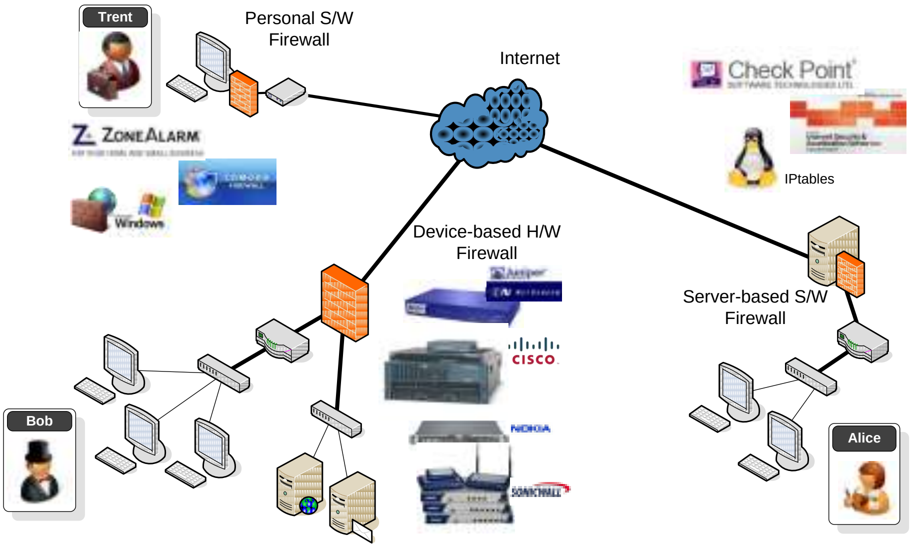
Planning

After the risk assessment, network policy, and firewall polices are complete, planning for firewall solutions can begin. Begin by selecting an appropriate solution based on the needs of the organisation. For example if a small branch office is being protected, something more than personal firewalls should be used. A defence in depth approach should be taken when planning a solution, using multiple firewalls, as well as other devices and techniques, such as VPNs, IDPS, and anti-virus software. Internal threats should always be kept in mind, as a large percentage of attacks are known to come from the inside of the trusted network these days.

Choosing a Firewall

When choosing a firewall there are many considerations to be taken into account:

- **Functionality** Choose a firewall based on the traffic types, in the organisation, which have to be protected, and which locations the firewall is to be deployed. A basic Packet Filtering firewall cannot protect HTTP Web traffic coming from the internet, and a Cisco ASA security appliance might be overkill for segregating internal networks. Also, Which other features are needed on the firewall? If it is a perimeter firewall system, does the organisation need **VPN termination**, **NAT**, **DMZ(s)**, or **content filtering** (HTTP filtering, email filtering). What type of **failover** is needed.
- **Performance** Firewall vendors have large ranges of products, and firewall performance is a key issue. Future needs should taken into account when purchasing a firewall. Key factors are **throughput**, **maximum concurrent connections**, **throughput of encrypted traffic** (VPN traffic), and **maximum number of VPN tunnels.**



- **Management** What type of interface for the management of the firewall - CLI or GUI, and the training required for staff to be to able to configure and manage the system.
- **Price** What will management allow to be spent on the firewall systems.

Many vendors offer a huge range of firewall products. The following is a brief introduction into some of the leading vendors and their offerings in both software and hardware firewalls.

Cisco

Cisco offer a several firewall solutions, from software running on routers, to enterprise level appliances. They are currently the market leaders, and have the widest deployment of any firewall. Juniper is catching Cisco in terms of deployment, and have some excellent, built from the ground up, products.

Packet filtering firewalling is offered as part of Cisco's router operating system (Router IOS). There are implemented using **Access Control Lists (ACLs)** and will be discussed in detail in the next section. Cisco also offer stateful (dynamic) packet filtering as part of the router IOS, using advanced features of ACLs. Application Inspection Firewalling (including stateful packet filtering) is offered through the **Cisco IOS Classic Firewall**, which uses **Content-Based Access Control (CBAC)**. Cisco have recently introduced an improved Router IOS based firewall system, called the new **Zone-Based Policy Firewall**. It has a simpler and more flexible configuration model than the CBAC based system, and can also be implemented and managed through a GUI. These are all software firewalls, which would be implemented on a hardened router, and would be appropriate for small to medium sized businesses. For better performance and greater functionality, a hardware-based firewall system would be needed. Hardware systems have several differences compared to software solutions such as, they are built for purpose and they are less vulnerable to direct attacks.

Cisco offer two entire ranges of hardware-based firewall – the Cisco Private Internet Exchange **(**PIX) 500 series, and the Cisco Adaptive Security Appliance 5500 series (ASA). The ASA range is more recent, and improves on the PIX in several ways, with added VPN and IDPS capabilities, better content filtering, simplified configuration, and better performance. The new range of ASA devices cater for everything, from small and home office (SOHO), to enterprise and service provider firewalls.

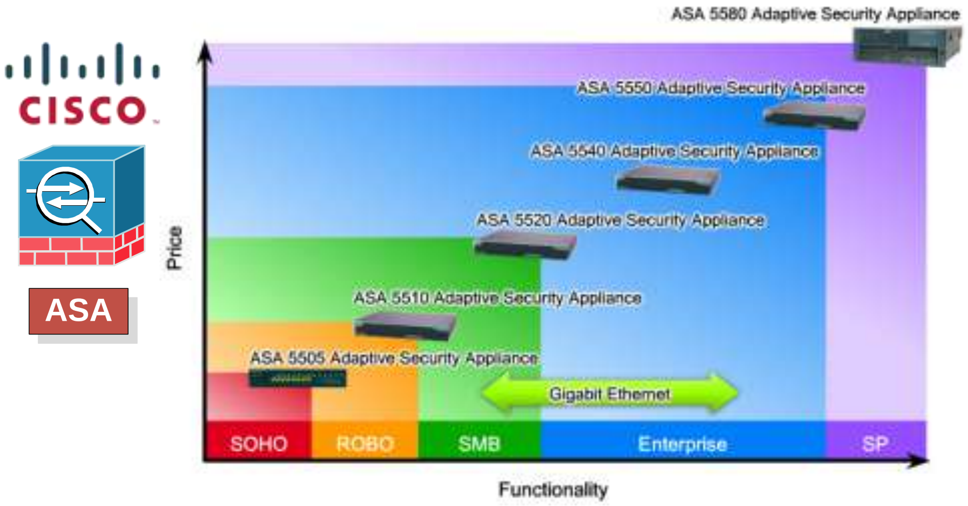

CISCO
ASA
ASA 5580 Adaptive Security Appliance
ASA 5550 Adaptive Security Appliance
ASA 5540 Adaptive Security Appliance
ASA 5520 Adaptive Security Appliance
ASA 5510 Adaptive Security Appliance
ASA 5505 Adaptive Security Appliance
Gigabit Ethernet
Price
SOHO
ROBO
SMB
Enterprise
SP
Functionality

Figure 2 Cisco ASA Device Range

Both devices are built around the **Adaptive Security Algorithm**, which is a stateful firewall concept. The Cisco ASA has been named after the algorithm the devices implement. By default, out of the box functionality means packets can only travel from a higher security level (or Trust Level) to a lower one (inside interface to outside interface). A network with a higher security level is a more trusted network. This is shown in Figure 3. The Internal Network is most trusted and always attached to the e0 inside interface, and has secuirity level 100 by default (100% trusted). The external public network is least trusted, and has security level 0 by default (0% trust). The DMZ is not fully trusted and security level 50 is assigned by default to the DMZ interface. For the latest on the ASA product range, see (18).

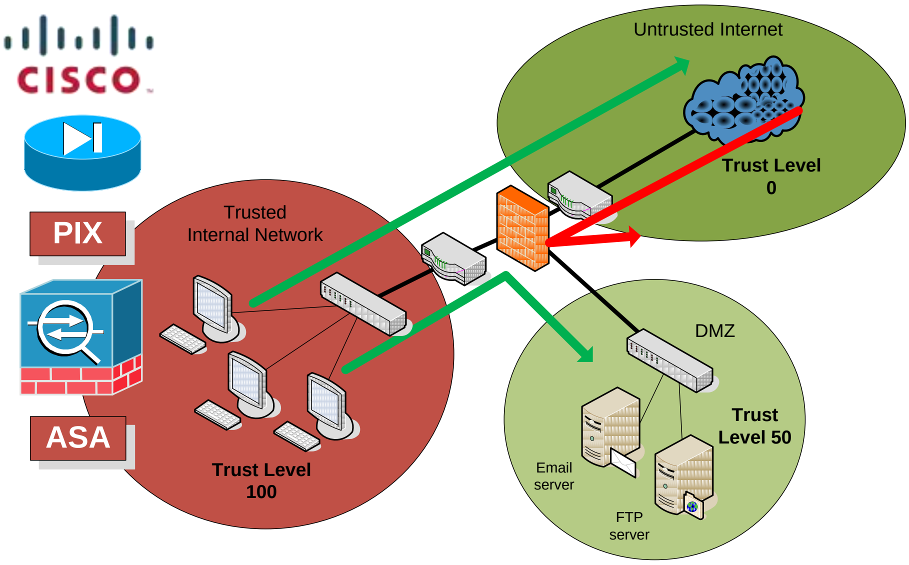


Figure 3 Cisco PIX/ASA Trust Levels and Default Traffic Flow Behaviour

Stateful Packet Filtering

Stateful Packet Filtering is implemented, on the PIX and ASA, via a firewall ruleset on each interface called an Access Control List (ACL), and a state cache. The access-list command can be used to create a ruleset for each interface. The PIX/ASA has the following default behaviour, which is shown in Figure 3.

- Traffic cannot exit an interface it entered.
- Higher to lower security level traffic is allowed, unless denied by an ACL.
- Lower to higher security level traffic is not allowed, unless specified by an ACL.
- ICMP packets are denied on all interfaces unless permitted by an ACL.

Application Protocol Inspection

By default the PIX/ASA appliance can perform application protocol inspection for compliance with standard protocol behaviour. A range of application protocols can be inspected including protocols which use dynamic channels, for example FTP, which can be configured to be inspected. Some of the devices in the range support Intrusion Prevention Systems (IPS), which traffic can be sent to for inspection. For a list of the Supported Application Protocols which can be inspected by the ASA see (18).

Content Filtering

Application layer payload content filtering is provided via a plug in module, and/or an external server. URL filtering, anti-phishing, anti-spam, antivirus, anti-spyware, and content filtering on a variety of application protocols, such as HTTP, FTP, POP3, SMTP is possible. The content filtering provided by the PIX/ASA is designed for small to medium sized businesses. Larger organisation would use external Dedicated Proxy Servers instead.

Other features

The PIX and ASA devices both provide Network Address Translation (NAT), were private addresses can be hidden from outside networks, preventing attackers learning the actual address of a host. Private IP Addressing can be used on the internal network, saving on Internet routable IP Addresses needed for the organisaton. Devices comply with protocol standards for IPSec VPNs and the ASA adds SSL (web) VPNs, to provide secure connections across unsecure networks. Different devices in the Cisco range provide different encrypted throughputs, and different maximum number of simultaneous VPN connections. The device support various plug in hardware modules which can provide Intrusion Detection and Prevention (IDSPS), to mitigate against viruses, worms, spyware, spam, and other unwanted application traffic. It accomplishes this by scanning the FTP, HTTP, POP3, and SMTP traffic that you configure the firewall to send to the IDPS module. User authentication for application protocols can be configured via the Application Layer Inspection mechanisms. Advanced, and highly configurable failover mechanisms are also included within both ASA and the PIX devices.

Branch Offices – **ASA 5505**. Throughput of up to 150Mbps firewall traffic, up to 10,000 concurrent connections. 100Mbps 3DES/AES VPN traffic for up to 10 VPN sessions. 8 Fast Ethernet Connections. Around £250.

Large Branch Offices/ Enterprise – **ASA 5520**. Has a throughput of 450Mbps for a maximum of 280,000 connections. 3DES/AES VPN up to 225Mbsp. 1 Fast Ethernet + 4 Gigabit Ethernet ports (expandable to 12 ports). Circa £2,500.

Enterprise – **ASA 5550**. Handles a throughput of 1.2Gbps for a maximum of 650,000 connections. 3DES/AES VPN performance of 425Mbsp for up to 5,000 VPN tunnels. 1 Fast Ethernet + 4 Gigabit Ethernet ports (expandable to 12 ports). Circa £7,000.

The PIX and ASA have embedded OS's in ROM. The OS is tuned and hardened, for firewall performance and security. Kernel simplification and reduced command set means very is it extremely efficient and secure. Both have received the ICSA Labs Firewall Product Certification, as well as the ISO Common Criteria EAL-4 Certification for firewalls.

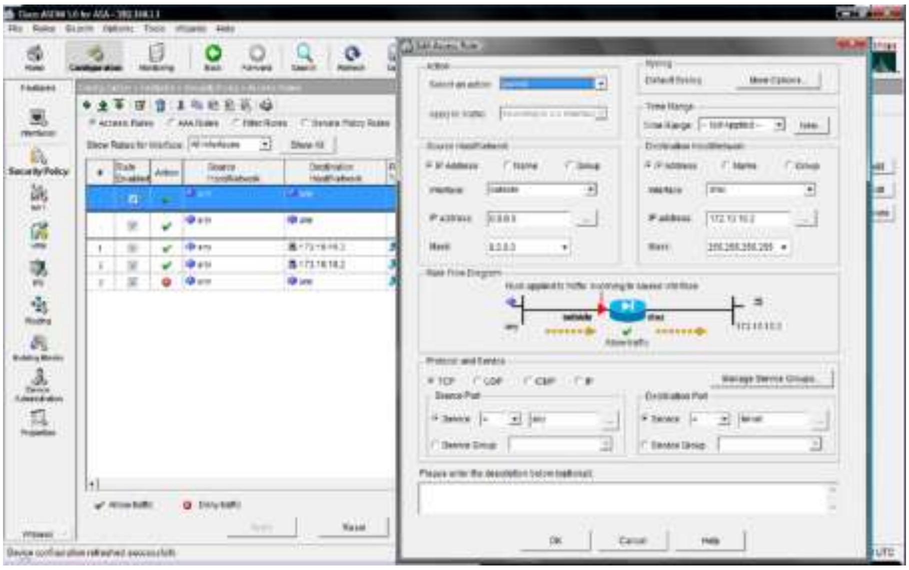


Figure 4 Adaptive Security Device Manager (ASDM)

Configuration & Management

Historically the PIX was mainly configured through the Command Line Interface (CLI), with more recently some of the management available via the PIX Device Manager (PDM). The Adaptive Security Device Manager (ASDM) interface, designed for use with the ASA, now provides virtually all configuration and management functionality for both devices, as an option to the CLI. The PIX OS used at the CLI was a flat command structure, and not as intuitive as the Cisco IOS which is found on most other Cisco devices. The latest OS, version 8, which was developed for the ASA, has been brought in line and is very similar to the Cisco Router IOS command structure. This now makes it significantly easier for Cisco users to migrate to the ASA OS.

Juniper

Juniper NetScreen is the fastest growing firewall range in the market today. NetScreen was purchased by Juniper in 2004 for $4 billion, to compete directly for the number one hardware firewall appliance market. The Juniper NetScreen range of devices provide integrated firewall and VPN functionality. Juniper has also recently released a new product line called Security Services Gateway (SSG), with new enhanced software security features, and built in Wide Area Network (WAN) interfaces.

Juniper NetScreen firewalls generally have the highest throughput of any firewall on the market currently. This includes both standard throughput and VPN throughput. Reduced Instruction Set (RISC) Processor hardware is used, rather than the less efficient Complex Instruction Set Computer (CISC) chips, used in systems such as Intel Pentium Processors.

The NetScreen hardware was developed for the purpose of the firewall device, to provide its market leading throughput. The hardware is custom built using Application-Specific Integrated Circuit (ASIC) technology, a Reduced Instruction Set (RISC) Processor, and SDRAM memory. This is significantly more efficient than using a general purpose micro-processor used in some other systems. The Security Application and OS which run on top of the hardware provide an integrated solution, designed from the ground up for purpose. The OS is ScreenOS, a Real Time Operating System (RTOS), designed to execute tasks in real world time.

Small Branch Offices – **SSG5** Secure Services Gateway. Can handle a throughput of up to 160Mbps firewall traffic, up to 8,000 concurrent connections. 40Mbps VPN traffic for up to 25 VPN tunnels. 7 Fast Ethernet Connections. Max 200 Security Policies and 8 Security Zones. Around £500.

Medium/Large Branch Offices – **SSG140**. This has a throughput of 350Mbps for a maximum of 48,000 connections. 3DES+SHA-1 VPN up to 100Mbsp. 8 Fast Ethernet + 2 Gigabit Ethernet ports. Up to 1,000 Security Policies, and 40 Security Zones. Circa £2,000.

Enterprise – **SSG550M**. Handles a throughput of 1Gbps for a maximum of 256,000 connections. 3DES+SHA-1 VPN performance of 650Mbsp for up to 1,000 VPN tunnels. 4 Gigabit Ethernet Ports. 4,000 Policies, 60 Zones. Circa £5,000.

Again, the Juniper devices have received both the ICSA Labs Firewall Product Certification and the ISO Common Criteria EAL-4 Certification. So we could create a VPN with a Cisco ASA at one end and a NetScreen device at the other, as they adhere to the common standards for IPSec.

The concept of Security Zones, or logical areas, allows logical segregation of networks for security. The zones can be associated with sections of networks, providing a more granular way of creating firewall policies. Several types of zones can exist on a device. The Security Zones define sections of the network were security is applied. Zones typically include **untrust** for the untrusted Internet, and **trust** for the trusted internal network.

Stateful Packet Filtering

Stateful packet filtering is done via a firewall ruleset called a **Policy**. A Policy is created for each pair of zones. As well as inter-zone, Policies can also be created intra-zone and globally. A default global policy exists denying any traffic from passing through the NetScreen firewall. Each product in the NetScreen range has a maximum number of Polices, based on its capacity, which cannot be exceeded. This measure aims to maximise performance on each device. This makes sense, as there is no point in creating 100,000 Polices on a low end firewall which would then give terrible performance.

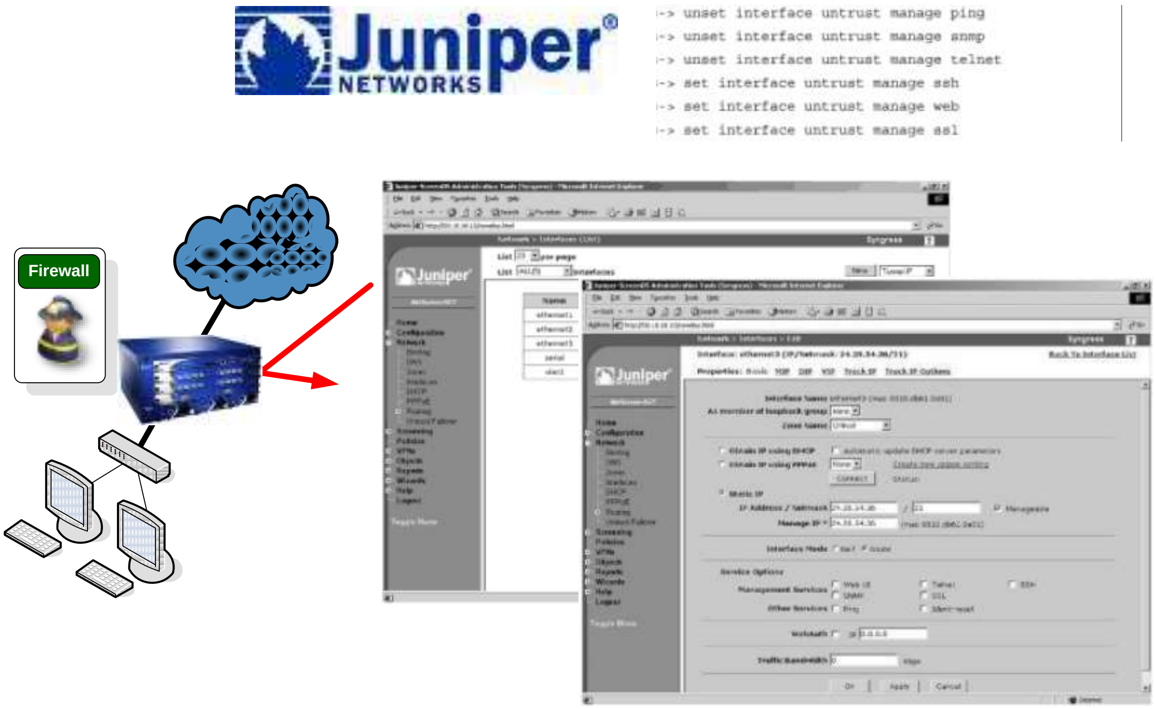
Application Level Protocol Inspection. Juniper security devices can look for attacks at the application layer. The Inspection is provided via an continually updated database, and signatures which can be customised, or created by the user.

Other Features

The Routing engine of the NetScreen-5200/5400 supports RIP, OSPF, BGP and NAT. NetScreen-5200/5400 provides IPS via Deep Packet Inspection, stateful signatures, and anomaly detection mechanisms. Some devices support URL and content filtering using external WebSense or SurfControl servers, which is similar to to the ASA. Virtual Routers are a key concept in NetScreen firewall applicances. They provide multiple routing tables within the device, allowing Zones to be mapped to different routing tables, and different interfaces. In this way external untrusted routes could be securely separated from trusted internal routes.

Configuration & Management

The entire NetScreen range of firewalls can be managed through both the CLI with the ScreenOS OS, or via their GUI interface Web User Interface (WebUI). The GUI provides everything that the CLI provides, and both are consistant throughout the range, unlike some other vendors devices. The Juniper CLI and GUI are shown in Figure 6.

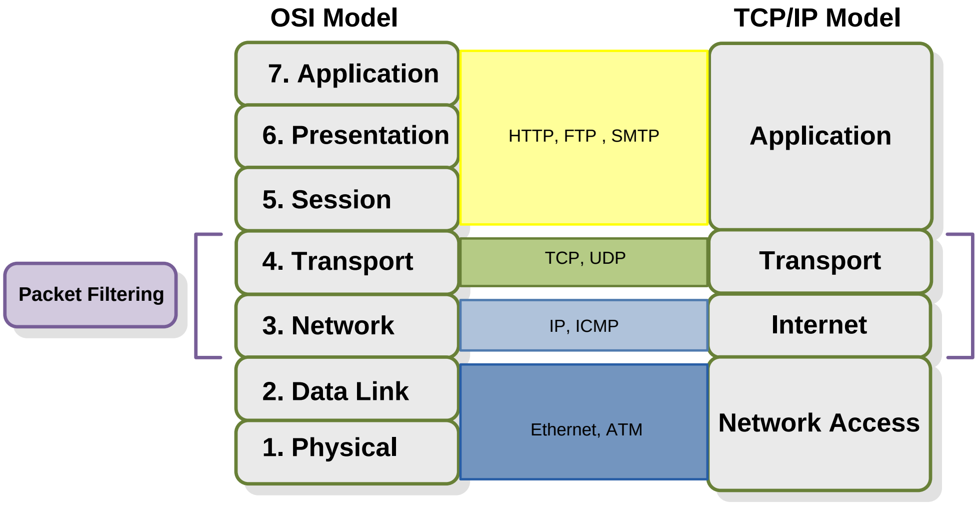


```
-> unset interface untrust manage ping
-> unset interface untrust manage snmp
-> unset interface untrust manage telnet
-> set interface untrust manage ssh
-> set interface untrust manage web
-> set interface untrust manage ssl
```

Figure 5 Juniper Firewall CLI and GUI

Firewall Configuration - Packet Filtering with Cisco ACLs

Cisco Access Control Lists (ACL) provide static packet filtering on the Cisco Routers *Internetwork Operating System (IOS)*. ACLs are also used to create stateful packet filtering firewalls, both on routers, and on Cisco firewall hardware appliances (PIX/ASA). As discusses in the section on firewall history, the first commercial packet filtering were Cisco Router ACLs, which were introduced to Router IOS version 8 in the mid 1980's.



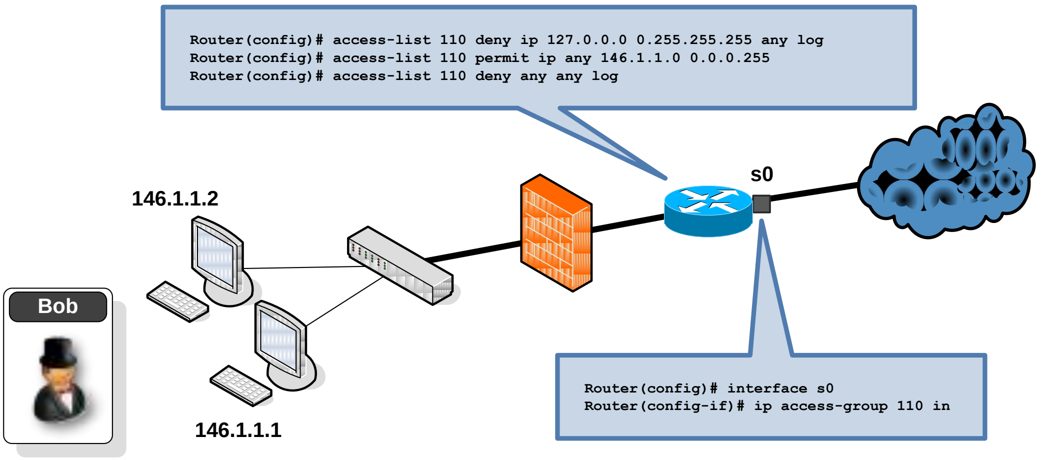
An Access Control List is a packet filtering ruleset, consisting of a list of packet filtering rules, which are entered into the router, via the Command Line Interface (CLI) or GUI. Each rule consists of the Access List Id, which can be a number or name, an **Action** - **Permit** or **Deny**, and the filtering rule itself. The rule is compared to the contents of the Layer 3 and Layer 4 packet header data of each packet in the traffic being filtered. An implicit deny all rule is automatically added to the end of the ruleset, so the ruleset has a closed security stance by default. An explicit deny all rule is sometimes added so that packets which don't match any of the rules in the ruleset are logged.

There are 2 steps to configuring an ACL:

1. Create the ACL rules
2. Apply the ACL ruleset to an interface, and in either the inbound or outbound direction.

An ACL ruleset entered being entered into router:

```
Router(config)# access-list 110 deny ip 127.0.0.0 0.255.255.255 any log
Router(config)# access-list 110 permit ip any 146.1.1.0 0.0.0.255
Router(config)# access-list 110 deny any any log
```



```
Router(config)# access-list 110 deny ip 127.0.0.0 0.255.255.255 any log
Router(config)# access-list 110 permit ip any 146.1.1.0 0.0.0.255
Router(config)# access-list 110 deny any any log
```

```
Router(config)# interface s0
Router(config-if)# ip access-group 110 in
```

Figure 6 ACL Created and Applied to Boundary Router, to filter inbound traffic

The ACL packet filtering ruleset is then applied to one of the interfaces of the router, either inbound or outbound. In this case it's on the serial0 interface and in the inbound direction, as we want to filter invalid traffic coming from the Internet:

```
Router(config)# interface s0
Router(config-if)# ip access-group 110 in
```

Each packet which comes from the Internet and enters this interface, will now be compared against the ACL 110 ruleset. If the packets header information matched a rule the action specified will be carried out and **no further rules will be compared**. This is called a **first match ruleset**. For example if a packet has a source (Layer 3) IP Address of 127.0.0.0, then the packet would match the first rule and the action deny would be carried out. i.e. the packet would be dropped.
Figure 7 shows the Layer 3 IP header fields, and Layer 4 TCP header fields, from each packet, which are used to match against the ACL rules.

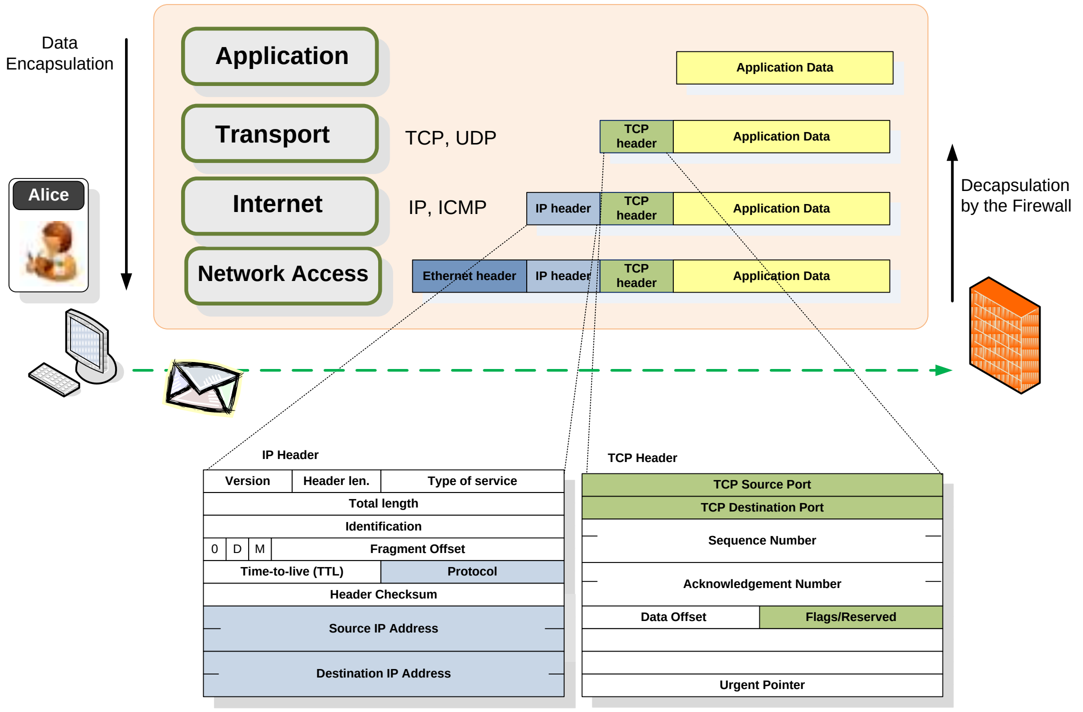


Figure 7 Packet Filtering Layers 3 IP Header, and Layer 4 TCP Header, Data

There are two types of Cisco ACL, **Standard** and **Extended**:

Standard ACLs

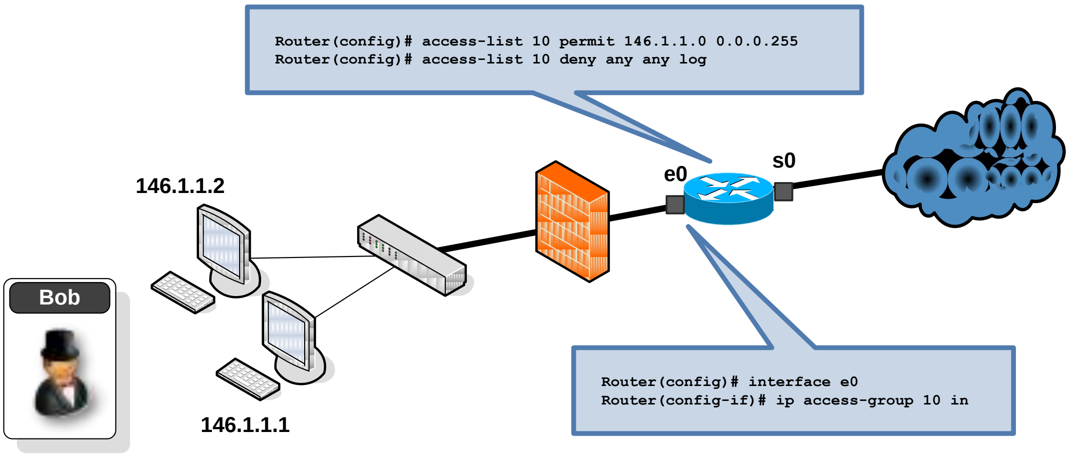
Standard ACLs filter IP Packets based on the (Layer3) Source IP Address only. The destination IP Address and (Layer 4) TCP information is ignored, so only basic filtering can be performed. The example below allows all traffic from the 146.1.1.0 network segment, Bobs subnet (with any destination, as it is not checked).

Command Syntax:
(this is a simplified version of the command syntax, for full command definitions, see (17)

```
access-list {1-99} {permit | deny} source-address [source-wildcard] [log]
```

Example:

```
Router(config)# access-list 10 permit 146.1.1.0 0.0.0.255
Router(config)# access-list 10 deny any any log
```



```
Router(config)# access-list 10 permit 146.1.1.0 0.0.0.255
Router(config)# access-list 10 deny any any log
```

```
Router(config)# interface e0
Router(config-if)# ip access-group 10 in
```

Extended ACLs

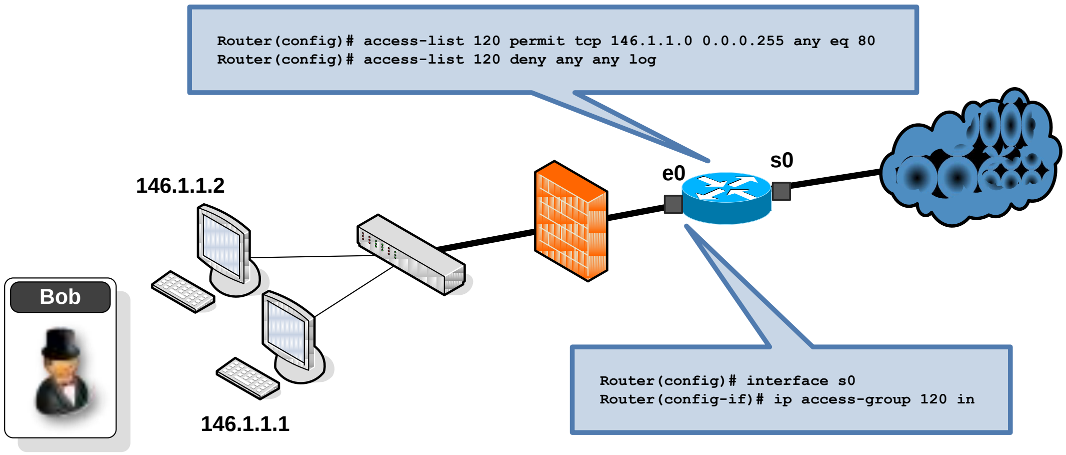
Extended ACLs filter packets based on (Layer 3) Source and Destination IP Addresses, (Layer 4) Source and Destination TCP and UDP Ports , and Protocol (IP, ICMP, TCP, UDP). The example below allows all traffic from Bobs subnet to any destination web server, out on the Internet (TCP Protocol & Port=80).

Command Syntax:

```
access-list {100-199} {permit | deny} protocol source-address [source-
wildcard] [operator port] destination-address [destination-wildcard] [operator
port] [established] [log]
```

Example:

```
Router(config)# access-list 120 permit tcp 146.1.1.0 0.0.0.255 any eq 80
Router(config)# access-list 120 deny any any log
```



```
Router(config)# access-list 120 permit tcp 146.1.1.0 0.0.0.255 any eq 80
Router(config)# access-list 120 deny any any log
```

```
Router(config)# interface s0
Router(config-if)# ip access-group 120 in
```

Applying the ACL to an Interface

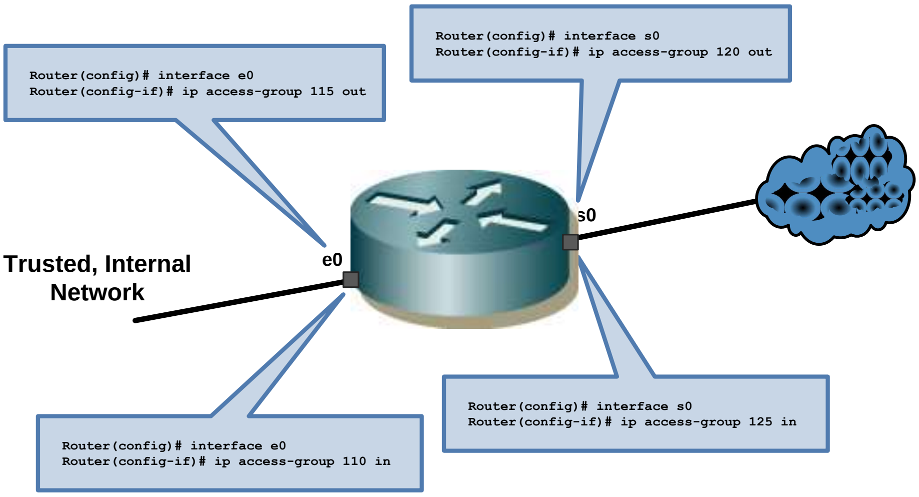
Once created, the ACL rulset is applied to an interface, either to filter outbound traffic (`out paramater`), or to filter inbound traffic (`in parameter`). Inbound ACLs are more efficient than outbound ACLs. With Outbound ACLs the router has to route the traffic to the outbound interface, and then compare the packet to the ACL. Inbound is compared to the ACL first, so if the packet is dropped it is not routed. Only one ACL in each direction can be applied to each interface. The example below shows filtering of traffic inbound and outbound on two different interfaces.

Command Syntax:

```
ip access-group access-list-number {in | out}
```

Example:

```
Router(config)# interface e0
Router(config-if)# ip access-group 120 in
```



```
Router(config)# interface e0
Router(config-if)# ip access-group 115 out
```

```
Router(config)# interface s0
Router(config-if)# ip access-group 120 out
```

```
Router(config)# interface e0
Router(config-if)# ip access-group 110 in
```

```
Router(config)# interface s0
Router(config-if)# ip access-group 125 in
```

ACLs can be numbered or named. Standard IP ACLs can be numbered 1 to 99, and 1300 to 1999. Extended IP ACLs can be numbered 100 to 199, and 2000 to 2699.

Named ACLs

Named ACLs, as opposed to numbered ACLs, are self documenting as well as provide all the functions of numbered ACLs. Named ACLs cannot have spaces or punctuation in the name, and best practice is to use CAPITAL LETTERS.

Command Syntax:

```
Router(config)# ip access-list [standard | extended] ACL_NAME
```

Named ACLs are different from numbered ACLs, in that the **`ip access-list`** command (not the **`access-list`** command used for numbered ACLs) takes the user into ACL line configuration mode, giving the prompt: **`Router(config-std-nacl)#.`** Individual ACL rules can then be entered, using the **`deny/permit`** commands until the **`exit`** command is entered. Rules can also be removed by prefixing the rule to be removed with the **`no`** command.

Command Syntax:

The Standard Named ACL gives the same options as the standard numberd:

```
Router(config-std-nacl)# {permit | deny} {source [source-wildcard] | any}
```

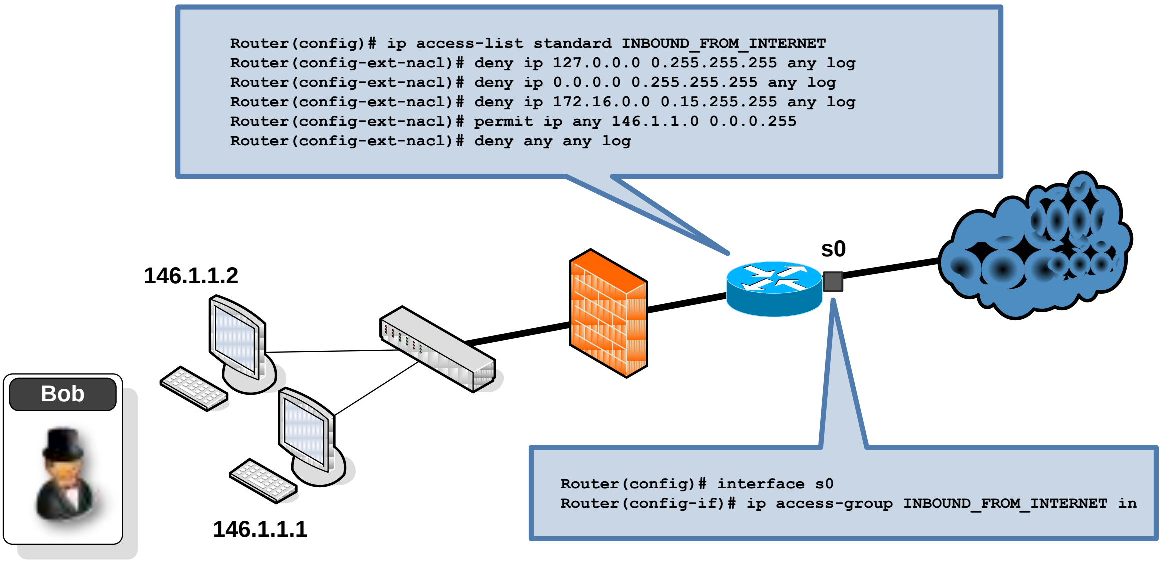
The extended command gives more commands (same as the extended numbered ACL):

```
Router(config-ext-nacl)# {permit | deny} protocol source-address [source-
wildcard] [operator operand] destination-address [destination-wildcard]
[operator operand] [established]
```

Example:

```
Router(config)# ip access-list standard INBOUND_FROM_INTERNET
Router(config-ext-nacl)# remark Block Invalid Source Addresses From Internet
Router(config-ext-nacl)# deny ip 127.0.0.0 0.255.255.255 any log
Router(config-ext-nacl)# deny ip 0.0.0.0 0.255.255.255 any log
Router(config-ext-nacl)# deny ip 172.16.0.0 0.15.255.255 any log
Router(config-ext-nacl)# permit ip any 146.1.1.0 0.0.0.255
Router(config-ext-nacl)# deny any any log
Router(config-ext-nacl)# exit
```

```
    Router(config)#
```



```
Router(config)# ip access-list standard INBOUND_FROM_INTERNET
Router(config-ext-nacl)# deny ip 127.0.0.0 0.255.255.255 any log
Router(config-ext-nacl)# deny ip 0.0.0.0 0.255.255.255 any log
Router(config-ext-nacl)# deny ip 172.16.0.0 0.15.255.255 any log
Router(config-ext-nacl)# permit ip any 146.1.1.0 0.0.0.255
Router(config-ext-nacl)# deny any any log
```

```
Router(config)# interface s0
Router(config-if)# ip access-group INBOUND_FROM_INTERNET in
```

Figure 8 Named ACL Created and Applied to Boundary Router, to filter inbound traffic

The Named ACL is identical to the first numbered ACL we created in this section. Go back and compare the two. The Named ACL is a fairly recent addition to the Cisco IOS, and sometimes numbered ACLs will be the system administrator’s only option, but when available Named ACLs should be used.

Comments and Logging

Notice in the previous example the **`remark`** command was used to enter a comment about the ACL that was being created. This is important, as ACLs can become very large and complex. It’s not unusual for large enterprise firewalls to have thousands of filtering rules. The name of the ACL and the remarks can also be seen in the logging information, so it is important to make them relevant and informative.

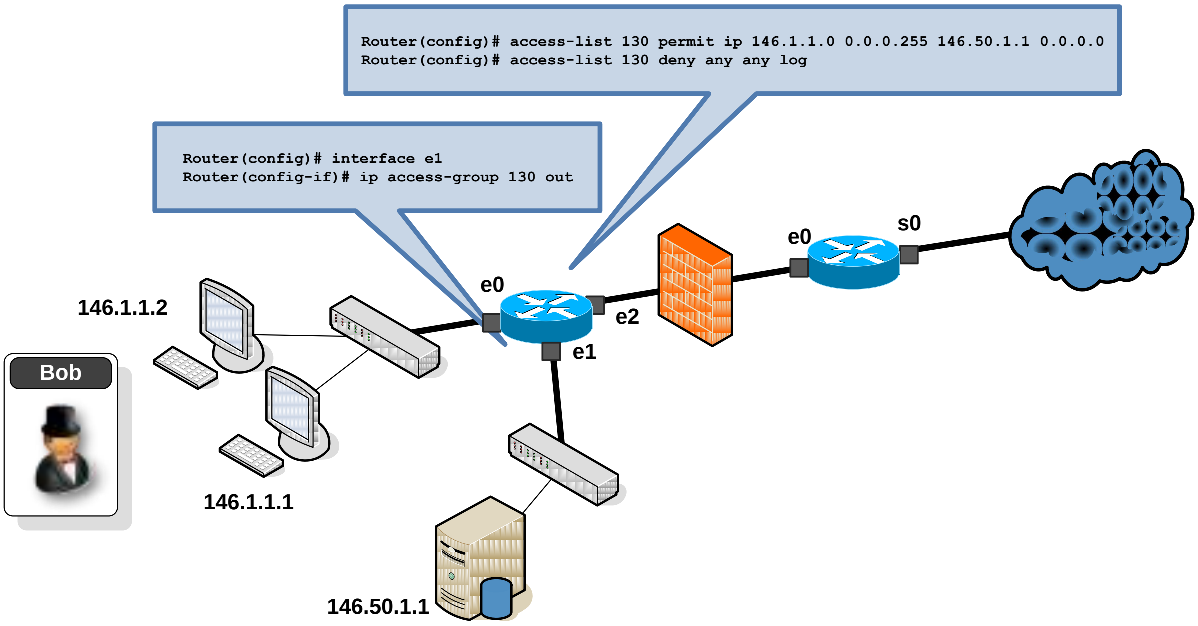
The optional **`log`** parameter was also used, so if a rule matches a packet, as well as performing the action specified, it logs the packet information and the action taken, locally or to a logging server. To switch on logging and log to the syslog server at 146.100.10.5 the following logging command would be used:

```
    Router(config)# logging on
    Router(config)# logging 146.100.10.5
```

This can then be analysed by the system administrator using tools such as the Cisco Monitoring, Analysis and Response System (MARS) to correlate bad traffic or intrusions, devices across the network. We will discover more about MARS in the IDPS chapter.

Wildcard Masks

In the example below, Bob’s network is given access to the database server at 146.50.1.1, and all other traffic is blocked, outbound on e1. The **`source-address`** and **`[source-wildcard]`** parameters have the values **`146.1.1.0`** and **`0.0.0.255`**. The **`146.1.1.0`** is the Source IP Address is a network address, in this case of Bobs network (note the zero in the last octet). The **`0.0.0.255`** is the Wildcard Mask, in which the zeros mean an exact match has to be made with the IP Address, and the 255 means the last octet will be ignored (a wildcard). So the source can be any host from Bob’s network. More wildcard examples are shown in Table 2.



```
Router(config)# access-list 130 permit ip 146.1.1.0 0.0.0.255 146.50.1.1 0.0.0.0
Router(config)# access-list 130 deny any any log
```

```
Router(config)# interface e1
Router(config-if)# ip access-group 130 out
```

The **`destination-address [destination-wildcard]`**, similarly has an IP Address, this time of the server itself, as opposed to a network address, and a wildcard mask of `0.0.0.0` which means match exactly with the IP Address in every octet. i.e. The servers IP Address.

```
Router(config)# access-list 130 permit ip 146.1.1.0 0.0.0.255 146.50.1.1 0.0.0.0
Router(config)# access-list 130 deny any any log
Router(config)# interface e1
Router(config-if)# ip access-group 130 out
Router(config)# exit
```

Table 2 Wildcard Bits to Decimal Value (one octet)

	128	64	32	16	8	4	2	1		Decimal
Match all bits in the address	0	0	0	0	0	0	0	0	=	0
Ignore all bits in the address (wildcard)	1	1	1	1	1	1	1	1	=	255
Match only last 2 address bits	1	1	1	1	1	1	0	0	=	252
Ignore Last 2 address bits(can have any value)	0	0	0	0	0	0	1	1	=	3
Match only odd numbered addresses	1	1	1	1	1	1	1	0	=	254

Matching a Single Host IP Address

In the previous example, we wanted to match a single host machine, so we used the address and wildcard values 146.50.1.1 and 0.0.0.0 to match the servers IP Address exactly. The **`host`** parameter can also be used instead of using 0.0.0.0:

For Example:

```
Router(config)# access-list 130 permit ip 146.1.1.0 0.0.0.255 host 146.50.1.1
```

Is the same as:

```
Router(config)# access-list 130 permit ip 146.1.1.0 0.0.0.255 146.50.1.1 0.0.0.0
```

Matching All IP Addresses

Another wildcard parameter which can be used as shorthand is `any`. This can be used instead of 0.0.0.0 255.255.255.255 for obvious reasons. For example:

```
Router(config)# access-list 130 deny any any log
```

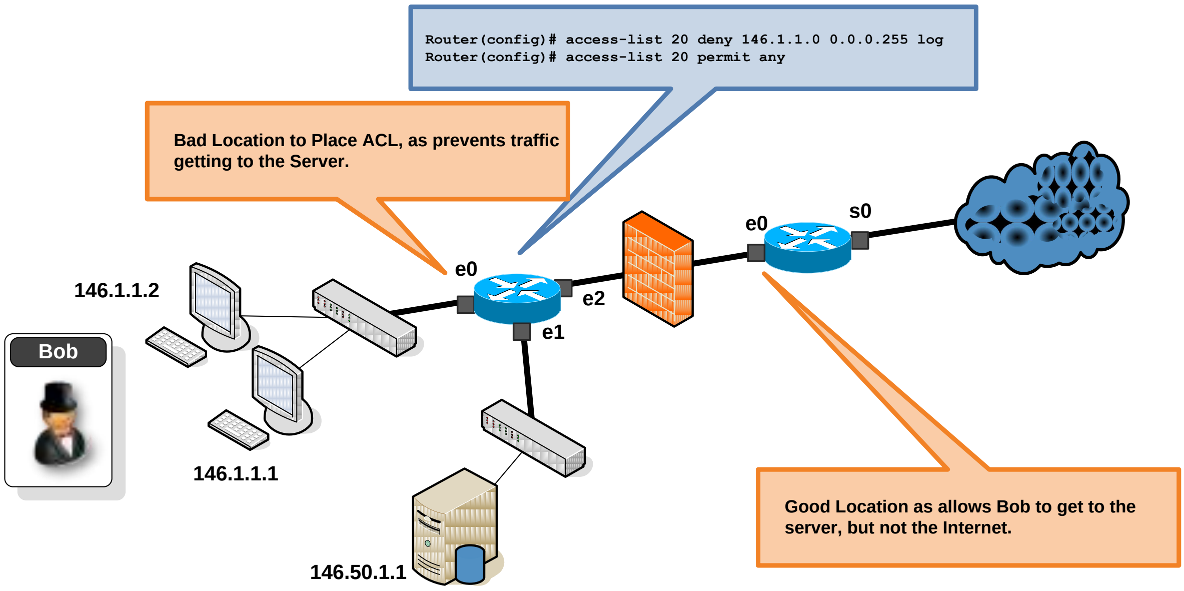
Rather than:

```
Router(config)# access-list 130 deny 0.0.0.0 255.255.255.255 0.0.0.0
255.255.255.255 log
```

Placement of ACLs

Cisco recommends placing Standard ACLs as close to the destination as possible, as they are based on the source address only. This is inefficient, but otherwise placement of standard ACLs too close to the source can result in the blocking of traffic which was not intended. In the following example a Standard ACL is being used to prevent Bobs network from getting to the Internet, but we still want him to be able to access the server.

```
    Router(config)# access-list 30 deny 146.1.1.0 0.0.0.255 log
    Router(config)# access-list 30 permit any
```



```
Router(config)# access-list 20 deny 146.1.1.0 0.0.0.255 log
Router(config)# access-list 20 permit any
```

Bad Location to Place ACL, as prevents traffic getting to the Server.

Good Location as allows Bob to get to the server, but not the Internet.

Figure 9 Standard ACL Placement

Extended ACLs are to be placed as close to the source as possible, as they are destination specific, they should filter out traffic as early as possible. This is more efficient, as otherwise packets could travel a long way across the network only to be dropped near to the destination.

Cisco also recommends using inbound ACLs for efficiency, as outbound have to be routed to the outbound interface before they are filtered, and so adding to the routers processing. This is a little simplistic, as outbound ACLs are more efficient in terms of the number of ACLs needed on a router. A good discussion on this and a good general resource for Cisco ACLs is (17).

Removing ACLs

To delete an ACL from an interface, first remove it from the interface It is applied to, then remove the ACL ruleset itself. Numbered ACLs cannot be edited, and new rules are appended to the ruleset, so when changes are done, deleting the ACL and then creating it again is the easiest way to make the changes. The ACLs should be stored securely away from the router for editing, before being added to the router. The firewall rulesets are the last thing you want an attacker to get their hands on.

```
Router(config-if)# no ip access-group 130
Router(config-if)# exit
Router(config)# no access-list 130
```

Ordering of ACL Rules

ACLs are based on a first match system, were no further matching is done if a match is found in the ACL. More specific rules have to be placed first in the ACL ruleset, otherwise they may never be reached, if more general rules, are before them in the ruleset. If all TCP traffic is blocked near the start of the ruleset, then a rule to permit HTTP traffic (TCP Port=80) to a web server will never be reached. This particular problem is known as **rule shadowing**, and is one of several anomalies which can occur within rulesets (and also between rulesets across multiple firewalls). Firewall ruleset anomaly analysis and conflict resolution has been afforded a great deal of research, and some very interesting solutions have been suggested and some automated tools created. These include visualisation methods, and some complex graph based analysis solutions (7) (8). The creation of filtering rulesets on a network device, and even the system administrators ability to understand legacy rulesets is a not a trivial task.

Securing Remote Access to the Router

To manage the router remotely, virtual terminal interfaces (or lines), commonly known as vty, are used to log into the router by a system administrator. Secure Shell (SSH) should be used (and not telnet) by the administrator to connect remotely to the virtual interfaces of the device for configuration. On a Cisco router there are 5 of these, numbered 0-4. Securing these virtual interfaces is an extremely important part of securing a router or firewall. To do this, use the **`access-class`** command. Note, that this time we log a successful connection to track who is logging into the device, as well as unsuccessful connections to track who is trying to connect to the router.

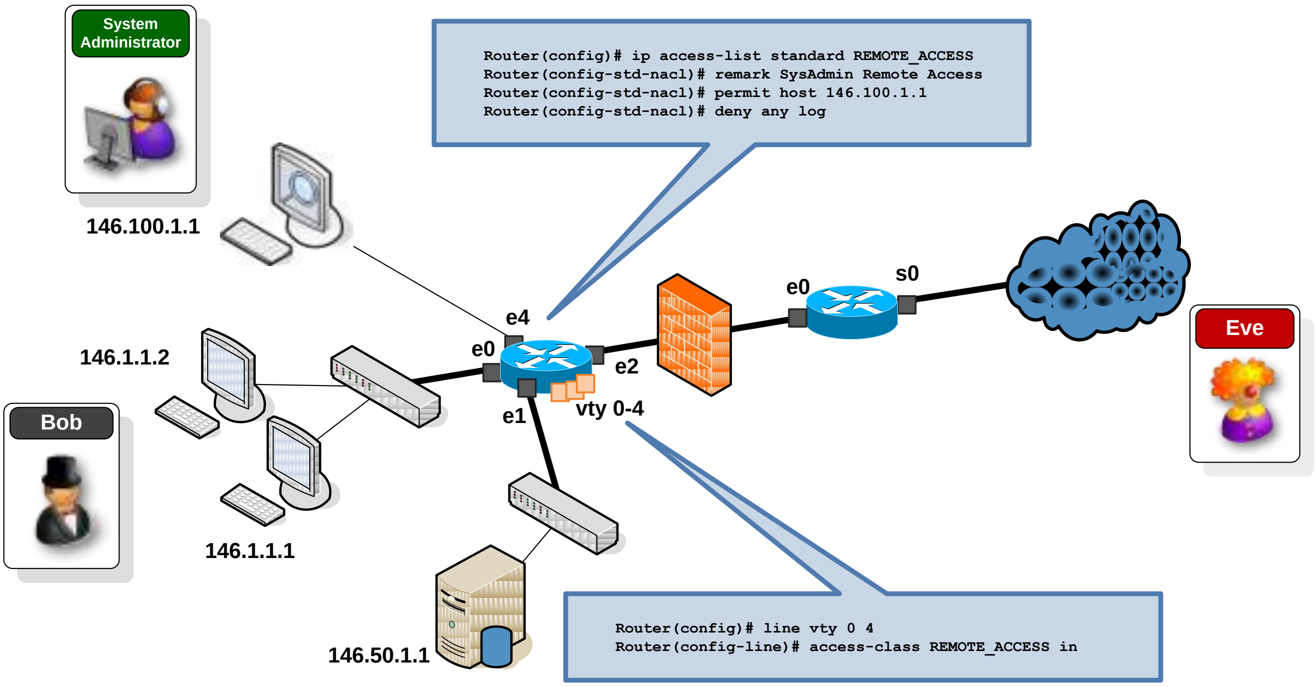
Command Syntax:

```
    Router(config-line)# access-class access-list-number {in | out}
```

Example Securing Virtual Terminal Lines:

```
    Router(config)# ip access-list standard REMOTE_ACCESS
    Router(config-std-nacl)# remark SysAdmin Remote Access
    Router(config-std-nacl)# permit host 146.100.1.1 log
    Router(config-std-nacl)# deny any log
    Router(config-std-nacl)# exit

    Router(config)# ! Set all virtual interfaces to ssh access for the sysadmin
    Router(config)# line vty 0 4
    Router(config-line)# transport input ssh
    Router(config-line)# access-class REMOTE_ACCESS in
```



```
Router(config)# ip access-list standard REMOTE_ACCESS
Router(config-std-nacl)# remark SysAdmin Remote Access
Router(config-std-nacl)# permit host 146.100.1.1
Router(config-std-nacl)# deny any log
```

```
Router(config)# line vty 0 4
Router(config-line)# access-class REMOTE_ACCESS in
```

Table 3 Virtual Interface (vty) Filtering for Remote Access

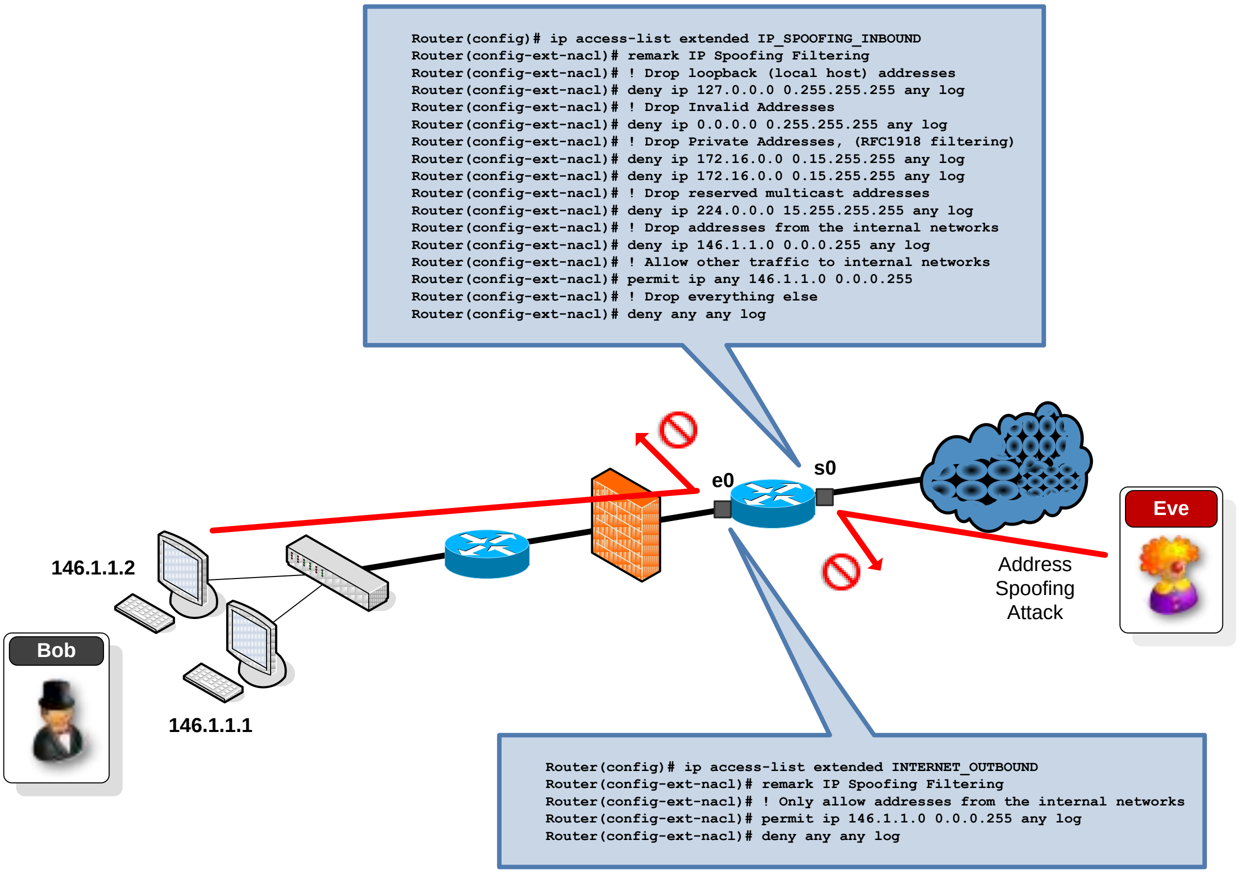
Implementing Firewall Policies with Static Packet Filtering

Mitigating IP Spoofing Attacks

Static packet filtering on a boundary router can provide IP Address spoofing mitigation. The objective of the anti-spoofing filtering is to prevent packets entering or leaving the trusted network with invalid source addresses. For example, packets inbound from the internet with a source address of an internal network obviously has a spoofed address, and should be dropped. Similarly, private and other invalid addresses should not seen inbound from the Internet.

Example IP Address Spoofing Inbound ACL (examples - not an exhaustive list):

```
    Router(config)# ip access-list extended INTERNET_INBOUND_IP_SPOOFING
    Router(config-ext-nacl)# remark IP Spoofing Filtering Inbound
    Router(config-ext-nacl)# ! Drop loopback (local host) addresses
    Router(config-ext-nacl)# deny ip 127.0.0.0 0.255.255.255 any log
    Router(config-ext-nacl)# ! Drop Invalid Addresses
    Router(config-ext-nacl)# deny ip 0.0.0.0 0.255.255.255 any log
    Router(config-ext-nacl)# ! Drop Private Addresses, (RFC1918 filtering)
    Router(config-ext-nacl)# deny ip 172.16.0.0 0.15.255.255 any log
    Router(config-ext-nacl)# deny ip 172.16.0.0 0.15.255.255 any log
    Router(config-ext-nacl)# ! Drop reserved multicast addresses
    Router(config-ext-nacl)# deny ip 224.0.0.0 15.255.255.255 any log
    Router(config-ext-nacl)# ! Drop addresses from the internal networks
    Router(config-ext-nacl)# deny ip 146.1.1.0 0.0.0.255 any log
    Router(config-ext-nacl)# ! Allow other traffic to internal networks
    Router(config-ext-nacl)# permit ip any 146.1.1.0 0.0.0.255
    Router(config-ext-nacl)# ! Drop everything else
    Router(config-ext-nacl)# deny any any log
    Router(config-ext-nacl)# exit
```



```
Router(config)# ip access-list extended IP_SPOOFING_INBOUND
Router(config-ext-nacl)# remark IP Spoofing Filtering
Router(config-ext-nacl)# ! Drop loopback (local host) addresses
Router(config-ext-nacl)# deny ip 127.0.0.0 0.255.255.255 any log
Router(config-ext-nacl)# ! Drop Invalid Addresses
Router(config-ext-nacl)# deny ip 0.0.0.0 0.255.255.255 any log
Router(config-ext-nacl)# ! Drop Private Addresses, (RFC1918 filtering)
Router(config-ext-nacl)# deny ip 172.16.0.0 0.15.255.255 any log
Router(config-ext-nacl)# deny ip 172.16.0.0 0.15.255.255 any log
Router(config-ext-nacl)# ! Drop reserved multicast addresses
Router(config-ext-nacl)# deny ip 224.0.0.0 15.255.255.255 any log
Router(config-ext-nacl)# ! Drop addresses from the internal networks
Router(config-ext-nacl)# deny ip 146.1.1.0 0.0.0.255 any log
Router(config-ext-nacl)# ! Allow other traffic to internal networks
Router(config-ext-nacl)# permit ip any 146.1.1.0 0.0.0.255
Router(config-ext-nacl)# ! Drop everything else
Router(config-ext-nacl)# deny any any log
```

```
Router(config)# ip access-list extended INTERNET_OUTBOUND
Router(config-ext-nacl)# remark IP Spoofing Filtering
Router(config-ext-nacl)# ! Only allow addresses from the internal networks
Router(config-ext-nacl)# permit ip 146.1.1.0 0.0.0.255 any log
Router(config-ext-nacl)# deny any any log
```

Figure 10 Antispoofing ACLs

Example IP Address Spoofing Outbound ACL:

```
Router(config)# ip access-list extended INTERNET_OUTBOUND
Router(config-ext-nacl)# remark IP Spoofing Filtering Outbound
Router(config-ext-nacl)# ! Only allow addresses from the internal networks
Router(config-ext-nacl)# permit ip 146.1.1.0 0.0.0.255 any log
Router(config-ext-nacl)# deny any any log
Router(config-ext-nacl)# exit
```

Similarly, outbound packets which do not have a source address from the internal networks allocated address space should be discarded. In Figure 10, the **INTERNET_INBOUND_IP_SPOOFING** is applied on the external s0 interface in the inbound direction. It drops any packets with spoofed (forged) source addresses, coming from the Internet, such as the attack from Eve, shown in the figure. The **INTERNET_OUTBOUND** ACL is applied to the internal interface, in the outbound direction, and will drop packets coming from the internal network with spoofed addresses. Another type of traffic , Bogon traffic - which are other invalid addresses, which have not been allocated to organisations yet - should also be added to the ACL inbound. Lists of Bogon addresses can be found at the IANA web site.

Mitigating ICMP Attacks

Various different network applications use ICMP traffic for valid reasons, so each implementation has to be analysed separately, and there is no template for ICMP use. ICMP traffic should be restricted to the minimum needed by the network in question. Attackers commonly use ICMP messages to perform reconnaissance on networks, as well as DoS attacks such as Ping of Death, and Smurf attacks. Unreachable messages are normally needed as well as echo messages, which may be useful for internal users to ping external hosts to test for connectivity, but most other types can be blocked. ICMP types and codes are defined at (13), and more recommendations for filtering ICMP can be found at (14). Cisco ACLs have a slightly different set of parameters if the protocol is ICMP.

Command Syntax for ICMP ACL rule:

```
    Router(config-ext-nacl)# {permit|deny} icmp source source-wildcard
           destination destination-wildcard [icmp-type [icmp-code] | icmp-
           message] [log]
```

Example ICMP Inbound ACL :

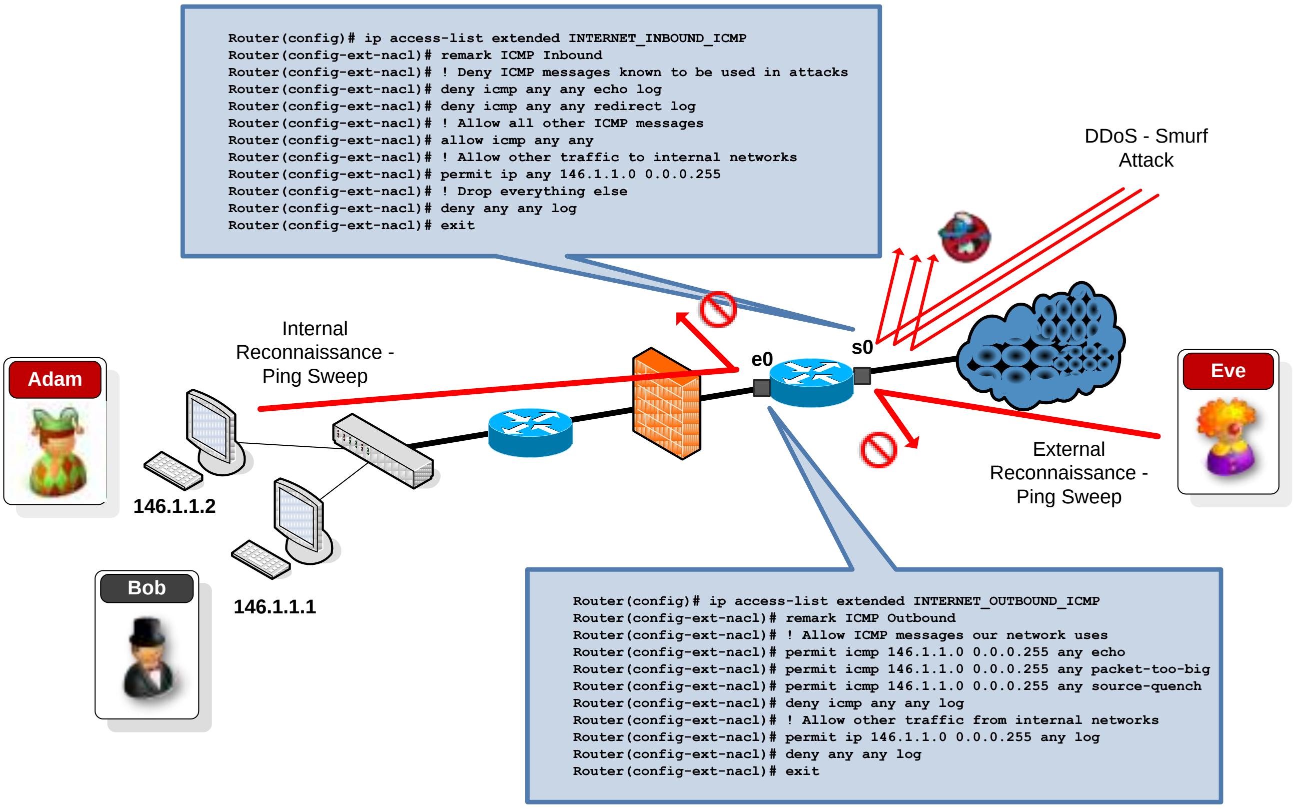
```
    Router(config)# ip access-list extended INTERNET_INBOUND_ICMP
    Router(config-ext-nacl)# remark ICMP Inbound
    Router(config-ext-nacl)# ! Deny ICMP messages known to be used in attacks
    Router(config-ext-nacl)# deny icmp any any echo log
    Router(config-ext-nacl)# deny icmp any any redirect log
    Router(config-ext-nacl)# ! Allow all other ICMP messages
    Router(config-ext-nacl)# allow icmp any any
    Router(config-ext-nacl)# ! Allow other traffic to internal networks
    Router(config-ext-nacl)# permit ip any 146.1.1.0 0.0.0.255
    Router(config-ext-nacl)# ! Drop everything else
    Router(config-ext-nacl)# deny any any log
    Router(config-ext-nacl)# exit
```

The above ACL drops ICMP packets known to be used in attacks. A better way to go about inbound filtering is to only allow the ICMP messages essential to network operation, and deny the rest (a closed security stance). This is more difficult to create. An example might be:

```
    Router(config)# ip access-list extended INTERNET_INBOUND_ICMP
    Router(config-ext-nacl)# remark ICMP Inbound
    Router(config-ext-nacl)# ! Allow ICMP messages our network uses
    Router(config-ext-nacl)# permit icmp any any unreachable log
    Router(config-ext-nacl)# permit icmp any any echo-reply log
    Router(config-ext-nacl)# ! Deny all other ICMP messages
    Router(config-ext-nacl)# deny icmp any any
    Router(config-ext-nacl)# ! Allow other traffic to internal networks
    Router(config-ext-nacl)# permit ip any 146.1.1.0 0.0.0.255
    Router(config-ext-nacl)# ! Drop everything else
    Router(config-ext-nacl)# deny any any log
    Router(config-ext-nacl)# exit
```

Example ICMP Outbound ACL:

```
    Router(config)# ip access-list extended INTERNET_OUTBOUND_ICMP
    Router(config-ext-nacl)# remark ICMP Outbound
    Router(config-ext-nacl)# ! Allow ICMP messages our network uses
    Router(config-ext-nacl)# permit icmp 146.1.1.0 0.0.0.255 any echo
    Router(config-ext-nacl)# permit icmp 146.1.1.0 0.0.0.255 any packet-too-big
    Router(config-ext-nacl)# permit icmp 146.1.1.0 0.0.0.255 any source-quench
    Router(config-ext-nacl)# deny icmp any any log
    Router(config-ext-nacl)# ! Allow other traffic from internal networks
    Router(config-ext-nacl)# permit ip 146.1.1.0 0.0.0.255 any log
    Router(config-ext-nacl)# deny any any log
    Router(config-ext-nacl)# exit
```


DDoS - Smurf Attack
Internal Reconnaissance - Ping Sweep
Adam
146.1.1.2
Bob
146.1.1.1
e0
s0
Eve
External Reconnaissance - Ping Sweep

```
Router(config)# ip access-list extended INTERNET_INBOUND_ICMP
Router(config-ext-nacl)# remark ICMP Inbound
Router(config-ext-nacl)# ! Deny ICMP messages known to be used in attacks
Router(config-ext-nacl)# deny icmp any any echo log
Router(config-ext-nacl)# deny icmp any any redirect log
Router(config-ext-nacl)# ! Allow all other ICMP messages
Router(config-ext-nacl)# allow icmp any any
Router(config-ext-nacl)# ! Allow other traffic to internal networks
Router(config-ext-nacl)# permit ip any 146.1.1.0 0.0.0.255
Router(config-ext-nacl)# ! Drop everything else
Router(config-ext-nacl)# deny any any log
Router(config-ext-nacl)# exit
```

```
Router(config)# ip access-list extended INTERNET_OUTBOUND_ICMP
Router(config-ext-nacl)# remark ICMP Outbound
Router(config-ext-nacl)# ! Allow ICMP messages our network uses
Router(config-ext-nacl)# permit icmp 146.1.1.0 0.0.0.255 any echo
Router(config-ext-nacl)# permit icmp 146.1.1.0 0.0.0.255 any packet-too-big
Router(config-ext-nacl)# permit icmp 146.1.1.0 0.0.0.255 any source-quench
Router(config-ext-nacl)# deny icmp any any log
Router(config-ext-nacl)# ! Allow other traffic from internal networks
Router(config-ext-nacl)# permit ip 146.1.1.0 0.0.0.255 any log
Router(config-ext-nacl)# deny any any log
Router(config-ext-nacl)# exit
```

Figure 11 Mitigating ICMP Attacks

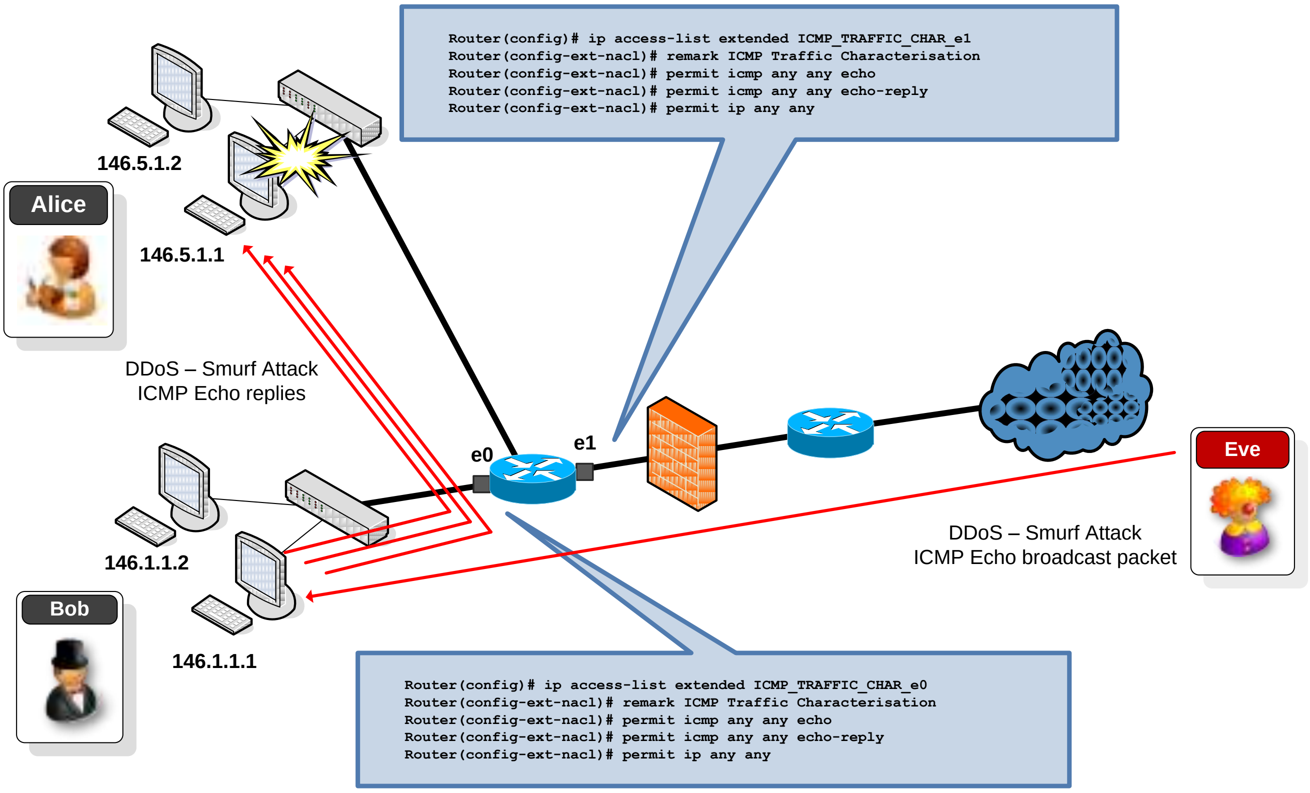
This and the IP Spoofing Inbound ACL, and any other inbound rules, would have to be combined, as only one ACL per interface are allowed in any one direction.

Traffic Characterisation

Another common use for ACLs, is to gather information about suspected attacks. A technique known as traffic characterisation can be used to do this. If an attack is suspected, through monitoring the traffic characteristics, the information can then be used to mitigate against the attack. ACLs can be used to group packets into their relevant attack streams and report on them. A series of **permit** statements can be used to set up these traffic streams, which can then be checked with the simple show **ip access-list** command.

Example ICMP Traffic Characterisation ACL:

```
    Router(config)# ip access-list extended ICMP_TRAFFIC_CHAR_e0
    Router(config-ext-nacl)# remark ICMP Traffic Characterisation
    Router(config-ext-nacl)# permit icmp any any echo
    Router(config-ext-nacl)# permit icmp any any echo-reply
    Router(config-ext-nacl)# permit ip any any
    Router(config-ext-nacl)# exit
```



```
Router(config)# ip access-list extended ICMP_TRAFFIC_CHAR_e1
Router(config-ext-nacl)# remark ICMP Traffic Characterisation
Router(config-ext-nacl)# permit icmp any any echo
Router(config-ext-nacl)# permit icmp any any echo-reply
Router(config-ext-nacl)# permit ip any any
```

```
Router(config)# ip access-list extended ICMP_TRAFFIC_CHAR_e0
Router(config-ext-nacl)# remark ICMP Traffic Characterisation
Router(config-ext-nacl)# permit icmp any any echo
Router(config-ext-nacl)# permit icmp any any echo-reply
Router(config-ext-nacl)# permit ip any any
```

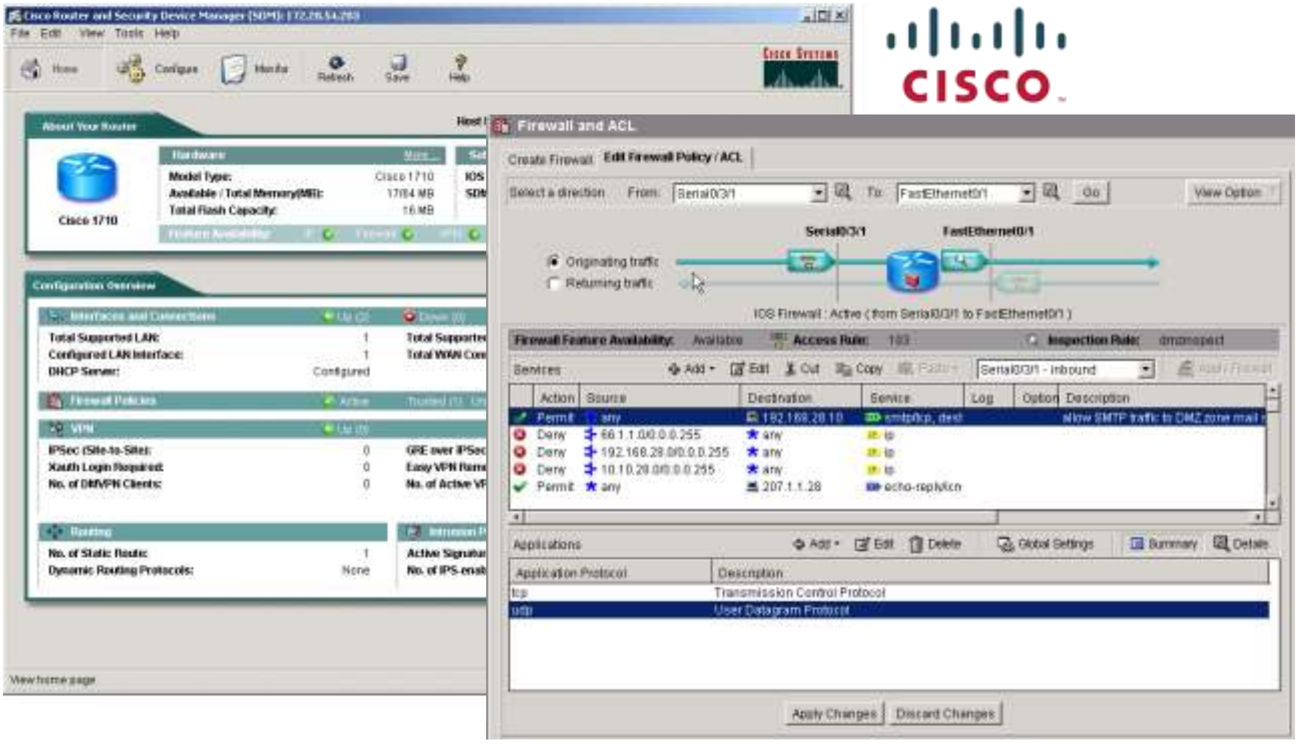
Check results of characterisation:

```
Router# show ip access-list
Extended IP access list ICMP_TRAFFIC_CHAR_e0
   permit icmp any any echo (10 matches)
   permit icmp any any echo-reply (5278 matches)
   permit ip any any (22687 matches)
```

The counters show there is a possible ICMP flood or Smurf attack occurring. The results from e0 show that there were very few echo packets, but a lot of echo-reply packets coming from Bobs network. The interface to Alice's subnet would also be investigated, and it would be realised that a Smurf Attack was taking place. The Smurf Attack is being created by Eve, sending directed broadcast echo packets to Bobs network (the reflector), with spoofed source addresses of 146.5.1.1 - Alice's machine.

Configuring Cisco ACLs with SDM

Cisco Security Device manager is a web-based GUI which can be used to configure security on routers. Wizards are provided, with default rules for several basic topologies which can be configured, or new rules added. The GUI can then be used, as an alternative to the CLI to create and maintain ACLs on all the devices interfaces. The router stateful application inspection firewall functionality (CBAC) can also be configured through the SDM interface.



Cisco Router and Security Device Manager (SDM): 172.28.54.203
File Edit View Tools Help
Home Configure Monitor Refresh Save Help
About Your Router
Cisco 1710
Configuration Overview
Firewall and ACL
Create Firewall
Edit Firewall Policy / ACL
Serial0/3/1
FastEthernet0/1
Apply Changes
Discard Changes
View home page

Hardware	
Model Type:	Cisco 1710
Available / Total Memory(MB):	17/64 MB
Total Flash Capacity:	16 MB

Interfaces and Connections	
Total Supported LAN:	1
Configured LAN Interface:	1
DHCP Server:	Configured

VPN	
IPSec (Site-to-Site):	0
Xauth Login Required:	0
No. of DMVPN Clients:	0

Routing	
No. of Static Route:	1
Dynamic Routing Protocols:	None

Select a direction From: Serial0/3/1 To: FastEthernet0/1 Go

● Originating traffic
○ Returning traffic

IOS Firewall : Active (from Serial0/3/1 to FastEthernet0/1)

Firewall Feature Availability: Available Access Rule: 103 Inspection Rule: dmzinspect

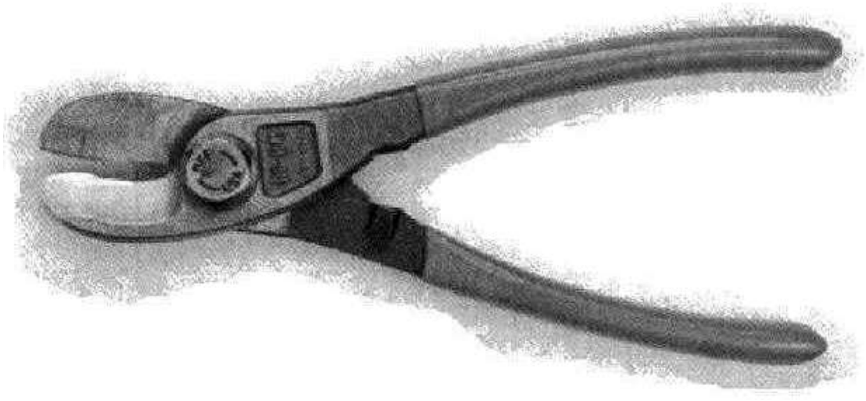
Services: Serial0/3/1 - inbound

Action	Source	Destination	Service	Log	Option	Description
Permit	any	192.168.28.10	smtp/tcp, dest			allow SMTP traffic to DMZ zone mail
Deny	66.1.1.0/0.0.0.255	any	ip			
Deny	192.168.28.0/0.0.0.255	any	ip			
Deny	10.10.28.0/0.0.0.255	any	ip			
Permit	any	207.1.1.28	echo-reply/icn			

Applications

Application Protocol	Description
tcp	Transmission Control Protocol
udp	User Datagram Protocol

Figure 12 Cisco SDM Configuring ACLs



The ULTIMATELY Secure Firewall

For Internet use install the firewall between the demarc of the T1 to the Internet. Place the jaws of the firewall across the T1 line lead, and bear down firmly. When your Internet service provider's network operations center calls to inform you that they have lost connectivity to your site, the firewall is correctly installed.

The fact is, that if you're connecting your network to anything else, you're running a risk. Period. Usually, that risk can be reduced, often dramatically, by employing basic security precautions such as firewalls. But a firewall is a risk reduction system, it is not a risk mitigation system -- there is, always, some danger that something can go fatally wrong with anything built by humans.

The firewall above is the only 100% guaranteed secure solution.

Marcus J. Ranum

5.3 References

1. *The Design of a Secure Internet Gateway.* **Cheswick, Bill AT&T Labs.** 1990.

2. **Badham, John.** *Wargames - http://www.imdb.com/title/tt0086567/.* Leonard Goldberg Production, 1983.

3. War dialing gets an upgrade. *SecurityFocus* . [Online] April 2009. http://www.securityfocus.com/brief/918.

4. **Stuart McClure, Joel Scambray, George Kurtz.** *Hacking Exposed 6.* s.l. : McGraw Hill, 2009.

5. Checkpoint Software Firewall. *Checkpoint.* [Online] Checkpoint. http://www.checkpoint.com/products/softwareblades/firewall.html.

6. **IANA.** IANA - Port numbers. *IANA.* [Online] http://www.iana.org/assignments/port-numbers.

7. **Hamed, Ehab Al-Shaer and Hazem.** Taxonomy of Conflicts in Network Security Policies. *IEEE Communications Magazine.* 2006, Vol. 44.

8. **Alfaro, Frederic Cuppens and Nora Cuppens-Boulahia and Joaquin G.** Detection and Removal of Firewall Misconfiguration. *Conference On Communication, Network, and Information Security (CNIS).* 2005.

9. **Schneier, Bruce.** *Secrets and Lies - Digital Security in a Networked World.* s.l. : Wiley, 2000.

10. **Ranum, Marcus.** What is "Deep Inspection". *Marcus Ranum.* [Online] http://www.ranum.com/security/computer_security/editorials/deepinspect/.

11. **BLOXX.** BLOXX Products. *BLOXX Web Site.* [Online]

12. **Microsoft.** ISA Server Home Page. *ISA Server.* [Online] http://www.microsoft.com/forefront/edgesecurity/isaserver/en/us/default.aspx.

13. **Schneier, Bruce.** Crypto-Gram Newsletter - May 15, 2000. *Bruce Schneier.* [Online] http://www.schneier.com/crypto-gram-0005.html.

14. RFC 1918. *IETF.* [Online] http://tools.ietf.org/html/rfc1918.

15. **IANA.** IANA - ICMP. *IANA.* [Online] http://iana.org/assignments/icmp-parameters.

16. **IETF.** RFC 4890. *IETF.* [Online] http://www.ietf.org/rfc/rfc4890.txt.

17. **Paquet, Catherine.** *Implementing Cisco IOS Network Security (IINS): Authorized Self-study Guide: CCNA.* s.l. : Cisco Press, 2009.

18. **Cisco.** Cisco ASA 5500 Series. *Cisco.* [Online] http://www.cisco.com/en/US/products/ps6120/index.html.

19. —. ASA Application Protocol Inspection. *Cisco Web Site.* [Online] http://www.cisco.com/en/US/docs/security/asa/asa72/configuration/guide/inspect.html#wp1383691.

20. —. Cisco IOS Security Command Reference. *Cisco Web Site.* [Online] http://www.cisco.com/en/US/docs/ios/security/command/reference/sec_book.html.

21. **Sedayao, Jeff.** *Cisco IOS access lists.* s.l. : O'Reilly, 2001.

22. **Bellovin, SM and Cheswick, WR.** Network Firewalls. *IEEE Communications Magazine.* 1994, Vol. 32, 9.

23. **Elizabeth D. Zwicky, Simon Cooper, D. Brent Chapman.** *Building Internet firewalls 2nd Edition.* s.l. : O'Reilly, 2000.