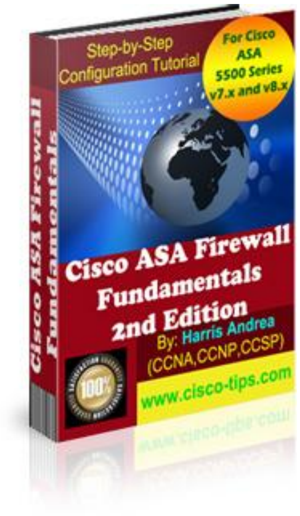
CISCO ASA FIREWALL FUNDAMENTALS 2ND EDITION


Step-by-Step Configuration Tutorial
For Cisco ASA 5500 Series v7.x and v8.x
Cisco ASA Firewall Fundamentals 2nd Edition
By: Harris Andrea (CCNA,CCNP,CCSP)
www.cisco-tips.com
Cisco ASA Firewall Fundamentals
100%

EVERYTHING YOU NEED TO KNOW TO CONFIGURE AND IMPLEMENT

THE BEST FIREWALL IN THE MARKET

WRITTEN BY: HARRIS ANDREA

MSC ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

CISCO CERTIFIED NETWORK PROFESSIONAL (CCNP)

CISCO CERTIFIED SECURITY PROFESSIONAL (CCSP)

CCNP CISCO CERTIFIED ® CCSP CISCO CERTIFIED ™

http://www.cisco-tips.com

ABOUT THE AUTHOR:

Harris Andrea is a Senior Network Security Engineer in a leading Internet Service Provider in Europe. He graduated from the University of Kansas USA in 1998 with a B.S and M.S degrees in Electrical Engineering and Computer Science. Since then, he has been working in the Networking field, designing, implementing and managing large scale networking projects with Cisco products and technologies. His main focus is on Network Security based on Cisco PIX/ASA Firewalls, Firewall Service Modules (FWSM) on 6500/7600 models, VPN products, IDS/IPS products, AAA services etc. To support his knowledge and to build a strong professional standing, Harris pursued and earned several Cisco Certifications such as CCNA, CCNP, and CCSP. He is also a technology blogger owing two networking blogs which you can visit for extra technical information and tutorials.

http://www.cisco-tips.com
http://www.tech21century.com

INTRODUCTION:

Thank you for purchasing this technical eBook about configuring Cisco ASA Firewalls. I firmly believe that you have made an important step towards your career in network security, which is a highly developing and profitable field in the networking area.

Information Security threats are on the rise, and although several products and technologies have been developed to mitigate these threats, the long-proven and trusted hardware firewall is still the heart of security for any network. Firewall administrators and designers are therefore in high demand. Cisco has the biggest market share in the hardware firewall market, so by learning to configure and implement one of the best firewall appliances you are guaranteed a successful career in this field.

This eBook is the result of my working experience with the Cisco Adaptive Security Appliance (ASA), and summarizes the most important features and most frequent configuration scenarios that a security engineer will encounter most of the times. I have tried to "squeeze" the vast volume of information about Cisco ASA firewalls into a handy, directly applicable handbook that will get you on track right away. You can use this eBook in conjunction with other documentation resources or as a quick reference guide for the most common configuration concepts of the Cisco ASA Firewall.

The second Edition ebook contains additional topics (Chapters 7 to 11) that focus on more advanced features of the ASA appliance which were not covered in the first edition book. Therefore this second edition version will be a valuable resource for both beginners and for advanced and experienced ASA firewall administrators. I believe that with the second edition ebook a Cisco professional will get the most complete experience about ASA firewalls.

The last Chapter is dedicated to providing complete real-life configuration examples. These will bind together all the concepts and knowledge presented in the previous Chapters, and will help you build a complete picture of configuring an ASA Firewall in different network topologies.

For any questions that you may have or clarifications about the information presented in this eBook, please contact me at: asaebook@cisco-tips.com

Have fun reading my eBook. I hope it will be a valuable resource for you.

LEGAL NOTICE: The information services and resources provided in this eBook are based upon the current Internet environment as well as the author's experience. The techniques presented here have been proven to be successful. Because technologies are constantly changing, the configurations and examples presented in this eBook may change, cease or expand with time. We hope that the skills and knowledge acquired from this eBook will provide you with the ability to adapt to inevitable evolution of technological services. However, we cannot be held responsible for changes that may affect the applicability of these techniques. The opinions expressed in this ebook belong to the author and are not necessarily those of Cisco Systems, Inc. The author is not affiliated with Cisco Systems, Inc.

All trademarks are trademarks of their respective owners. Rather than put a trademark symbol after every occurrence of a trademarked name, we use names in an editorial fashion only, and to the benefit of the trademark owner, with no intention of infringement of the trademark.

All product names, logos and artwork are copyrights of their respective owners. None of the owners have sponsored or endorsed this publication. While all attempts have been made to verify information provided, the author assumes no responsibility for errors, omissions, or contrary interpretation of the subject matter herein. Any perceived slights of peoples or organizations are unintentional. The purchaser or reader of this publication assumes responsibility for the use of these materials and information. No guarantees of income are made. The author reserves the right to make changes and assumes no responsibility or liability whatsoever on behalf of any purchaser or reader of these materials.

TABLE OF CONTENTS

CHAPTER 1:
Getting Started with Cisco Firewalls

User Interface

This lesson describes the access modes and commands associated with the operation of Cisco security appliances. We assume that you know how to connect to the appliance using a console cable (the blue flat cable with RJ-45 on one end, and DB-9 Serial on the other end) and a Terminal Emulation software (e.g HyperTerminal), and how to use basic Command Line Interface.

Security Appliance Access Modes

A Cisco security appliance (PIX or ASA) has four main administrative access modes:

- **Monitor Mode**: Displays the **monitor>** prompt. A special mode that enables you to update the image over the network or to perform password recovery. While in the monitor mode, you can enter commands to specify the location of a TFTP server and the location of the software image or password recovery binary image file to download. You access this mode by pressing the "Break" or "ESC" keys immediately after powering up the appliance.
- **Unprivileged Mode**: Displays the **>** prompt. Available when you first access the appliance. If the appliance is a Cisco PIX 500 series, the prompt for unprivileged mode is **pixfirewall>** and if the appliance is the new Cisco ASA 5500 Series, the prompt is **ciscoasa>**

  This mode provides restricted view of the security appliance. You cannot configure anything from this mode. To get started with configuration, the first command you need to know is the **enable** command. Type **enable** and hit Enter. The initial password is empty, so hit Enter again to move on the next access mode (Privileged Mode).

```
ciscoasa> enable                ← Unprivileged Mode
password:                       ← Enter a password here (initially its blank)
ciscoasa#                       ← Privileged Mode
```

- **Privileged Mode**: Displays the # prompt. Enables you to change the current settings. Any unprivileged command also works in this mode. From this mode you can see the current configuration by using **show running-config**. Still, you cannot configure anything yet until you go to **Configuration Mode.** You access the Configuration Mode using the **configure terminal** command from the Privileged Mode.

- **Configuration Mode**: This mode displays the **(config)#** prompt. Enables you to change all system configuration settings. Use **exit** from each mode to return to the previous mode.

```
ciscoasa> enable                  ← Unprivileged Mode
password:                         ← Enter a password here (initially its blank)
ciscoasa# configure terminal      ← Privileged Mode
ciscoasa(config)#                 ← Configuration Mode
ciscoasa(config)# exit
ciscoasa# exit                    ← Back to Privileged Mode
ciscoasa>                         ← Back to Unprivileged Mode
```

The **(config)#** mode is sometimes called **Global Configuration Mode**. Some configuration commands from this mode enter a command-specific mode and the prompt changes accordingly. For example the **interface** command enters interface configuration mode as shown below:

```
ciscoasa(config)# interface GigabitEthernet0/1
ciscoasa(config-if)#              ← Configure Interface specific parameters
```

File Management

This lesson describes the file management system in the security appliance.

Viewing and Saving Your Configuration

There are two configuration instances in the Cisco security appliances: **running-configuration** and **startup-configuration**.

The first one (running-configuration) is the one currently running on the appliance, and its stored in the RAM of the firewall. You can view this configuration by typing **show running-config** from the Privileged Mode. Any command that you enter in the firewall is directly written in the running-config and takes effect immediately. Since the running-config is written in the RAM memory, if the appliance loses power it will lose also any configuration changes that were not previously saved. To save the currently running configuration use the command **copy run start** or **write memory.** These two commands copy the running-config into the startup-config which is stored in Flash Memory.

As mentioned above, the **startup-configuration** is the backup configuration of the running one. It is stored in Flash Memory, so it is not lost when the appliance is rebooted. Also, the **startup-configuration** is the one which is loaded when the appliance boots-up. To view the stored startup-configuration type **show startup-config.**

SECURITY LEVELS

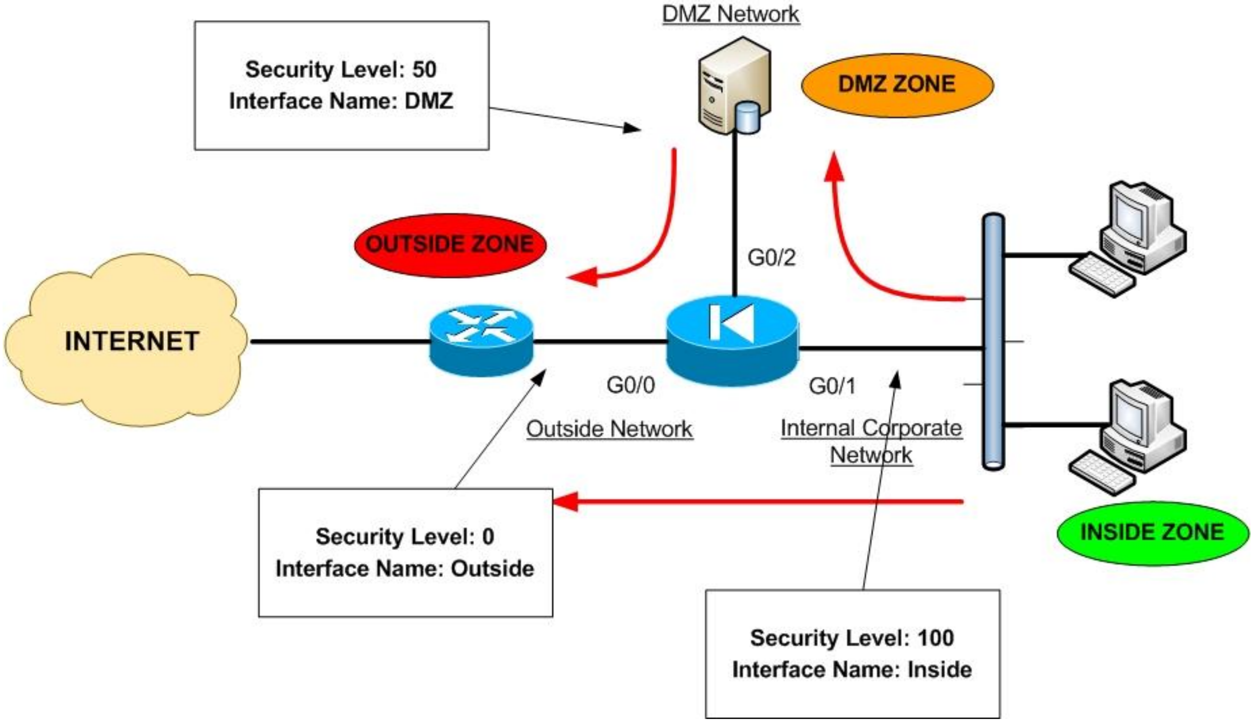
This lesson describes the security levels concept as used in the firewall appliance.

The Security Level is assigned to interfaces (either physical or logical sub-interfaces) and it is basically a number from 0 to 100 designating how trusted an interface is relative to another interface on the appliance. The higher the security level, the more trusted the interface (and hence the network connected behind it) is considered to be, relative to another interface. Since each firewall interface represents a specific network (or security zone), by using security levels we can assign 'trust levels' to our security zones. The primary rule for security levels is that an interface (or zone) with a higher security level can access an interface with a lower security level. On the other hand, an interface with a lower security level cannot access an interface with a higher security level, without the explicit permission of a security rule (Access Control List - ACL).

SECURITY LEVEL EXAMPLES

Let us see some examples of security levels below:

- **Security Level 0**: This is the lowest security level and it is assigned by default to the '**Outside**' Interface of the firewall. It is the least trusted security level and must be assigned accordingly to the network (interface) that we don't want it to have any access to our internal networks. This security level is usually assigned to the interface connected to the Internet. This means that every device connected to the Internet can not have access to any network behind the firewall, unless explicitly permitted by an ACL rule.
- **Security Levels 1 to 99**: These security levels can be assigned to perimeter security zones (e.g. DMZ Zone, Management Zone, Database Servers Zone etc).
- **Security Level 100**: This is the highest security level and it is assigned by default to the '**Inside**' Interface of the firewall. It is the most trusted security level and must be assigned accordingly to the network (interface) that we want to apply the most protection from the security appliance. This security level is usually assigned to the interface connecting the Internal Corporate network behind it.



The diagram above illustrates a typical example of security levels assignment in a network with an Inside, Outside, and DMZ zones. Throughout this book we will represent the Cisco Firewall with the "Electrical Diode" symbol. As you can see, the Internal Corporate Network is connected to the Interface with the highest security level (Interface G0/1 with Security Level 100) which is also named as 'Inside'. The Interface name 'Inside' is given by default to the interface with the highest security level. Also, the INTERNET facing interface (G0/0) is named 'Outside' and is assigned a security level of 0. A Perimeter Zone (DMZ) is also created with a Security Level of 50. The Red Arrows in the diagram represent the flow of traffic. As you can see, the Inside Zone can access both DMZ and Outside Zones (Security Level 100 can access freely the Security Levels 50 and 0). The DMZ Zone can access only the Outside Zone (Security Level 50 can access Level 0), but not the Inside Zone. Lastly, the Outside Zone cannot access either the Inside or the DMZ zones.

What is described in the example above is the default behavior of the Cisco PIX/ASA Firewalls. We can override the default behavior and allow access from Lower Security Levels to Higher Security Levels by using Static NAT and Access Control Lists, as we will see in the next chapters of this book.

RULES FOR TRAFFIC FLOW BETWEEN SECURITY LEVELS

- **Traffic from Higher Security Level to Lower Security Level**: Allow ALL traffic originating from the higher Security Level unless specifically restricted by an Access Control List (ACL). If NAT-Control is enabled on the device, then there must be a **nat/global** translation pair between High-to-Low Security Level interfaces.
- **Traffic from Lower Security Level to Higher Security Level:** Drop ALL traffic unless specifically allowed by an ACL. If NAT-Control is enabled on the device (more on this later), then there must be a **Static NAT** between High-to-Low Security Level interfaces.
- **Traffic between interfaces with same Security Level:** By default this is not allowed, unless you configure the **same-security-traffic permit** command (ASA version 7.2).

BASIC FIREWALL CONFIGURATION

BASIC CONFIGURATION STEPS

The following configuration commands constitute the basis for setting up the security appliance from the ground up:

- **STEP1: Configure a privileged level password (enable password)**

By default there is no password for accessing the ASA firewall, so the first step before doing anything else is to configure a privileged level password, which will be needed to allow subsequent access to the appliance. Configure this under Configuration Mode:

```
ciscoasa(config)# enable password mysecretpassword
```

- **STEP2: Enable Command Line Management**

You can access the security appliance remotely for Command Line Interface management (CLI) using either Telnet or SSH, and for Web-based graphical management using HTTPS (ASDM management). It is recommended to use SSH for CLI management since all communication with the firewall will be encrypted, compared with using Telnet which is not encrypted. To enable SSH on the firewall, we need first to create a username/password for authentication, then generate encryption keys (RSA keys), and also specify the IP address of the management host/network.

```
! Create a username "ciscoadmin" with password "adminpassword" and use this LOCAL username
!to authenticate for SSH connections
ciscoasa(config)#username ciscoadmin password adminpassword
ciscoasa(config)#aaa authentication ssh console LOCAL

! Generate a 1024 bit RSA key pair for the firewall which is required for SSH
ciscoasa(config)# crypto key generate rsa modulus 1024
Keypair generation process begin. Please wait...
ciscoasa(config)#

! Specify the hosts allowed to connect to the security appliance.
ciscoasa(config)#ssh 10.1.1.1 255.255.255.255 inside
ciscoasa(config)#ssh 200.200.200.1 255.255.255.255 outside
```

- **STEP3: Configure a Firewall Hostname**

The default hostname for Cisco ASA appliances is **ciscoasa**, and for the Cisco PIX appliance is **pixfirewall**. It is advisable to configure a unique hostname for a new firewall so that you can differentiate it from other firewalls that you may have in the network.

```
ciscoasa(config)# hostname NewYork-FW
NewYork-FW(config)#
```

Notice how the CLI prompt has changed to the new Hostname that you just configured.

- **STEP4: Configure Interface Commands**

The Cisco ASA interfaces are numbered as **GigabitEthernet0/0, GigabitEthernet0/1, GigabitEthernet0/2** etc (for Cisco ASA 5510 model, the interfaces are numbered as Etherne0/0, Ethernet0/1 etc). The "**Interface**" command will put you into a special configuration mode for the interface you specify (**interface configuration mode**), and then allow you to configure other interface sub-commands inside the interface mode. For Cisco ASA 5505, the interface commands are configured under the **"Interface Vlan x"** mode.

```
ciscoasa(config)# interface GigabitEthernet0/1
ciscoasa(config-if)# ← Configure Interface specific sub-commands
```

For Cisco ASA 5505:

```
ciscoasa(config)# interface Vlan [vlan number]
ciscoasa(config-if)# ← Configure Interface specific sub-commands
```

The absolutely necessary Interface Sub-commands that you need to configure in order for the interface to pass traffic are the following:

- **nameif "*interface name*":** Assigns a name to an interface
- **ip address "*ip_address*" "*subnet_mask*" :** Assigns an IP address to the interface
- **security-level "*number 0 to 100*" :** Assigns a security level to the interface
- **no shutdown :** By default all interfaces are shut down, so enable them.

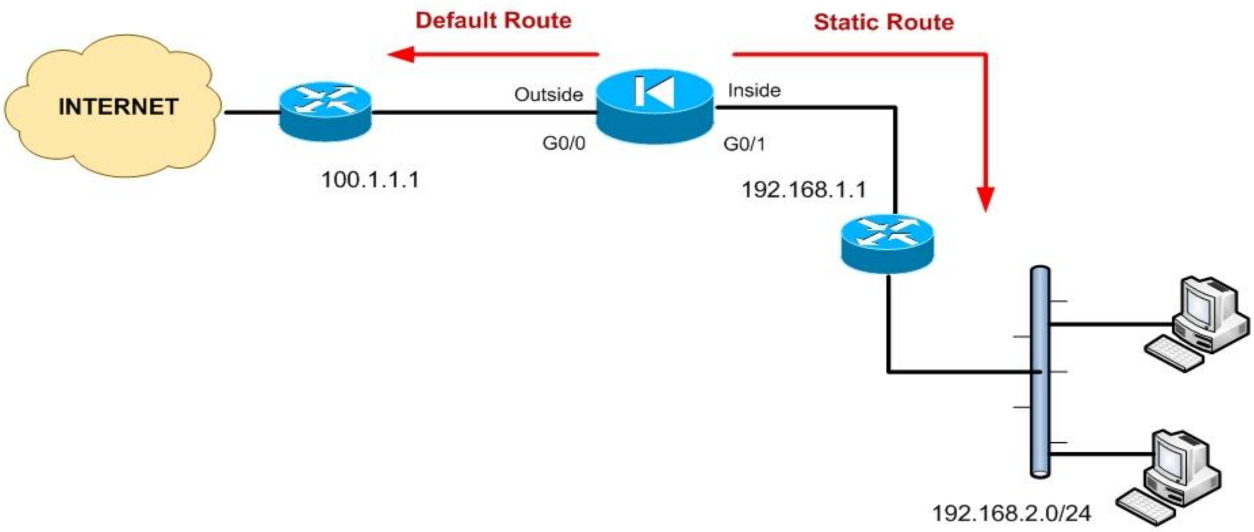
The configuration snapshot below shows all necessary interface sub-commands:

```
ciscoasa(config)# interface GigabitEthernet0/1
ciscoasa(config-if)# nameif inside
ciscoasa(config-if)# ip address 10.0.0.1 255.255.255.0
ciscoasa(config-if)# security-level 100 ← By default "inside" interface is sec-level 100
ciscoasa(config-if)# no shutdown

ciscoasa(config)# interface GigabitEthernet0/0
ciscoasa(config-if)# nameif outside
ciscoasa(config-if)# ip address 10.1.1.1 255.255.255.0
ciscoasa(config-if)# security-level 0 ← By default "outside" interface is sec-level 0
ciscoasa(config-if)# no shutdown
```

- **STEP5: Configure NAT Control as needed**

Another important configuration step is **nat-control**. NAT (Network Address Translation) was a mandatory configuration in older Cisco PIX firewalls (PIX version 6.x) but with ASA Firewalls it is not. **Nat-Control** (which is disabled by default) specifies whether or not the security appliance will enforce address hiding (i.e address translation) to ALL traffic passing from a high security level to a lower one. If you stay with the default configuration (i.e **nat-control** is disabled), this will allow you to apply NAT (address hiding) to only selected traffic as you require. If you enable nat-control (using the command: **asa(config)#nat-control**) then you MUST have a NAT rule for ALL traffic passing from a high security interface to a lower security interface. The NAT rule must match a corresponding "**global**" command (more on NAT later).

- **STEP6: Configure routing**



Routing is an essential step to configure, otherwise the Firewall appliance will not know how to send traffic to its destination. In the example above we show only default and static routing although dynamic routing protocols (RIP, OSPF, EIGRP) can be configured also. My recommendation is to use only default or static routing and avoid dynamic protocols in small networks. However, dynamic routing protocols on the ASA are also useful in larger and complex networks. More on dynamic routing protocols in a later Chapter.

Use the **route** command to enter either a static or default route for an interface. The command format is:

ciscoasa(config)# route "*interface-name*" "*destination-ip-address*" "*netmask*" "*gateway*""

Let's see an example configuration below:

ciscoasa(config)# route outside 0.0.0.0 0.0.0.0 100.1.1.1 ← **Default Route**
ciscoasa(config)# route inside 192.168.2.0 255.255.255.0 192.168.1.1 ← **Static Route**

For the default route (usually towards the Internet), you set both the *destination-ip-address* and *netmask* to 0.0.0.0. Create also static routes to access specific known networks beyond the locally connected networks, as shown on the diagram above.

The routing configuration concludes the “Minimum Mandatory” steps needed for the security appliance to become operable. Next we will get into more details for further configuration features that will enhance the security of the networks protected by the firewall.

CHAPTER 2:
CONFIGURING TRANSLATIONS

In this Chapter we will talk about a very important security mechanism that has to do with IP address translation (address hiding), its different types, and how the firewall appliance handles the translation mechanisms.

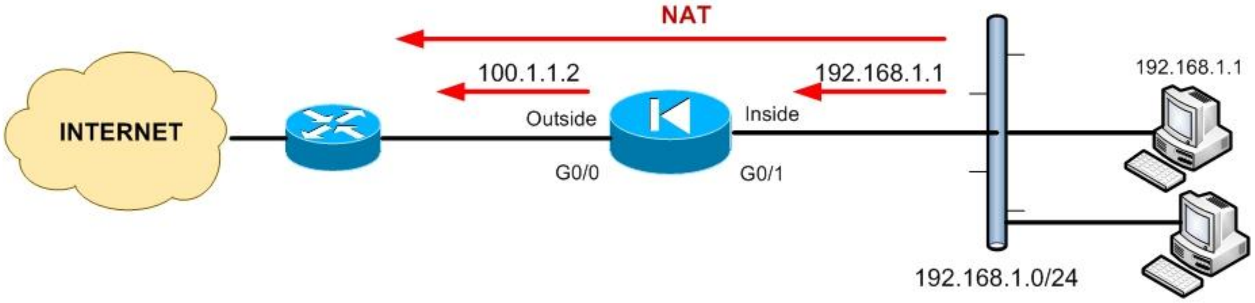
NETWORK ADDRESS TRANSLATION (NAT)

The depletion of the public IPv4 address space has forced the Internet Community to think about alternative ways of addressing networked hosts. NAT therefore was created to overcome these addressing problems that occurred with the expansion of the Internet.

Some of the advantages of using NAT in IP networks are the following:

- NAT helps to mitigate global public IP address depletion.
- Networks can use the RFC 1918 private address space internally.
- NAT increases security by hiding the internal network topology and addressing.

The figure below shows a basic topology with an "inside" network for which the ASA Firewall performs a NAT action to translate the "inside" address into an "outside" address, thus hiding the internal IP range. Note that the translation is applied to the "source" IP address of the packets.



NAT is always used for OUTBOUND traffic, that is, traffic from an internal network (higher security level) towards an outside network (lower security level). In the figure above, traffic from the host with private IP address 192.168.1.1 is translated into a public, routable address, 100.1.1.2 in order to be routed towards the Internet. Now, the reply packets from the Internet back to our internal host, will have as destination address the IP 100.1.1.2, for which the firewall already has a translation rule established. The firewall will then translate the public address 100.1.1.2 back into 192.168.1.1 and deliver it to the internal host.

The **nat** and **global** commands work together to create the translation rules which enable your internal network to use any IP addressing scheme and at the same time remain hidden from the outside world.

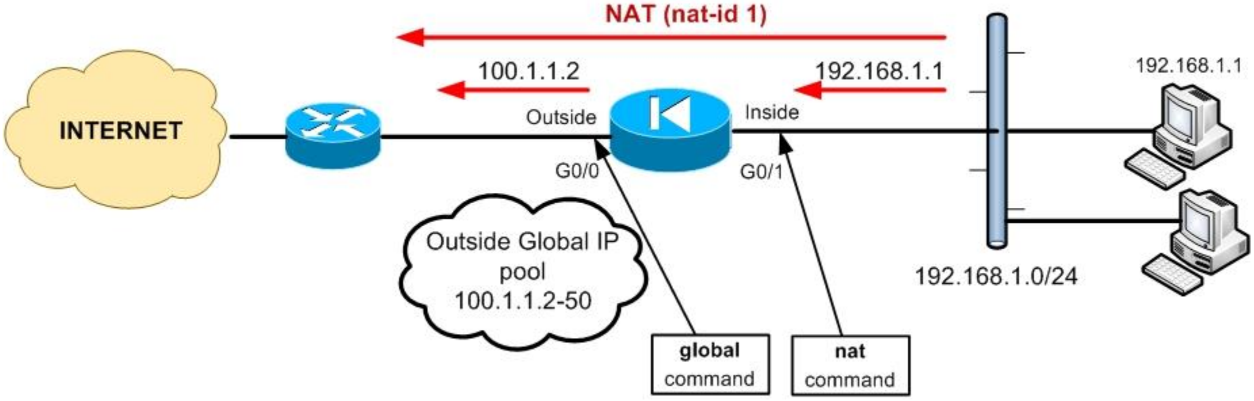
Cisco ASA firewalls support two main types of address translations:

- **Dynamic NAT translation**: Translates source addresses on higher security interfaces into a range (or pool) of IP addresses on a less secure interface, for outbound connections. The **nat** command defines which internal hosts will be translated, and the **global** command defines the address pool on the outgoing interface.
- **Static NAT translation**: Provides a permanent, one-to-one address mapping between an IP on a more secure interface and an IP on a less secure interface. With the appropriate Access Control List (ACL), static NAT allows hosts on a less secure interface (e.g Internet) to access hosts on a higher security interface (e.g Web Server on DMZ) without exposing the actual IP address of the host on the higher security interface.

In this section we will describe **Dynamic NAT translation** with several scenarios. The format of the nat and global commands as used in Dynamic NAT is shown below:

```
ciscoasa(config)# nat (internal_interface_name) “nat-id” “internal network IP subnet”
ciscoasa(config)# global (external_interface_name) “nat-id” “external IP pool range”
```

Scenario 1: Simple Dynamic Inside NAT Translation

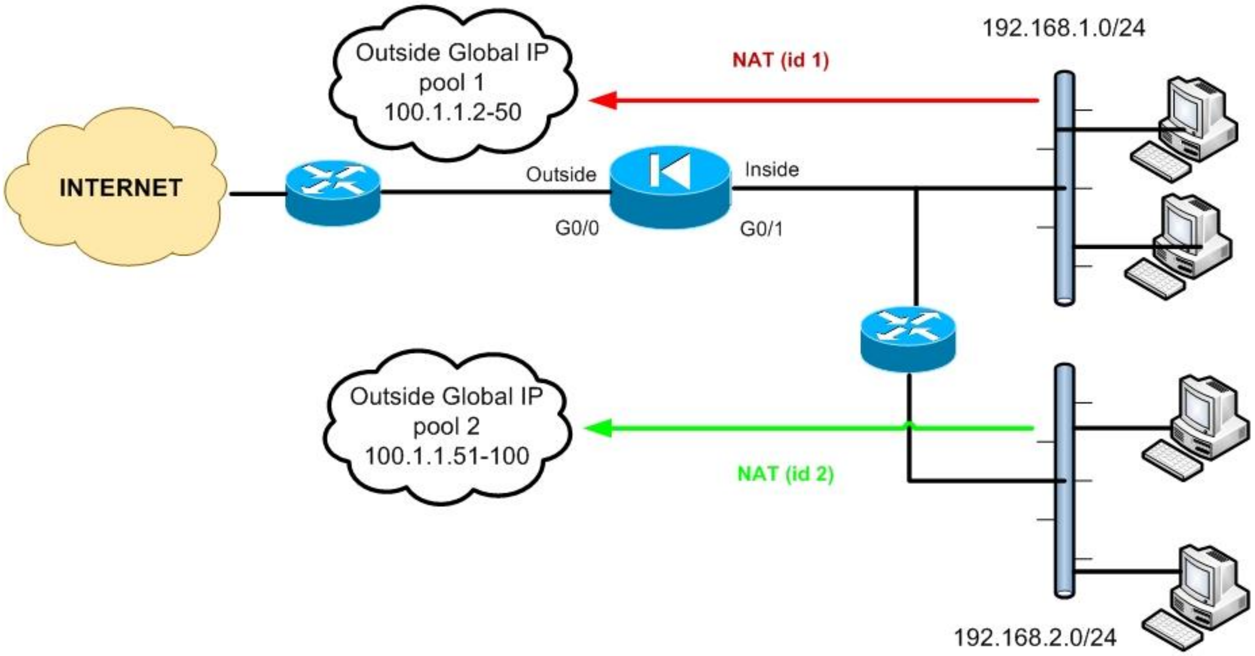


ciscoasa(config)# nat (inside) 1 192.168.1.0 255.255.255.0 ←Inside net to be translated
ciscoasa(config)# global (outside) 1 100.1.1.2-100.1.1.50 netmask 255.255.255.0 ←Outside pool

In the scenario above the firewall will perform dynamic NAT to all inside hosts (192.168.1.0/24). The IP addresses of outbound traffic from inside to outside will be translated into addresses from the Outside Global pool 100.1.1.2 up to 100.1.1.50. Notice the **nat-id** value (**1**). This number binds the **nat** command with the **global** command. Its importance will be clearer in our next scenarios.

Also note the names "**inside**" and "**outside**" used in the **nat** and **global** commands. These names are the ones assigned under the interface configuration with the "**nameif**" command.

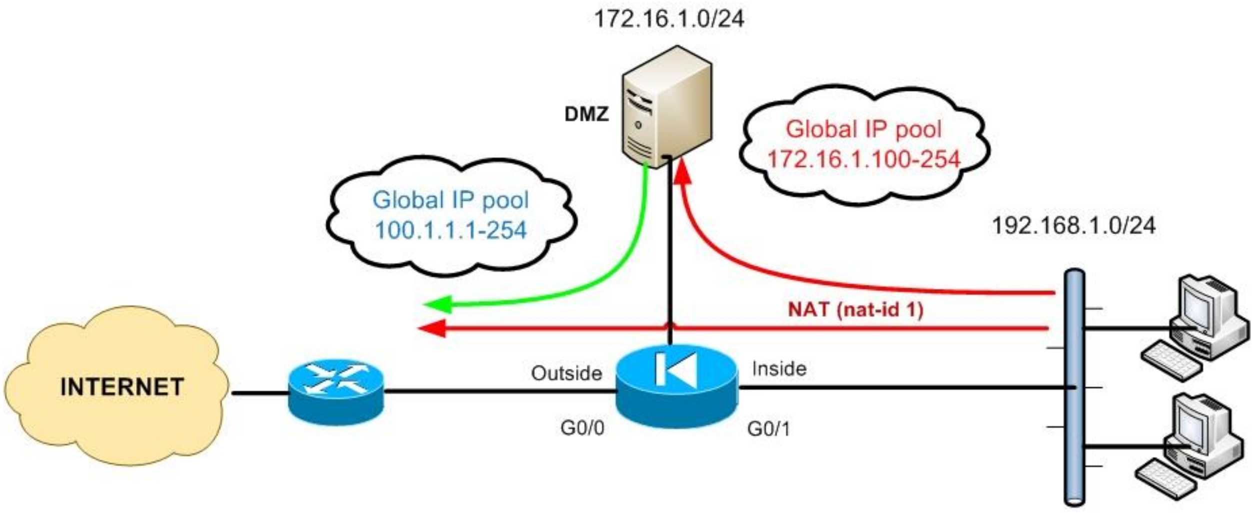
Scenario 2: Dynamic NAT Translation of two internal networks



ciscoasa(config)# nat (inside) 1 192.168.1.0 255.255.255.0 ←**First Internal Network**
ciscoasa(config)# nat (inside) 2 192.168.2.0 255.255.255.0 ←**Second Internal Network**
ciscoasa(config)# global (outside) 1 100.1.1.2-100.1.1.50 netmask 255.255.255.0
ciscoasa(config)# global (outside) 2 100.1.1.51-100.1.1.100 netmask 255.255.255.0

The scenario here shows the importance of the **nat-id** parameter and how this is used to bind together a **nat/global** command pair. The **nat-id** (**1**) in the first **nat** command statement tells the firewall to translate the internal network 192.168.1.0/24 addresses into those in the mapped global IP pool containing the same nat-id (i.e 100.1.1.2 up to 100.1.1.50). Similarly, the nat-id (**2**) in the second **nat** statement tells the firewall to translate addresses for hosts in 192.168.2.0/24 to the addresses in the mapped global pool 2 with nat-id (**2**) (i.e 100.1.1.51 up to 100.1.1.100).

Scenario 3: Dynamic NAT Translation with three interfaces



```
ciscoasa(config)# nat (inside) 1 192.168.1.0 255.255.255.0 ←Inside Subnet
ciscoasa(config)# nat (DMZ) 1 172.16.1.0 255.255.255.0 ←DMZ Subnet
ciscoasa(config)# global (outside) 1 100.1.1.1-100.1.1.254 netmask 255.255.255.0
ciscoasa(config)# global (DMZ) 1 172.16.1.100-172.16.1.254 netmask 255.255.255.0
```

In the scenario above, assume that "inside" interface has security level 100, "DMZ" interface has security level 50, and "outside" interface has security level 0. This means that "inside" hosts can start connections to lower security level interfaces (i.e to both "DMZ" and "outside"). Also, these security levels allow hosts on the DMZ interface to start connections towards the outside interface.

Because both of the mapped pools (global commands) and the **nat(inside)** command use the same nat-id of 1, addresses for hosts on the inside network (192.168.1.0/24) can be translated to those in either mapped pool, depending on the direction of the traffic. Therefore, when hosts on the inside interface access hosts on the DMZ, the **global(DMZ)** command causes their source addresses to be translated to addresses in the range 172.16.1.100 – 172.16.1.254. Similarly, when inside hosts access hosts on the outside, the **global (outside)** command will cause their source addresses to be translated into the range 100.1.1.1 – 100.1.1.254.

Moreover, the configuration above allows also hosts on the DMZ to use NAT when accessing outside hosts. The **nat (DMZ)** together with **global (outside)** commands will cause the source addresses of DMZ hosts (172.16.1.0/24) to be translated into the outside range 100.1.1.1 – 100.1.1.254.

Monitoring NAT Translations
The **ciscoasa# show xlate** command displays the contents of the NAT translation table.

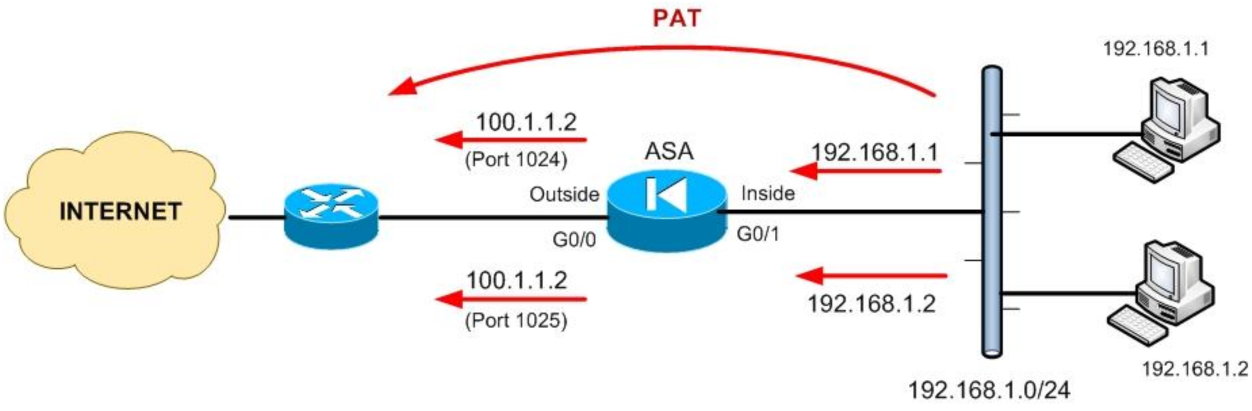
e.g *Global 100.1.1.10 Local 192.168.1.10*

The output above shows that a private local address 192.168.1.10 is assigned a global pool address of 100.1.1.10.

PORT ADDRESS TRANSLATION (PAT)

With Dynamic NAT we assume that we have a range (pool) of public addresses that we use to translate our internal network private addresses. In real situations, an enterprise network receives only a limited number of public addresses from its ISP, whereas the number of internal private addresses is much bigger. This means that if we use Dynamic NAT in such a situation, the external public address pool will be depleted really fast when many internal hosts access the internet simultaneously.

To overcome this problem, we can use a "**many-to-one**" address translation, called also Port Address Translation (PAT). Using PAT, multiple connections from different internal hosts can be **multiplexed** over a single global (public) IP address using different source port numbers. Let's see an example below:



ciscoasa(config)# nat (inside) 1 192.168.1.0 255.255.255.0 ←Inside Subnet to use PAT
ciscoasa(config)# global (outside) 1 100.1.1.2 netmask 255.255.255.255 ← Use a single global IP address for PAT

In the example above, all internal private addresses (192.168.1.0/24) will use a single public IP address (100.1.1.2) with different source port numbers. That is, when host 192.168.1.1 connects on an Internet outside host, the firewall will translate its source address and port into 100.1.1.2 with source port 1024. Similarly, host 192.168.1.2 will be translated again into 100.1.1.2 but with a different source port (1025). The source ports are dynamically changed to a unique number greater than 1023. A single PAT address can support around 64,000 inside hosts.

Monitoring PAT Translations

The **ciscoasa# show xlate** command displays the contents of the PAT translation table.

e.g *PAT Global 100.1.1.2 (1024) Local 192.168.1.1 (4513)*

The output above shows that a connection from the private local address 192.168.1.1 with source port 4513 is translated into address 100.1.1.2 with source port 1024.

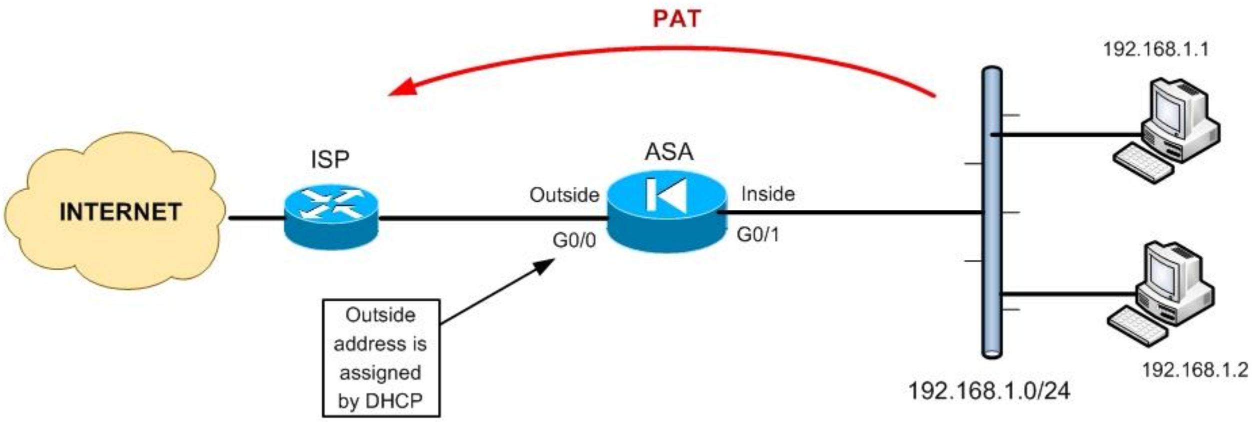
The firewall keeps track of all NAT sessions using its **xlate** table, so that when a reply packet comes back from outside, the firewall will check its translation table to see which port number belongs to the particular reply packet in order to deliver it to the correct internal host.

There are several different scenarios in which PAT can be used in a network. We will describe these next.

Scenario 1: PAT using outside interface IP address

Instead of configuring a specific IP address in the global command to be used for PAT (as the example above), we can specify the outside Interface as the PAT address. This scenario is important when our firewall obtains a dynamic public IP address from the Internet Service Provider (ISP), in which case we don't know the exact address to configure it on the global command.

Refer to the diagram below for a configuration example using DHCP outside address for PAT:

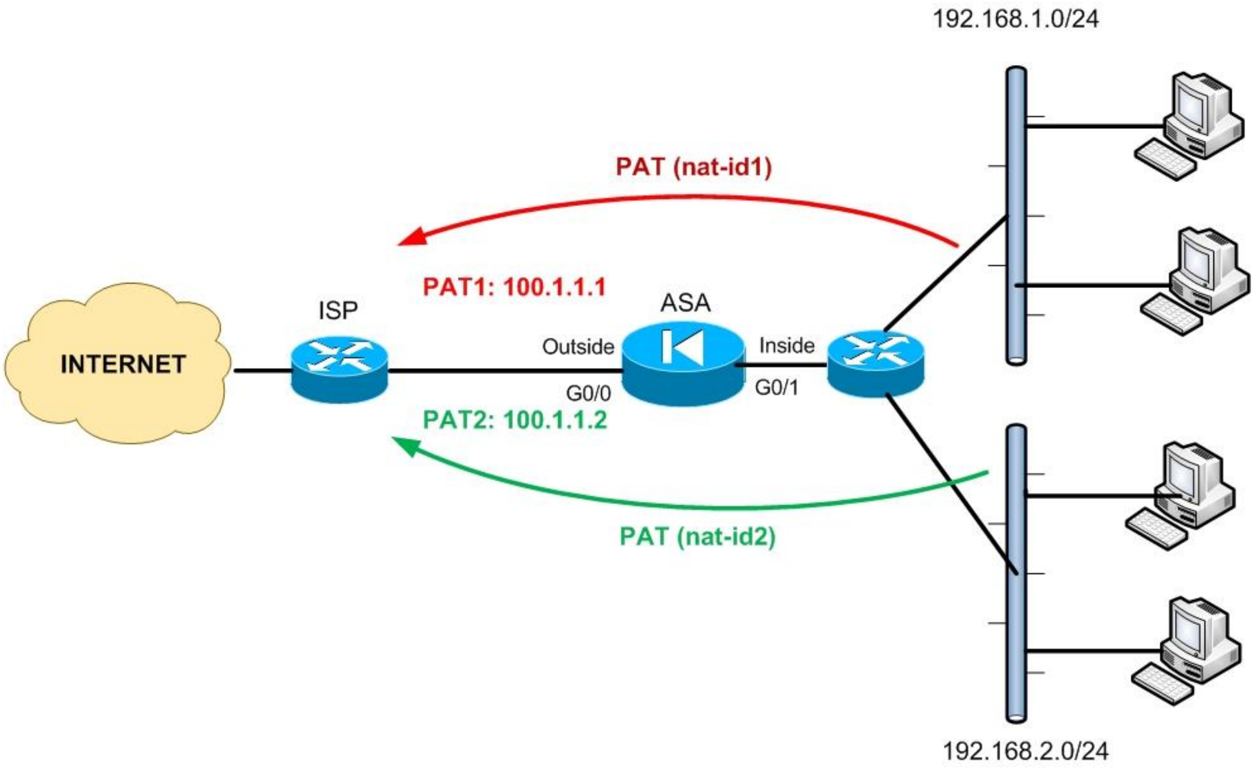


```
ciscoasa(config)# interface G0/0
ciscoasa(config-if)# ip address dhcp setroute ←Get outside address and gateway from ISP
ciscoasa(config)# nat (inside) 1 192.168.1.0 255.255.255.0 ←Inside Subnet to use PAT
ciscoasa(config)# global (outside) 1 interface ← Use the outside IP address for PAT
```

The "**ip address dhcp setroute**" interface command configures the firewall to work as a DHCP client for the ISP and obtain a public address automatically. The "**setroute**" parameter tells the Cisco Firewall to set its default route using the default gateway value that the DHCP server returns. **Do not** configure a default route when using the **setroute** option.

Scenario 2: Mapping different internal subnets to different PAT addresses

Using the **nat-id** parameter we can bind two or more **nat/global** statement pairs in order to map different internal network subnets to different PAT addresses, as shown in the diagram below:



```
ciscoasa(config)# nat (inside) 1 192.168.1.0 255.255.255.0
ciscoasa(config)# global (outside) 1 100.1.1.1 netmask 255.255.255.255

ciscoasa(config)# nat (inside) 2 192.168.2.0 255.255.255.0
ciscoasa(config)# global (outside) 2 100.1.1.2 netmask 255.255.255.255
```

Outbound connections from internal subnet 192.168.1.0/24 will seem to originate from address 100.1.1.1 and outbound connections from subnet 192.168.2.0/24 will seem to originate from address 100.1.1.2.

Scenario 3: Combining Dynamic NAT with PAT Translation

We can use a pool of external public IP addresses for Dynamic NAT translation, and augment this pool with a single PAT address in case the addresses in the global pool are exhausted.

```
ciscoasa(config)# nat (inside) 1 192.168.1.0 255.255.255.0
ciscoasa(config)# global (outside) 1 100.1.1.100-100.1.1.253 netmask 255.255.255.0
ciscoasa(config)# global (outside) 1 100.1.1.254 netmask 255.255.255.255
```

Outbound connections from the internal network 192.168.1.0/24 are assigned addresses from the range 100.1.1.100 up to 100.1.1.253. If the firewall assigns all addresses from its dynamic pool, it will overflow to its PAT address 100.1.1.254.

STATIC ADDRESS TRANSLATION (STATIC NAT)

The two translation types that we discussed in the previous sections (Dynamic NAT and PAT) are used for Outbound communication only (i.e from higher security level to lower security level). However, if an outside host (let's say a host on the Internet) wants to initiate communication to an Internal host behind the firewall, this is not possible if we have only Dynamic NAT or PAT configured. This is very good in terms of security, but there are several cases that we must allow Inbound access as well (i.e access from lower security to higher security levels – Outside to Inside). To achieve this, we MUST use a **Static NAT** translation and also configure an appropriate **Access Control List**. Static NAT maps permanently a host address to a fixed global (outside) address.

The most important reasons to use Static NAT are the following:

- We have an internal server (e.g our company's email or web server) that must always appear with a fixed public IP address on the Outside interface of the firewall.
- We want to allow hosts from the Outside (e.g Internet) to initiate connections to a local internal server (e.g our Web or email server).
- We want to use Port Redirection (more on this later).

The command format of Static NAT is:

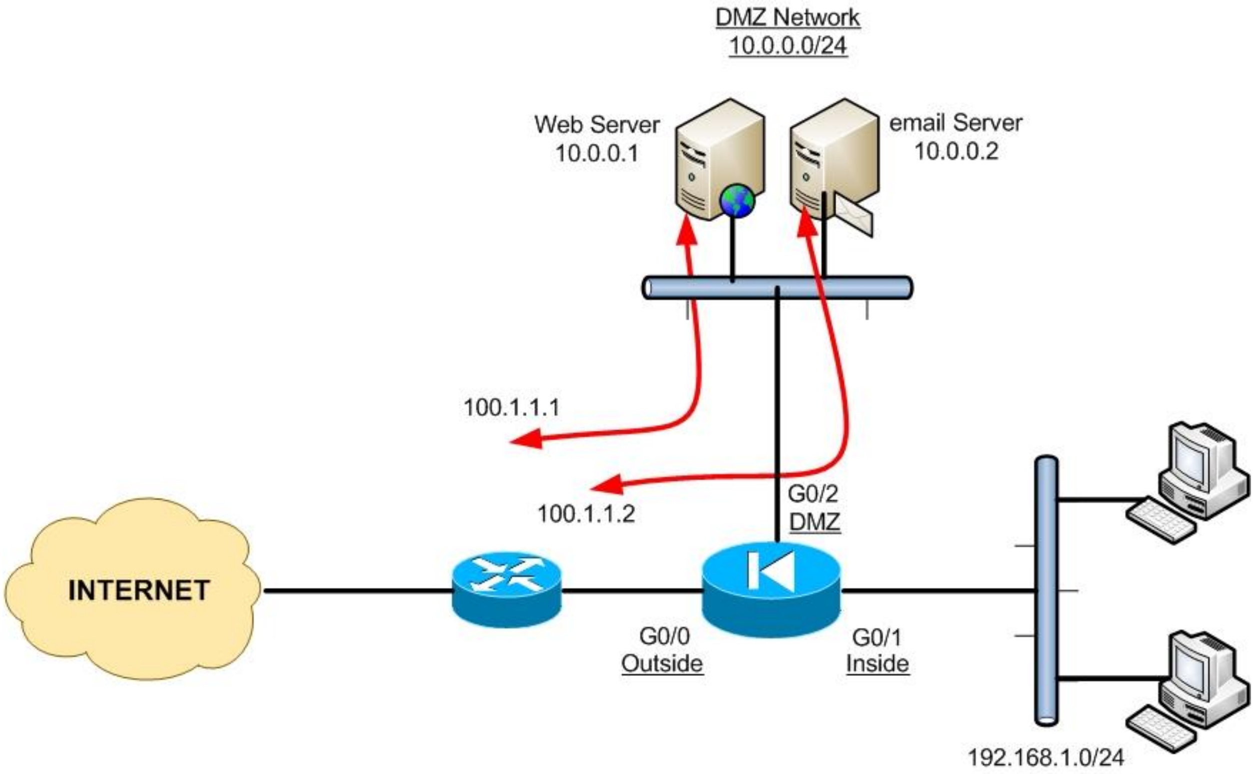
ciscoasa(config)# static (*real_interface_name , mapped_interface_name*) "mapped_IP" "real_IP" netmask "subnet_mask"

To configure static NAT we need to know the following parameters:

1. Between which two interfaces the translation is performed. The two interfaces are defined as the **real_interface** and the **mapped_interface**. The real interface (e.g DMZ interface or Inside interface) must have higher security level than the mapped interface (e.g Outside interface).
2. The **real_IP** address of the host (the IP actually configured on the Network Card of the host).
3. The **mapped_IP** (or translated) IP address of the host (i.e the address that the host will be known to the Outside networks).

A little "catch" that you need to be careful with the **static** command is the following: when entering the interface names in the parenthesis, you enter the **real_interface** name first followed by the **mapped_interface** name (see command format above). However, when you configure the IP addresses after the interface names, you enter the **mapped_IP** address first followed by the **real_IP** address. Let's see some example scenarios for making things clear:

Scenario 1: Static NAT with a Web Server and email Server on DMZ



The network topology above is classic in many enterprises. Usually there is an Inside network on the firewall which hosts all internal employees' computers, an Outside network that connects to the Internet, and there is also a Demilitarized Zone (DMZ) that hosts servers which should be accessible from the Internet (in our example, a Web Server and an email Server). In this scenario static NAT must be used for the DMZ servers so that their real private IP address is always translated to a fixed public IP address (10.0.0.1 translated to 100.1.1.1 and 10.0.0.2 translated to 100.1.1.2).

In our scenario we have the following:

- Real Interface name : **DMZ**
- Mapped Interface name: **Outside**
- Real IP addresses: **10.0.0.1** and **10.0.0.2**
- Mapped IP adresses: **100.1.1.1** and **100.1.1.2**

Let's see the configuration snapshot below:

```
ciscoasa(config)# static (DMZ , outside) 100.1.1.1  10.0.0.1 netmask 255.255.255.255
ciscoasa(config)# static (DMZ , outside) 100.1.1.2  10.0.0.2 netmask 255.255.255.255
```

The above statements enable bi-directional communication for the web and email servers. Now Internet hosts can access our web and email servers via their public address 100.1.1.1 and 100.1.1.2 Of course an ACL is still needed on the outside interface to allow communication.

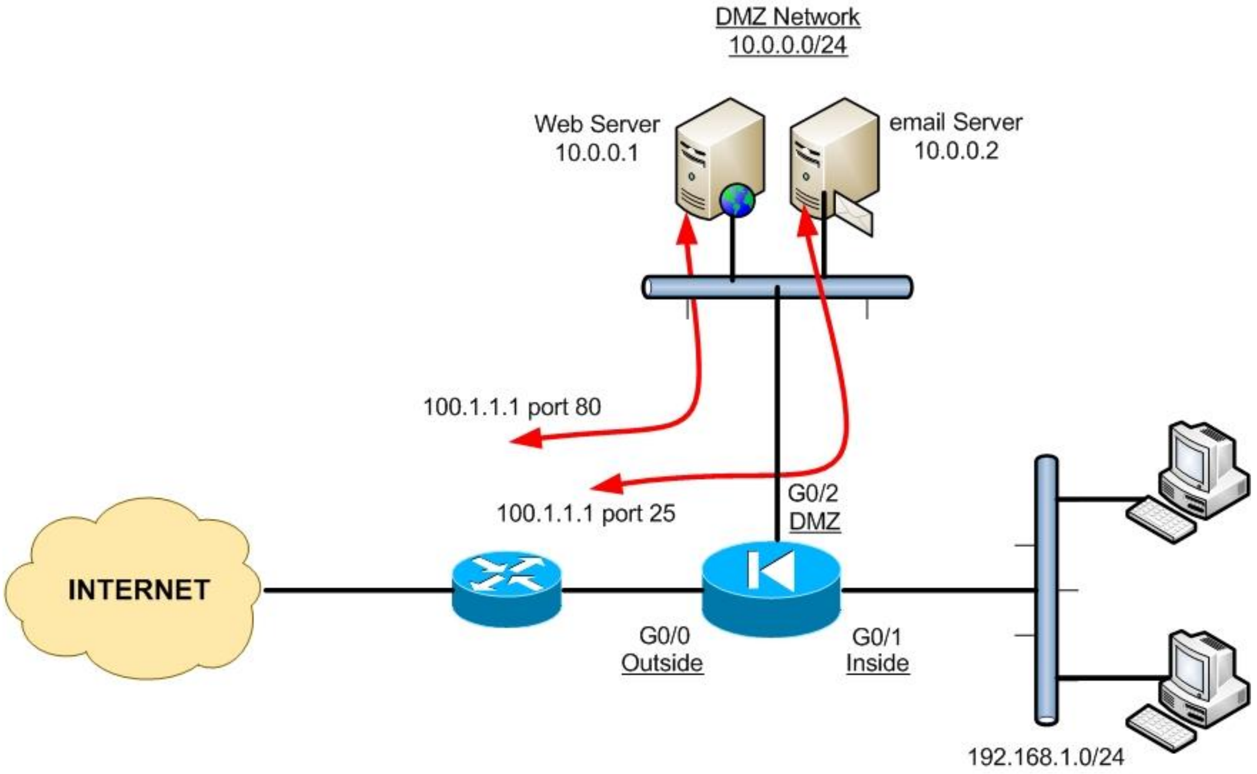
Scenario 2: Static NAT for a whole network range (Net Static)

Instead of permanently translating single host addresses, we can create permanent address mappings to a whole subnet with just one command. Referring to the previous diagram, assume that we have a whole class C public address range 100.1.1.0/24. We can translate the whole DMZ range 10.0.0.0/24 to 100.1.1.0/24.

```
ciscoasa(config)# static (DMZ , outside) 100.1.1.0  10.0.0.0 netmask 255.255.255.0
```

Any packet sourced from a server address on subnet 10.0.0.0/24 on the DMZ will be translated to a host address on the 100.1.1.0/24 subnet on the outside interface (e.g. host 10.0.0.20 will be translated to 100.1.1.20).

Scenario 3: Static Port Address Translation (Port Redirection)



A pretty common scenario is the one shown on the diagram above. Assume we have only one public IP address available (100.1.1.1) but we have two (or more) servers that we need to provide public access for. We know that our Web Server listens to port 80 and our email Server listens to port 25. All inbound traffic hitting address 100.1.1.1 port 80 can be redirected by the firewall to 10.0.0.1 port 80, and all traffic hitting address 100.1.1.1 port 25 will be redirected to 10.0.0.2 port 25.

The command format for Port Redirection is the following:

ciscoasa(config)# static (*real_interface_name , mapped_interface_name*) [tcp|udp] “mapped_IP” “mapped_port” “real_IP” “real_port” netmask “subnet_mask”

For the network topology in our example scenario above, the port redirection commands are the following:

ciscoasa(config)# static (DMZ , outside) tcp 100.1.1.1 80 10.0.0.1 80 netmask 255.255.255.255

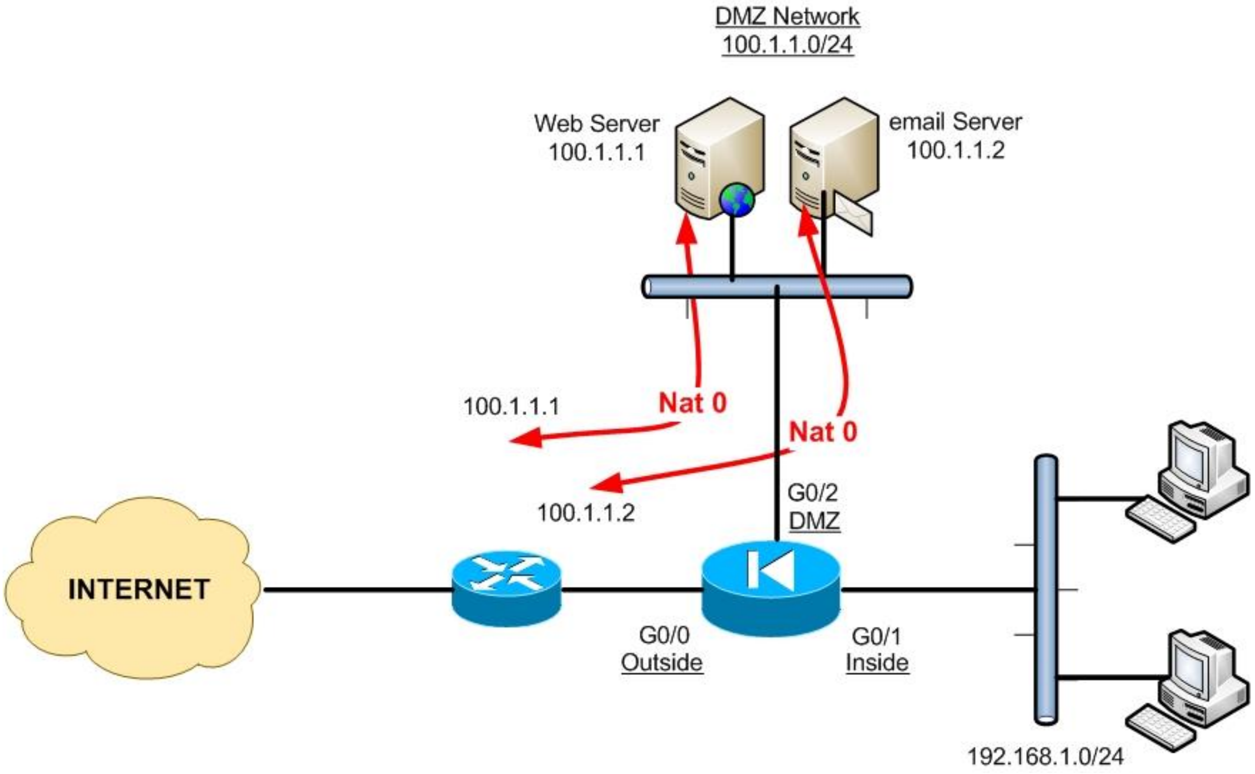
ciscoasa(config)# static (DMZ , outside) tcp 100.1.1.1 25 10.0.0.2 25 netmask 255.255.255.255

Now, what if we have two web servers that both listen to port 80? We can configure the firewall to redirect a different public mapped port (e.g. 8080 for example) to our second web server.

We can use also the Port Redirection feature to translate a well-known port to a lesser-known port or vice-versa. This will help to increase security. For example you can tell your web users to connect to a lesser-known port 5265 and then translate them to the correct port 80 on the local network.

IDENTITY NAT (NAT 0 COMMAND)

It is worth mentioning another type of NAT mechanism called Identity NAT (or **nat 0**). If you enabled **nat-control** on your firewall, it is mandatory that all packets traversing the security appliance must match a translation rule (either **nat/global** or **static nat** rules). If we want to have some hosts (or whole networks) to pass through the firewall without translation, then the **nat 0** command must be used. This creates a transparent mapping. If Identity NAT is used on an interface, IP addresses on this interface translate to themselves on all lower security interfaces.



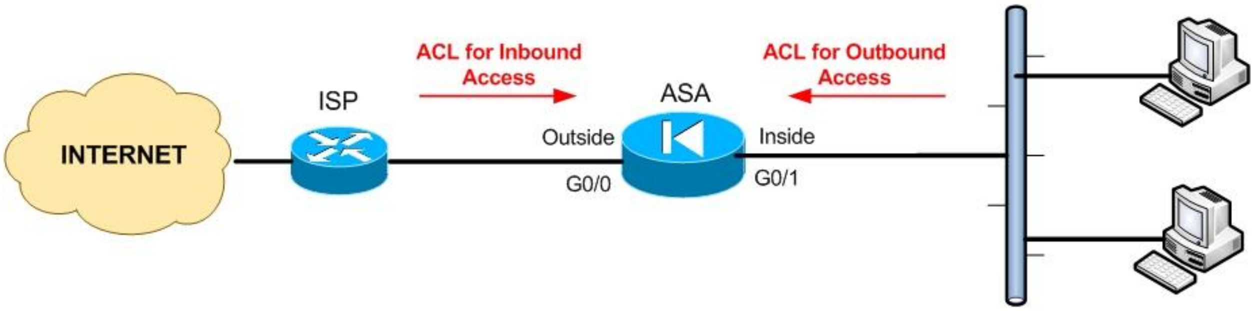
Assume that our DMZ network is assigned a public IP address range (100.1.1.0/24). This means that the servers located on the DMZ have public IP addresses configured on their Network Interface cards. Therefore, we don’t need to translate the DMZ real IP addresses into mapped global addresses.

```
ciscoasa(config)# nat (DMZ) 0 100.1.1.0 255.255.255.0
```

You still need to add a **static** command on the configuration above together with an ACL in order to allow users on the outside to connect with the DMZ servers.

CHAPTER 3:
USING ACCESS CONTROL LISTS (ACL)

In Chapter 2 we described the translation security mechanism, which is one of the two major elements that an administrator needs to configure in order to enable communication through the firewall. The second major element needed to enable traffic flow communication is the Access Control mechanism, also called **Access Control List**.



The Access Control List, as the name implies, is a List of statements (called Access Control Entries) that permit or deny traffic from a source to a destination. After an ACL is configured, it is applied to an interface with an **access-group** command. If no ACL is applied to an interface, outbound access traffic (from inside to outside) is permitted by default, and inbound access traffic (from outside to inside) is denied by default. The ACL can be applied (using the **access-group** command) both to the "**in**" and "**out**" direction of the traffic with respect to the interface. The "**in**" direction of ACL controls traffic entering an interface, and the "**out**" direction of ACL controls traffic exiting an interface. In the diagram above, both ACLs shown (for Inbound and for Outbound Access) are applied to the "**in**" direction of Outside and Inside interfaces respectively.

The following are guidelines for designing and implementing ACLs:

- For **Outbound Traffic** (Higher to Lower Security Levels), the source address argument of an ACL entry is the actual real address of the host or network.
- For **Inbound Traffic** (Lower to Higher Security Levels), the destination address argument of an ACL entry is the translated global IP address.
- ACLs are always checked <u>before</u> translation is performed on the security appliance.

- ACLs, in addition to restricting traffic flow through the firewall, they can be used also as a traffic selection mechanism for applying several other actions to the selected traffic, such as encryption, translation, policing, Quality of Service etc.

The command format of an Access Control List is the following:

ciscoasa(config)# access-list "***access_list_name"*** **[line *line_number*] [extended]**
{*deny | permit*} ***protocol "source_address" "mask" [operator source_port] "dest_address" "mask" [operator dest_port]***

The command format of an Access-Group command used to apply an ACL is the following:

ciscoasa(config)# access-group "***access_list_name"*** **[in|out] interface "*interface_name*"**

Let's see all the elements of the ACL command below:

- ***access_list_name*** : Give a descriptive name of the specific ACL. The same name is used in the access-group command.
- **line *line_number*** : Each ACL entry has its own line number.
- **extended**: Use this when you specify both source and destination addresses in the ACL.
- ***deny/permit*** : Specify whether the specific traffic is permitted or denied.
- ***protocol:*** Specify here the traffic protocol (IP, TCP, UDP etc).
- ***source_address mask:*** Specify the source IP address/network that the traffic originates. If it's a single IP address, you can use the keyword "**host**" without a mask. You can also use the keyword "**any**" to specify any address.
- ***[operator source_port]:*** Specify the source port number of the originating traffic. The "operator" keyword can be "**lt**" (less than), "**gt**" (greater than), "**eq**" (equal), "**Neq**" (Not equal to), "**range**" (range of ports). If no ***source_port*** is specified, the firewall matches all ports.
- ***dest_address mask:*** This is the destination IP address/network that the source address requires access to. You can use also the "**host**" or "**any**" keywords.
- ***[operator dest_port]:*** Specify the destination port number that the source traffic requires access to. The "operator" keyword can be "**lt**" (less than), "**gt**" (greater than), "**eq**" (equal), "**Neq**" (Not equal to), "**range**" (range of ports). If no ***dest-port*** is specified, the firewall matches all ports.

The ACL examples below will give us a better picture of the command format:

```
ciscoasa(config)# access-list DMZ_IN extended permit ip any any
ciscoasa(config)# access-group DMZ_IN in interface DMZ
```

The above will allow ALL traffic from DMZ network to go through the firewall.

```
ciscoasa(config)# access-list INSIDE_IN extended deny tcp 192.168.1.0 255.255.255.0 200.1.1.0 255.255.255.0
ciscoasa(config)# access-list INSIDE_IN extended deny tcp 192.168.1.0 255.255.255.0 host 210.1.1.1 eq 80
ciscoasa(config)# access-list INSIDE_IN extended permit ip any any
ciscoasa(config)# access-group INSIDE_IN in interface inside
```

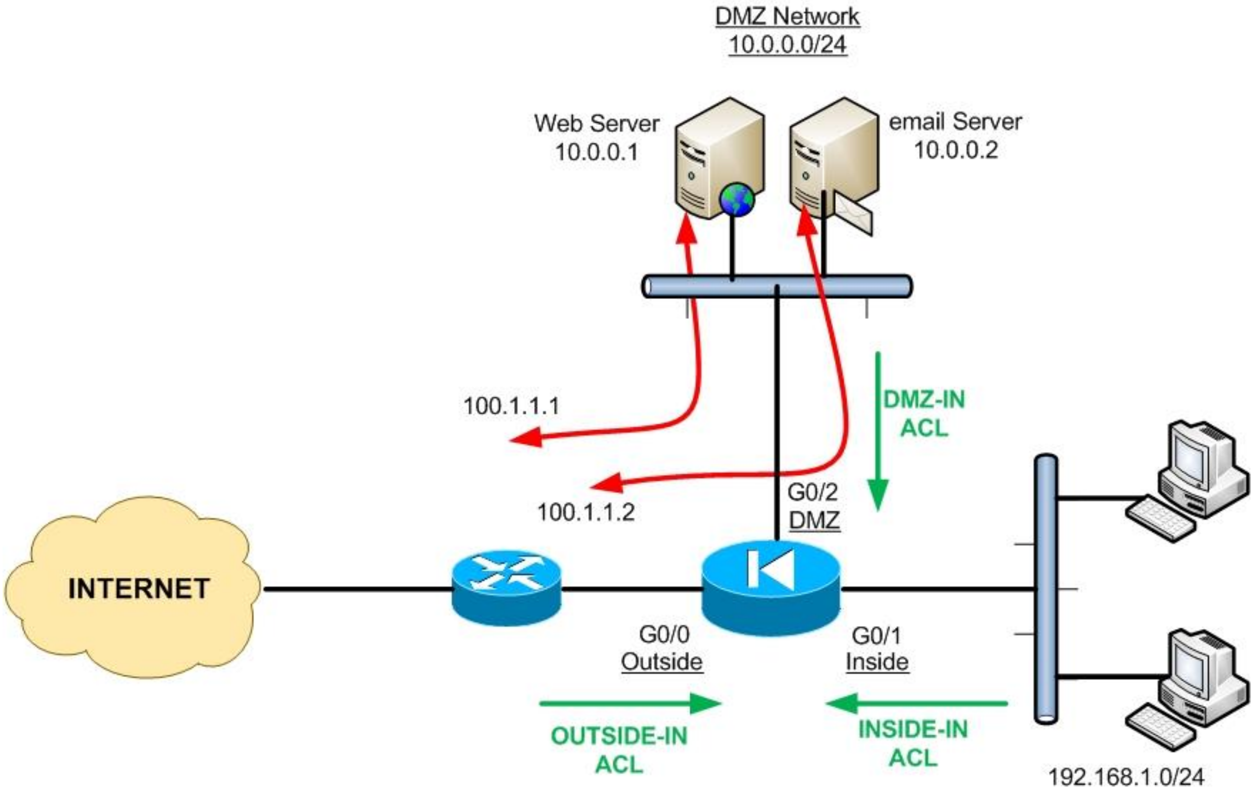
The above example will deny ALL TCP traffic from our internal network 192.168.1.0/24 towards the external network 200.1.1.0/24. Also, it will deny HTTP traffic (port 80) from our internal network to the external host 210.1.1.1. All other traffic will be permitted from inside.

```
ciscoasa(config)# access-list OUTSIDE_IN extended permit tcp any host 100.1.1.1 eq 80
ciscoasa(config)# access-group OUTSIDE_IN in interface outside
```

The ACL above will allow ANY host on the Internet to access our Web Server host (100.1.1.1). Notice that address 100.1.1.1 is the public global translated address of our Web server.

CONTROLLING INBOUND AND OUTBOUND TRAFFIC WITH ACLS

A picture is a thousand words. Refer to the picture diagram below for the example scenarios that will follow. These examples will show you how to control Inbound and Outbound Traffic flow:



Scenario 1: Allow Inbound Access to DMZ Servers

For the Web and email Servers above, we have created static NAT mappings in order to translate their real private addresses into public addresses that are accessible from the Internet. In addition to the static NAT statements, we have to use also ACLs to allow the appropriate Inbound traffic towards our servers.

```
ciscoasa(config)# static (DMZ , outside) 100.1.1.1  10.0.0.1 netmask 255.255.255.255
ciscoasa(config)# static (DMZ , outside) 100.1.1.2  10.0.0.2 netmask 255.255.255.255
ciscoasa(config)# access-list OUTSIDE-IN extended  permit tcp any host 100.1.1.1 eq 80
ciscoasa(config)# access-list OUTSIDE-IN extended  permit tcp any host 100.1.1.2 eq 25
ciscoasa(config)# access-group OUTSIDE-IN in interface outside
ciscoasa(config)# access-list DMZ-IN extended  deny ip any any log
ciscoasa(config)# access-group DMZ-IN in interface DMZ
```

As you can see from the ACL statements, we allow "any" traffic (i.e all internet traffic) to access the public IP addresses of our Web and email servers on the appropriate ports only (80 and 25). Also, all traffic originating from the DMZ servers is denied and logged using the DMZ-IN ACL. This is a good security practice to follow because if a DMZ server is compromised from outside, the attacker will not be able to access anything else from the DMZ zone.

Scenario 2: Apply Identity NAT (nat 0) to Inside Network when accessing DMZ

As we have mentioned earlier, ACLs, in addition to restricting traffic flow, they can be used also to identify traffic for applying other actions to it. For our diagram above, assume that we want to apply Identity NAT to our Inside network when this communicates with the DMZ. In other words, when hosts in network 192.168.1.0/24 initiate communication to network 10.0.0.0/24, then we don't want to translate them. To disable NAT translation from a specific high security interface to a lower security interface, we can use the **nat 0** command. An ACL can be used together with the **nat 0** command to identify which traffic flow will <u>not</u> be translated.

```
ciscoasa(config)# access-list NO-NAT extended  permit ip 192.168.1.0 255.255.255.0 10.0.0.0
255.255.255.0 ←Match Traffic from Inside to DMZ
 ciscoasa(config)# nat (inside) 0 access-list NO-NAT ←Do not translate traffic matched by this
                                                                                          ACL
ciscoasa(config)# nat (inside) 1 192.168.1.0 255.255.255.0
ciscoasa(config)# global (outside) 1 interface ←Use PAT when going from Inside to Outside
```

Scenario 3: Apply Outbound Restrictions from Inside to DMZ

Now, assume that users on the Inside network (192.168.1.0/24) are only allowed to access the email Server at port 25 on the DMZ (to retrieve email) but should not have any access to the rest of the DMZ network. All access however towards the Internet should be allowed.

```
ciscoasa(config)# access-list INSIDE-IN extended  permit tcp 192.168.1.0 255.255.255.0 host
10.0.0.2 eq 25
ciscoasa(config)# access-list INSIDE-IN extended  deny ip 192.168.1.0 255.255.255.0 10.0.0.0
255.255.255.0
ciscoasa(config)# access-list INSIDE-IN extended  permit ip 192.168.1.0 255.255.255.0 any
ciscoasa(config)# access-group INSIDE-IN in interface inside

ciscoasa(config)# access-list NO-NAT extended  permit ip 192.168.1.0 255.255.255.0 10.0.0.0
255.255.255.0
ciscoasa(config)# nat (inside) 0 access-list NO-NAT
ciscoasa(config)# nat (inside) 1 192.168.1.0 255.255.255.0
ciscoasa(config)# global (outside) 1 interface
```

CONFIGURING OBJECT GROUPS FOR ACLS

Imagine that you are responsible for a huge network with hundreds of hosts protected by a Cisco Firewall. Imagine also that your organization's security policy dictates that there should be strict access control for all hosts in your network. Creating and maintaining Access Control Lists in such an environment could be a daunting task.

Fortunately, Cisco introduced the **object-group** command which allows the firewall administrator to group together objects such as hosts, networks, ports etc. These object groups can then be used with the access-list command to reference all objects within the group. This helps to reduce multiple lines in the access list and makes ACL administration much easier. Also, any changes in hosts, ports etc are done inside the **object-group** and are automatically applied in the access-list command.

There are four types of object groups:

- **Network**: Used to group together hosts or subnets.
- **Service**: Used to group TCP or UDP port numbers.
- **Protocol**: Used to group protocols.
- **ICMP-type**: Used to group ICMP message types.

We will describe the first two types (Network and Service object groups) since they are the most important.

Network Object Groups

The command format of the Network Object Group is the following:

```
ciscoasa(config)# object-group network "group_name" ←First Define a name of the object
group. This will put you in a subcommand mode (config-network)
ciscoasa(config-network)# network-object host "ip_addr" ←Define a single Host
ciscoasa(config-network)# network-object "net_addr netmask" ←Define a whole subnet
ciscoasa(config-network)# exit
ciscoasa(config)#
```

Example:

- Create the Network Object Group:

```
ciscoasa(config)# object-group network WEB_SRV
ciscoasa(config-network)# network-object host 10.0.0.1
ciscoasa(config-network)# network-object host 10.0.0.2

ciscoasa(config)# object-group network DMZ_SUBNET
ciscoasa(config-network)# network-object 10.0.0.0 255.255.255.0
```

- Using the object group with an ACL:

```
ciscoasa(config)# access-list OUTSIDE-IN extended  permit tcp any object-group WEB_SRV eq 80
```

In the example above, we created a network object group (WEB_SRV) for our Web Servers (10.0.0.1 and 10.0.0.2). With a single ACL statement, we allowed TCP access from Outside towards this specific object-group for port 80. Notice that the network object-group in the access-list command is used in place of the destination address. It could be used also in place of the source address accordingly.

SERVICE OBJECT GROUPS

The command format of the Service Object Group is the following:

```
ciscoasa(config)# object-group service "group_name" {tcp | udp | tcp-udp} ←First Define a name of the obj. group and specify what kind of service ports will follow (tcp, udp or both)
ciscoasa(config-service)# port-object {eq | range} "port_number" ←Define service ports
ciscoasa(config-service)# exit
ciscoasa(config)#
```

Example:

- Create the Service Object Group:

```
ciscoasa(config)# object-group service DMZ_SERVICES tcp
ciscoasa(config-service)# port-object eq http
ciscoasa(config-service)# port-object eq https
ciscoasa(config-service)# port-object range 21 23

ciscoasa(config)# object-group network DMZ_SUBNET
ciscoasa(config-network)# network-object 10.0.0.0 255.255.255.0
```

- Using the object group with an ACL:

```
ciscoasa(config)# access-list OUTSIDE-IN extended  permit tcp any object-group DMZ_SUBNET object-group DMZ_SERVICES
```

In our example above, assume that we have a DMZ network 10.0.0.0/24 hosting servers with tcp services of http, https, ftp (port 21), ssh (port 22) and telnet (port 23). For this scenario we created a DMZ network object group (DMZ_SUBNET) together with a service object group (DMZ_SERVICES). The DMZ_SUBNET group is used in place of the destination address, and the DMZ_SERVICES group is used in place of the destination port.

CHAPTER 4:
CONFIGURING VLANS AND SUBINTERFACES

In this Chapter we will focus on Interface Layer 2 connectivity of the Cisco ASA firewall. Let me remind you that each interface (physical or logical) of the ASA appliance is used to create a security zone, which is basically a network segment (Layer 3 subnet) hosting PCs, Servers etc. Each security zone is protected by the firewall from the other security zones on the appliance.

In order to build a secure network that follows the principles of "**Layered Security**", it is a good practice to segment your network into different security zones (Layer 3 subnets) which are controlled and protected by the firewall. To create security zones, you can use either Physical or Logical Interfaces on the appliance. However, in order to create Layer 3 subnets, you must have also a different Layer 2 VLAN for each subnet.

Cisco ASA firewalls support multiple 802.1q VLANs on a Physical interface. This means that an administrator can configure multiple Logical interfaces (subinterfaces) on a single physical interface and assign each logical interface to a specific VLAN. For example, a Cisco firewall appliance with 4 physical interfaces is not limited to having only 4 security zones. We can create for example 3 logical subinterfaces on each physical interface, which will give us 12 (4x3) different security zones (12 VLANs and 12 Layer 3 subnets). Depending on the ASA model, up to 250 maximum VLANs can be configured on a single appliance.

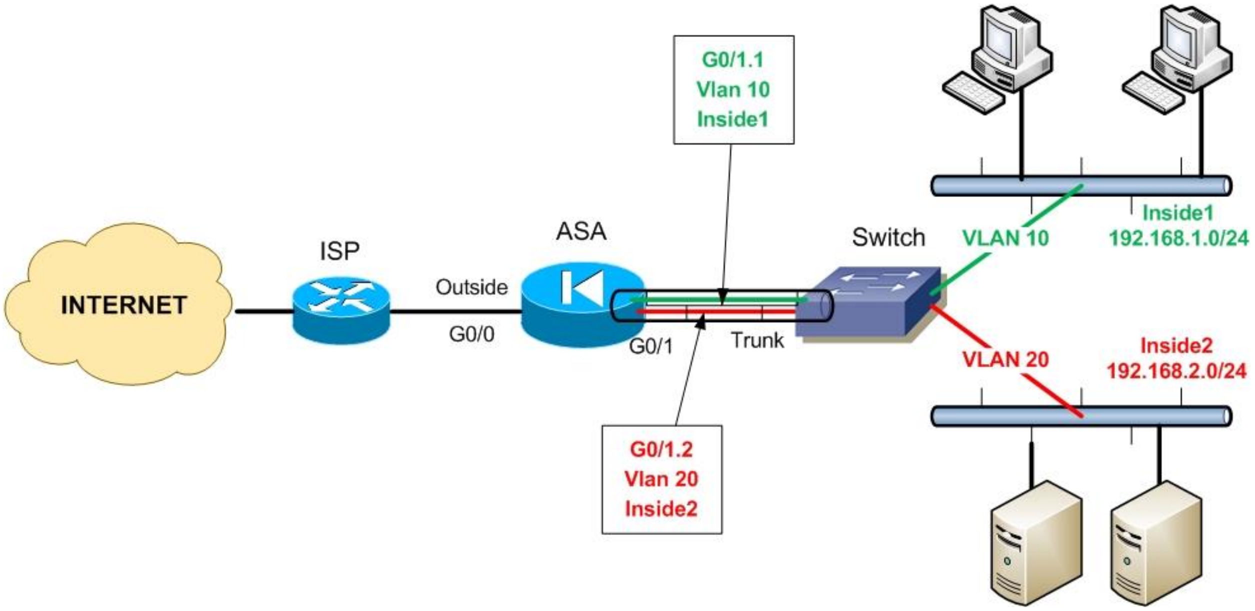
If you configure subinterfaces (VLANs) on a physical interface, then this physical interface must be connected to a Trunk Port on a Layer 2 switch. In addition, if you enable subinterfaces, you typically do not want the main physical interface to also be passing traffic. You can achieve this by omitting the **nameif** command (**no nameif**) on the physical interface.

To configure logical subinterfaces, use the *subinterface* argument of the **interface** command in global configuration mode. This will put you in subinterface configuration mode, where you have to assign a VLAN ID using **vlan** *id* command. As we mentioned in "**Basic Configuration Steps**" Section of Chapter 1, we also have to configure a name for the subinterface (**nameif**), a security level, and an IP address.

The command format for configuring VLAN logical subinterfaces is shown below:

ciscoasa(config)# interface "*physical_interface.subinterface*" ←Use the *subinterface* argument
ciscoasa(config-subif)# ←This is the subinterface configuration mode
ciscoasa(config-subif)# vlan "*id*" ←Assign a VLAN to the subinterface
ciscoasa(config-subif)# nameif "*subif_name*" ←Assign a name to the subinterface
ciscoasa(config-subif)# security-level "*0-100*" ←Assign a security level to the subinterface
ciscoasa(config-subif)# ip address "*IP*" "*netmask*" ←Assign IP address

Let's see an example scenario below with a network diagram.



In the example above, assume that we wanted to segment our internal network into two security zones (**Inside1** and **Inside2**). Maybe Inside1 zone will host all user PCs, and Inside2 zone will host all internal corporate servers (email server, domain server etc). To build this topology, we need to create two VLANs on the switch (10 and 20), one for each network subnet. Instead of using two Physical Interfaces of the ASA firewall (one for each zone), we used one physical interface with two logical interfaces, as shown below:

- G0/1 = Physical Interface
- G0/1.1 = Logical Interface (subinterface) assigned to VLAN 10
- G0/1.2 = Logical Interface (subinterface) assigned to VLAN 20

The two logical interfaces (G0/1.1 and G0/1.2) behave just like the physical interface, and they are two separate "legs" of the firewall.

See the sample configuration below for details:

```
ciscoasa(config)# interface gigabitethernet 0/1
ciscoasa(config-if)# no nameif ←Disable the physical interface from passing traffic
ciscoasa(config-if)# no security-level
ciscoasa(config-if)# no ip address
ciscoasa(config-if)# exit

ciscoasa(config)# interface gigabitethernet 0/1.1
ciscoasa(config-subif)# vlan 10
ciscoasa(config-subif)# nameif  inside1
ciscoasa(config-subif)# security-level  80
ciscoasa(config-subif)# ip address 192.168.1.1 255.255.255.0

ciscoasa(config)# interface gigabitethernet 0/1.2
ciscoasa(config-subif)# vlan 20
ciscoasa(config-subif)# nameif  inside2
ciscoasa(config-subif)# security-level  90
ciscoasa(config-subif)# ip address 192.168.2.1 255.255.255.0
```

CHAPTER 5:
IPSEC VPNS

This Chapter discusses Virtual Private Networks using the IPSec protocol standard. Cisco ASA appliances, in addition to their core firewall functionality, can be used also to securely connect together distant LAN networks (Site-to-Site VPN) or allow remote users/teleworkers to securely communicate with their corporate network (Remote-Access VPN). In this Chapter we will focus on these two types of VPNs.

Before proceeding with the technical details of configuring IPSEc VPNs, it will be very useful to briefly describe the theory behind IPSEc in order to have a knowledge base for understanding the discussion in later sections of this Chapter.

WHAT IS IPSEC

IP Security (**IPSEc**) is an open IETF standard that enables encrypted communication. It is a suit of protocols that provide data confidentiality, integrity, and authentication. A Virtual Private Network (**VPN**) is a secure private tunnel over an insecure path (e.g over the Internet). IPSEc therefore is ideal to build VPNs over the Internet or any other non-secure networks.

IPSEc works at the network layer, encrypting and authenticating IP packets between a firewall security appliance and other participating IPSEc devices (peers), such as Cisco routers, other Cisco firewalls, VPN software clients etc.

The following IPSEc protocols and standards will be used later in our discussion, so it’s a good idea to briefly explain their functionality and usage:

- **ESP (Encapsulation Security Payload):** This is the first of the two main protocols that make up the IPSEc standard. It provides data integrity, authentication, and confidentiality services. ESP is used to encrypt the data payload of the IP packets.
- **AH (Authentication Header):** This is the second of the two main protocols of IPSEc. It provides data integrity, authentication, and replay-detection. It does not provide encryption

services, but rather it acts as a "digital signature" for the packets to ensure that tampering of data has not occurred.

- **Internet Key Exchange (IKE):** This is the mechanism used by the security appliance for securely exchanging encryption keys, authenticating IPSEc peers and negotiating IPSEc Security parameters. On the ASA firewall, this is synonymous with **ISAKMP** as we will see in the IPSEc configuration.
- **DES, 3DES, AES:** All these are encryption algorithms supported by the Cisco ASA Firewall. DES is the weakest one (uses 56-bit encryption key), and AES is the strongest one (uses 128, 192, or 256 bit encryption keys). 3DES is a middle choice using 168-bit encryption key.
- **Diffie-Hellman Group (DH):** This is a public-key cryptography protocol used by IKE to establish session keys.
- **MD5, SHA-1:** These are both Hash Algorithms used to authenticate packet data. SHA is stronger than MD5.
- **Security Association (SA):** An SA is a connection between two IPSEc peers. Each IPSEc peer maintains an SA database in its memory containing SA parameters. SAs are uniquely identified by the IPSEc peer address, security protocol, and security parameter index (SPI).

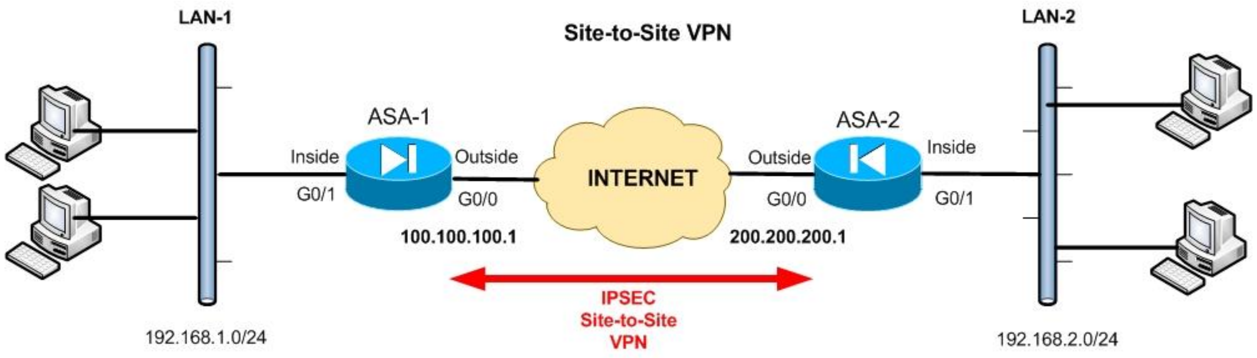
HOW IPSEC WORKS

There are five main steps followed by the IPSEc devices:

1. **Interesting Traffic**: The IPSEc devices recognize the traffic to protect.
2. **Phase 1 (ISAKMP)**: The IPSEc devices negotiate an IKE security policy and establish a secure channel for communication.
3. **Phase 2 (IPSEc)**: The IPSEc devices negotiate an IPSEc security policy to protect data.
4. **Data Transfer**: Data is transferred securely between the IPSEc peers based on the IPSEc parameters and keys negotiated during the previous phases.
5. **IPSEc Tunnel Terminated**: IPSEc SAs terminate when timing out or a certain data volume is reached.

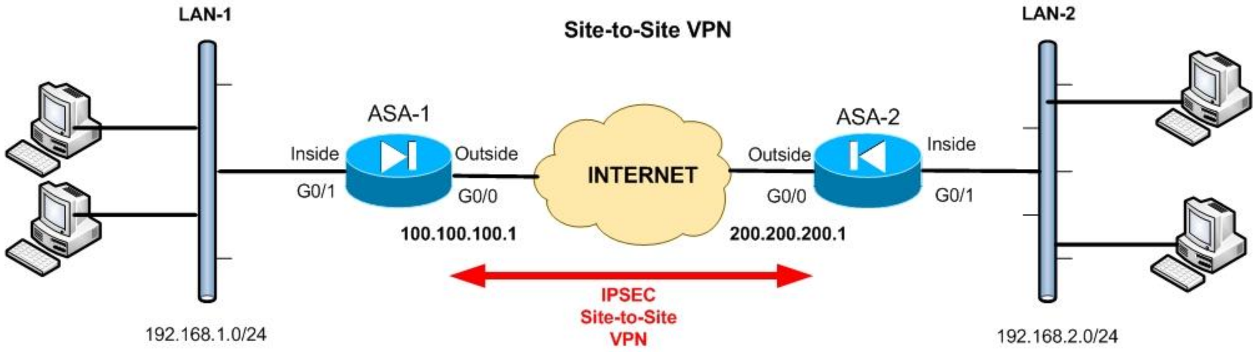
The steps above will become clear when we see actual configuration examples. Let's start with the first IPSEc VPN type that we will describe in this Chapter. Site-to-Site VPN.

SITE-TO-SITE IPSEC VPN



Site-to-Site IPSEc VPN is sometimes called LAN-to-LAN VPN. As the name implies, this VPN type connects together two distant LAN networks over the Internet. Usually, Local Area Networks use private addressing as shown on our diagram above. Without VPN connectivity, the two LAN networks above (LAN-1 and LAN-2) wouldn't be able to communicate. By configuring a Site-to-Site IPSEc VPN between the two ASA firewalls, we can establish a secure tunnel over the Internet, and pass our private LAN traffic inside this tunnel. The result is that hosts in network 192.168.1.0/24 can now directly access hosts in 192.168.2.0/24 network (and vice-versa) as if they are located in the same LAN. The IPSEc tunnel is established between the Public IP addresses of the firewalls (100.100.100.1 and 200.200.200.1).

CONFIGURING SITE-TO-SITE IPSEC VPN



As we described above in "How IPSEc Works", there are five steps in the operation of IPSEc. Next we will describe the configuration commands needed for each step in order to set up the VPN. All configuration examples below refer to the network diagram for site-to-site VPN.

- **STEP 1: Configure Interesting Traffic**

We need first to define the Interesting Traffic, that is, traffic that will be encrypted. Using Access-Lists (**Crypto ACL**) we can identify which traffic flow must be encrypted. In our example diagram above, we want all traffic flow between private networks 192.168.1.0/24 and 192.168.2.0/24 to be encrypted.

ASA-1(config)# access-list LAN1-to-LAN2 extended permit ip 192.168.1.0 255.255.255.0 192.168.2.0 255.255.255.0

ASA-2(config)# access-list LAN2-to-LAN1 extended permit ip 192.168.2.0 255.255.255.0 192.168.1.0 255.255.255.0

Notice that we have to configure the exact mirror access-list for each ASA firewall participating in the IPSEc VPN. The Crypto ACL needs to identify only outbound traffic. The **permit** statement in the ACL means that the specific traffic must be encrypted.

One important issue to consider is the case of using NAT on the firewall for normal Internet access. Because IPSEc does not work with NAT, we need to exclude the traffic to be encrypted from the NAT operation. This means in our example that the Interesting Traffic in the Crypto ACL must not be translated (we can use the **nat 0** command for this).

```
ASA-1(config)# access-list NONAT extended permit ip 192.168.1.0 255.255.255.0 192.168.2.0 255.255.255.0
ASA-1(config)# nat (inside) 0 access-list NONAT
```

```
ASA-2(config)# access-list NONAT extended permit ip 192.168.2.0 255.255.255.0 192.168.1.0 255.255.255.0
ASA-2(config)# nat (inside) 0 access-list NONAT
```

- **STEP 2: Configure Phase 1 (ISAKMP)**

Phase 1 of the IPSEc operation is used to establish a secure communication channel for further data transmission. In Phase 1, VPN peers exchange shared secret keys, authenticate each other, negotiate IKE security policies etc. In this Phase we configure an **isakmp policy** which MUST match the policy configured on the other peer(s). This **isakmp policy** tells the other peer(s) what security parameters must be used in the VPN (e.g encryption protocol, hash algorithm, authentication method, Diffie Hellman Group (DH), lifetime threshold for the tunnel etc).

The command format of the isakmp policy is the following:

```
ASA(config)# isakmp policy "priority number" ←Lower number means higher priority
ASA(config-isakmp-policy)# encryption {aes | 3des | des}
ASA(config-isakmp-policy)# hash {sha | md5}
ASA(config-isakmp-policy)# authentication {pre-share | rsa-sig}
ASA(config-isakmp-policy)# group {1 | 2 | 5 | 7} ←DH Group
ASA(config-isakmp-policy)# lifetime "seconds" ←Up to 86400 seconds
ASA(config)# isakmp enable "interface-name" ←Attach the policy on an interface
ASA(config)# isakmp identity address ←Identify the ASA with its address and not FQDN
```

Several isakmp policies can be configured to match different requirements from different IPSEc peers. The priority number uniquely identifies each policy. The lower the priority number, the higher the priority will be given to the specific policy.

The following example parameters can be used to create a strong isakmp policy:

- Encryption **aes**
- Hash **sha**
- Authentication **pre-share**
- Group **2 or 5**
- Lifetime **3600** (the Security Association – SA will expire and renegotiate every 1 hour)

The next thing we need to specify is the pre-shared key and the type of the VPN (Lan-to-Lan, Remote Access or WebVPN). These are configured by the **tunnel-group** command.

```
ASA(config)# tunnel-group "peer IP address" type {ipsec-l2l | ipsec-ra | webvpn}
ASA(config)# tunnel-group "peer IP address" ipsec-attributes
ASA(config-tunnel-ipsec)# pre-shared-key "key"
```

Note: *The tunnel-group types "**ipsec-ra**" and "**webvpn**" were deprecated from ASA version 8.0(2). These two are replaced by the new "**remote-access**" type.*

Let's see the complete example configuration for both firewalls for Phase 1 setup:

```
ASA-1(config)# isakmp policy 10
ASA-1(config-isakmp-policy)# encryption aes
ASA-1(config-isakmp-policy)# hash sha
ASA-1(config-isakmp-policy)# authentication pre-share
ASA-1(config-isakmp-policy)# group 2
ASA-1(config-isakmp-policy)# lifetime 3600
ASA-1(config)# isakmp enable outside
ASA-1(config)# isakmp identity address
ASA-1(config)# tunnel-group 200.200.200.1 type ipsec-l2l
ASA-1(config)# tunnel-group 200.200.200.1 ipsec-attributes
ASA-1(config-tunnel-ipsec)# pre-shared-key somestrongkey
```

```
ASA-2(config)# isakmp policy 10
ASA-2(config-isakmp-policy)# encryption aes
ASA-2(config-isakmp-policy)# hash sha
ASA-2(config-isakmp-policy)# authentication pre-share
ASA-2(config-isakmp-policy)# group 2
ASA-2(config-isakmp-policy)# lifetime 3600
ASA-2(config)# isakmp enable outside
ASA-2(config)# isakmp identity address
ASA-2(config)# tunnel-group 100.100.100.1 type ipsec-l2l
ASA-2(config)# tunnel-group 100.100.100.1 ipsec-attributes
ASA-2(config-tunnel-ipsec)# pre-shared-key somestrongkey
```

- **STEP 3: Configure Phase 2 (IPSEc)**

After a secured tunnel is established in Phase 1, the next step in setting up the VPN is to negotiate the IPSEc security parameters that will be used to protect the data and messages within the tunnel. This is achieved in Phase 2 of the IPSEc. In this Phase the following functions are performed:

- Negotiation of IPSEc security parameters and IPSEc **transform sets**.
- Establishment of IPSEc SAs.
- Renegotiation of IPSEc SAs periodically to ensure security.

The ultimate goal of IKE Phase 2 is to establish a secure IPSEc session between peers. Before that can happen, each pair of endpoints negotiates the level of security required (encryption and authentication algorithms for the session). Rather than negotiate each encryption and authentication protocol individually, the protocols are grouped into sets, called **transform sets**. IPSEc transform sets are exchanged between peers and they must match between peers in order for the session to be established.

The command format of configuring a transform set is the following:

ASA(config)# crypto ipsec transform-set "*name*" "*transform1*" "*transform2*"

The following transforms (protocols/algorithms) can be used in place of *transform1* and *transform2*:

Transform	Description
esp-des	ESP transform using DES cipher (56 bits)
esp-3des	ESP transform using 3DES cipher (168 bits)
esp-aes	ESP transform using AES-128 cipher
esp-aes-192	ESP transform using AES-192 cipher
esp-aes-256	ESP transform using AES-256 cipher
esp-md5-hmac	ESP transform using HMAC-MD5 authentication
esp-sha-hmac	ESP transform using HMAC-SHA authentication
esp-none	ESP with no authentication
esp-null	ESP with null encryption

The following guidelines might be useful when choosing transform protocols:

- For providing data confidentiality (encryption), use an ESP encryption transform such as the first 5 in the list above.
- Also consider using an ESP authentication transform by choosing MD5-HMAC or SHA-HMAC algorithms.
- SHA is stronger than MD5 but it is slower.

Consider the following example combinations of transform sets:

- ESP-DES for high performance encryption but with no authentication.
- ESP-3DES and ESP-MD5-HMAC for strong encryption and authentication.
- ESP-AES-192 and ESP-SHA-HMAC for stronger encryption and authentication.

After configuring a transform set on both IPSEc peers, we need to configure a **crypto map** which contains all Phase 2 IPSEc parameters. This crypto map is then attached to the firewall interface (usually "outside") on which the IPSEc will be established.

The command format of a crypto map is:

```
ASA(config)# crypto map "name" "seq-num" match address "Crypto-ACL" ←Assign the Crypto ACL which specifies the Interesting Traffic to be encrypted.
ASA(config)# crypto map "name" "seq-num" set peer "Peer_IP_address" ←Specify the remote peer IP address
ASA(config)# crypto map "name" "seq-num" set transform-set "Transform_set_name" ←This is the transform set name configured above
ASA(config)# crypto map "name" "seq-num" set security-association lifetime seconds {Seconds} ←Specify how often the SA will expire and get renegotiated.

ASA(config)# crypto map "name" interface "interface-name" ←Attach the map to an interface
```

The *seq-num* parameter in the crypto map is used to specify multiple map entries (with the same name) for cases where we have more than one IPSEc peer for the firewall (e.g three ASA firewalls in a hub-and-spoke configuration).

Let’s see the complete example configuration for both firewalls for Phase 2 setup:

```
ASA-1(config)# crypto ipsec transform-set ASA1TS esp-aes-192 esp-sha-hmac
ASA-1(config)# crypto map ASA1VPN 10 match address LAN1-to-LAN2
ASA-1(config)# crypto map ASA1VPN 10 set peer 200.200.200.1
ASA-1(config)# crypto map ASA1VPN 10 set transform-set ASA1TS
ASA-1(config)# crypto map ASA1VPN 10 set security-association lifetime seconds 36000
ASA-1(config)# crypto map ASA1VPN interface outside
```

```
ASA-2(config)# crypto ipsec transform-set ASA2TS esp-aes-192 esp-sha-hmac
ASA-2(config)# crypto map ASA2VPN 10 match address LAN2-to-LAN1
ASA-2(config)# crypto map ASA2VPN 10 set peer 100.100.100.1
ASA-2(config)# crypto map ASA2VPN 10 set transform-set ASA2TS
ASA-2(config)# crypto map ASA2VPN 10 set security-association lifetime seconds 36000
ASA-2(config)# crypto map ASA2VPN interface outside
```

- **STEP 4: Verify Encrypted Data Transfer**

With the three steps above we concluded the configuration of a site-to-site IPSEc VPN. An essential step though is to verify that everything is working fine and that our data is actually getting encrypted by the firewalls. There are two important commands that will help you verify if the tunnel is established and if data is bi-directionally encrypted between the IPSEc peers.

<u>Verify that tunnel is established</u>

The **show crypto isakmp sa** command verifies that the Security Association (SA) is established which means that the tunnel is up and running. Let’s see an example output of this command below:

ASA-1# **show crypto isakmp sa**

```
  Active SA: 1
   Rekey SA: 0 (A tunnel will report 1 Active and 1 Rekey SA during rekey)
Total IKE SA: 1

1   IKE Peer: 200.200.200.1
   Type    : L2L           Role    : responder
   Rekey   : no            State   : MM_ACTIVE
```

The important point to observer here is the **State : MM_ACTIVE**. This verifies that the IPSEc tunnel is established successfully.

Verify that data is bi-directionally encrypted

The **show crypto ipsec sa** command verifies that data is being encrypted and decrypted successfully by the firewall appliance, as shown below:

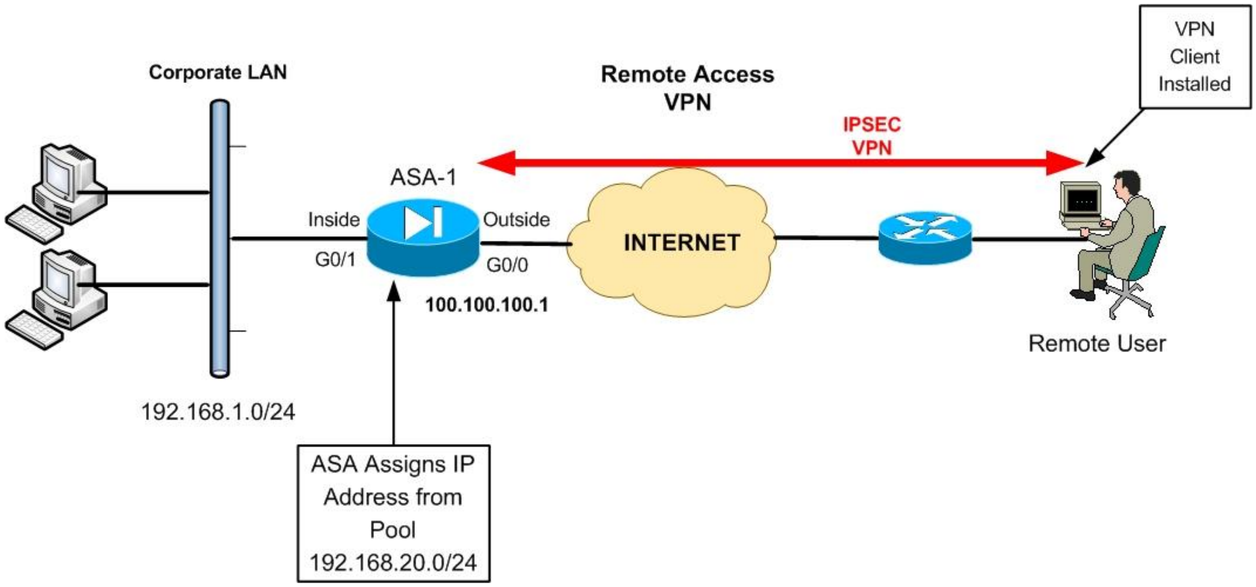
ASA-1# **show crypto ipsec sa**

```
interface: outside
Crypto map tag: ASA1VPN, seq num: 10, local addr: 100.100.100.1

   access-list LAN1-to-LAN2 permit ip 192.168.1.0 255.255.255.0 192.168.2.0 255.255.255.0
   local ident (addr/mask/prot/port): (192.168.1.0/255.255.255.0/0/0)
   remote ident (addr/mask/prot/port): (192.168.2.0/255.255.255.0/0/0)
   current_peer: 200.200.200.1

   #pkts encaps: 2050, #pkts encrypt: 2050, #pkts digest: 2050
   #pkts decaps: 2108, #pkts decrypt: 2108, #pkts verify: 2108
   #pkts compressed: 0, #pkts decompressed: 0
   #pkts not compressed: 2050, #pkts comp failed: 0, #pkts decomp failed: 0
   #pre-frag successes: 0, #pre-frag failures: 0, #fragments created: 0
   #PMTUs sent: 0, #PMTUs rcvd: 0, #decapsulated frgs needing reassembly: 0
   #send errors: 0, #recv errors: 0

   local crypto endpt.: 100.100.100.1, remote crypto endpt.: 200.200.200.1

   path mtu 1500, ipsec overhead 58, media mtu 1500
   current outbound spi: 3CFDDAE7

  inbound esp sas:
   spi: 0x0647B7A6 (105363366)
     transform: esp-aes-192 esp-sha-hmac none
     in use settings ={L2L, Tunnel, }
     slot: 0, conn_id: 2, crypto-map: ASA1VPN
     sa timing: remaining key lifetime (kB/sec): (4274994/26580)
     IV size: 8 bytes
     replay detection support: Y
  outbound esp sas:
   spi: 0x3CFDDAE7 (1023269607)
     transform: esp-aes-192 esp-sha-hmac none
     in use settings ={L2L, Tunnel, }
     slot: 0, conn_id: 2, crypto-map: ASA1VPN
     sa timing: remaining key lifetime (kB/sec): (4274956/26568)
     IV size: 8 bytes
     replay detection support: Y
```

The output field **#pkts encrypt:2050** and **#pkts decrypt:2108** show indeed that we have encryption of data bi-directionally.

REMOTE ACCESS VPN

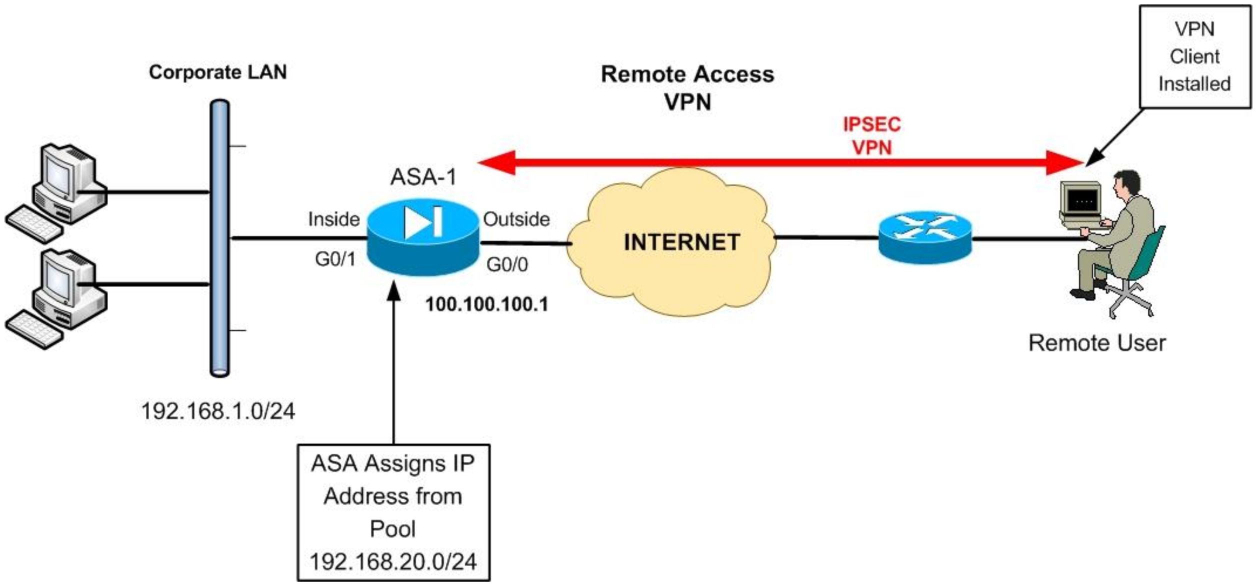
The second type of IPSEc VPN that we will describe in this Chapter is **Remote Access VPN** using a Cisco VPN client on the remote user. This type of VPN allows remote users/teleworkers with Internet access to establish a secure IPSEc VPN tunnel with their central corporate network. The user must have a Cisco VPN client software installed on his/her computer which will enable a secure communication with the ASA firewall in the central office. After the VPN is established between the remote user and the ASA firewall, the user is assigned a private IP address from a predefined pool, and then gets attached on the Corporate LAN network.



Our example network topology above shows a central ASA firewall protecting the Corporate LAN, and a remote user with a software VPN client establishing a secure connection with the ASA. An IP address in the range 192.168.20.0/24 will be assigned to the VPN client, which will be allowed to communicate with the Internal Corporate network 192.168.1.0/24. Once the Remote Access VPN is established, the remote user by default will not be able to access anything else on the Internet, except the Corporate LAN network. This behavior can be altered by configuring the "**split tunneling**" feature on the Firewall, which however is not recommended for security purposes.

Next we will discuss the configuration required both on the ASA Firewall and the Cisco Software client to build a remote access connection.

CONFIGURING REMOTE ACCESS VPN



A lot of configuration statements are the same as the site-to-site VPN, especially for IKE Phase 1 and Phase 2 stages. Also, an IP address pool must be configured on the firewall for dynamically assigning addresses to the remote users. Let's get started with the configuration:

- **STEP 1: Configure an IP address Pool**

The command format is the following:

ASA(config)# ip local pool "*name of pool*" {*first IP address*}-{*last IP address*}

In our example we want to assign addresses to the remote users from the range 192.168.20.0/24:

Example:

ASA-1(config)# ip local pool VPNPOOL 192.168.20.1-192.168.20.254

- **STEP 2: Encrypted Traffic should be excluded from NAT**

Similarly with site-to-site VPN, we need to identify with an ACL the traffic flow from our Internal LAN network (192.168.1.0/24) towards the Remote Users (192.168.20.0/24) in order to be excluded from NAT.

Example:

```
ASA-1(config)# access-list NONAT extended permit ip 192.168.1.0 255.255.255.0
192.168.20.0 255.255.255.0

ASA-1(config)# nat (inside) 0 access-list NONAT
```

- **STEP 3: Configure Group Policy**

The Group Policy allows you to separate different remote access users into groups with different attributes. For example System Administrators can be assigned in a group having 24-hours VPN access, while normal remote user can be in a different group with 9am-5pm VPN access. The Group Policy also provides DNS or WINS server addresses, connection filtering, idle timeout settings etc.

The command format is the following:

```
ASA(config)# group-policy "policy name" internal
ASA(config)# group-policy "policy name" attributes
```

Example:

```
ASA-1(config)# group-policy company-vpn-policy internal
ASA-1(config)# group-policy company-vpn-policy attributes
ASA-1(config-group-policy)# vpn-idle-timeout 30
ASA-1(config-group-policy)# dns-server value 192.168.1.5
ASA-1(config-group-policy)# wins-server value 192.168.1.6
```

Assume that all remote users will use the same group policy, with the name "**company-vpn-policy**" as configured above. This policy assigns DNS and WINS server addresses so that users can resolve internal domain and host names. It sets also the idle timeout to 30 minutes.

- **STEP 4: Configure Usernames for Remote Access authentication**

When remote users connect with the VPN client, they will be presented with a login screen in order to authenticate with the firewall. We need therefore to create username/password combinations for authentication. The command format is:

ASA(config)# username "*name*" password "*password*"

Example:

```
ASA-1(config)# username user password 1234
```

- **STEP 5: Configure Phase 1 (ISAKMP Policy)**

This is similar with site-to-site VPN.

Example:

```
ASA-1(config)# isakmp policy 20
ASA-1(config-isakmp-policy)# encryption 3des
ASA-1(config-isakmp-policy)# hash sha
ASA-1(config-isakmp-policy)# authentication pre-share
ASA-1(config-isakmp-policy)# group 2
ASA-1(config-isakmp-policy)# lifetime 3600
ASA-1(config)# isakmp enable outside
```

- **STEP 6: Configure Phase 2 (IPSEC parameters)**

This Step also has similarities with site-to-site VPNs. We need an IPSEC transform set which will specify the encryption and authentication protocols for the Remote Access VPN. Also, we need to configure a dynamic crypto map which will be assigned to a static crypto map.

Example:

```
! Configure a Transform Set
ASA-1(config)# crypto ipsec transform-set RA-TS esp-3des esp-sha-hmac
! Configure a dynamic crypto map (DYN_MAP)
ASA-1(config)# crypto dynamic-map DYN_MAP 10 set transform-set RA-TS
! Attach the dynamic crypto map (DYN_MAP) to a static map (VPN_MAP)
ASA-1(config)# crypto map VPN_MAP 30 ipsec-isakmp dynamic DYN_MAP
ASA-1(config)# crypto map VPN_MAP interface outside
```

- **STEP 7: Configure a Tunnel Group for Remote Access**

The tunnel group configuration is the heart of remote access VPN. It binds together the Group Policy configured before, the IP pool assignment, the pre-shared key etc.

The command format of the Tunnel Group is the following:

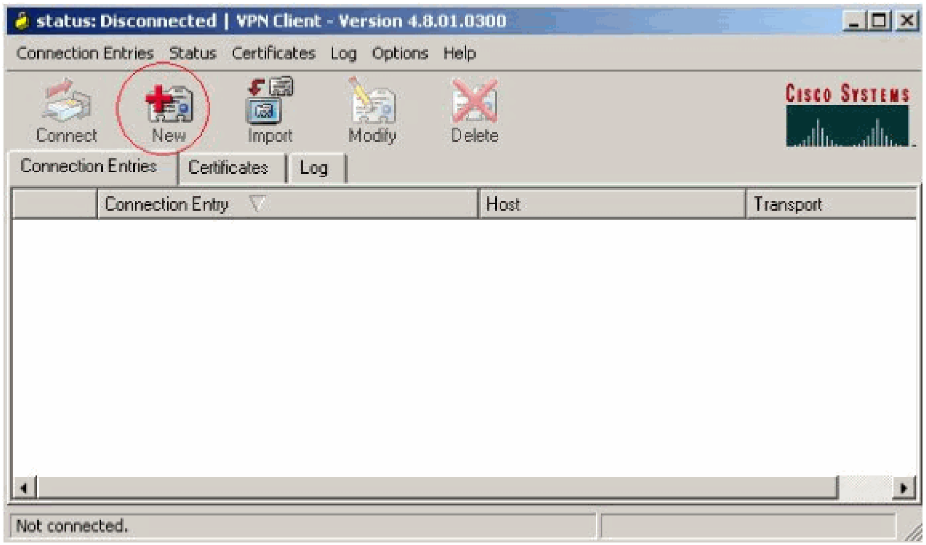
```
ASA(config)# tunnel-group "Group Name" type ipsec-ra
ASA(config)# tunnel-group "Group Name" {general-attributes | ipsec-attributes}
```

The ***Group Name*** is important here because we will have to specify the same exact name when configuring the VPN client software, as we will see later.

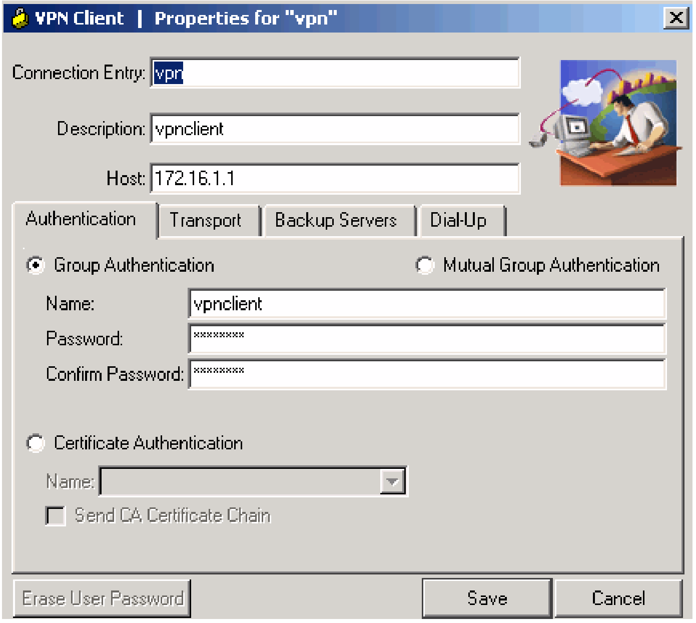
Example:

```
ASA-1(config)# tunnel-group vpnclient type ipsec-ra
ASA-1(config)# tunnel-group vpnclient general-attributes
ASA-1(config-tunnel-general)# address-pool VPNPOOL ←Attach the local IP pool
ASA-1(config-tunnel-general)# default-group-policy company-vpn-policy ←Assign Group
                                                            Policy from Step 3
ASA-1(config)# tunnel-group vpnclient ipsec-attributes
ASA-1(config-tunnel-ipsec)# pre-shared-key groupkey123
```

- **STEP 8: Configure The VPN Client Software**



After installing the VPN client, start the application and select "New" (see above) to create a new connection entry.



VPN Client | Properties for "vpn"

Connection Entry: vpn

Description: vpnclient

Host: 172.16.1.1

● Group Authentication ○ Mutual Group Authentication

Name: vpnclient

Password: ********

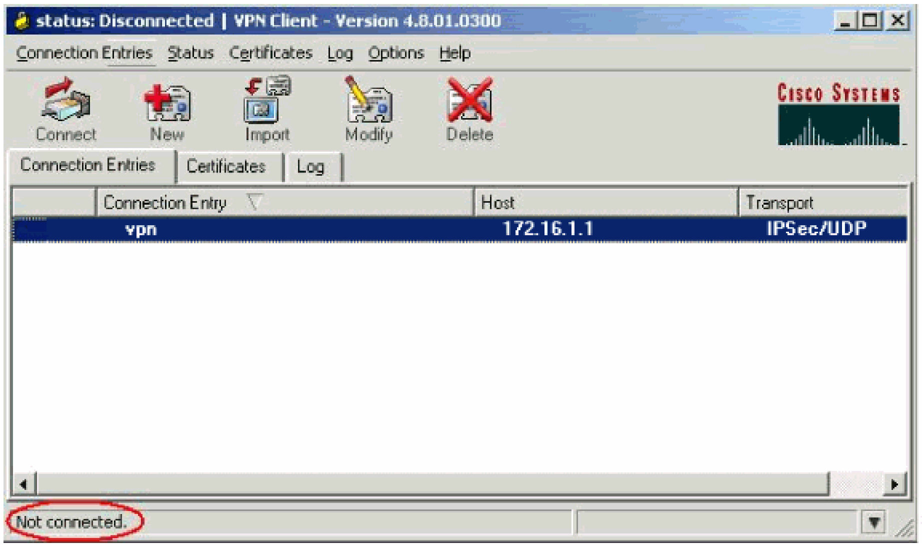
Confirm Password: ********

○ Certificate Authentication

Name:

□ Send CA Certificate Chain

Name the connection entry (e.g "vpn") and provide a description. In the "Host" field, specify the **public IP address** of the outside interface of the central ASA Firewall. The example image above shows 172.16.1.1 but this should be changed accordingly. Also, on the "Group Authentication" Tab, the **Name** and **Password** of the Group must be the same as the **tunnel-group name** and **pre-shared-key** from Step 7 above. In our example configuration, the Group Authentication Name is "**vpnclient**" and the Password (pre-shared-key) is "**groupkey123**". Press "Save" to save the settings.

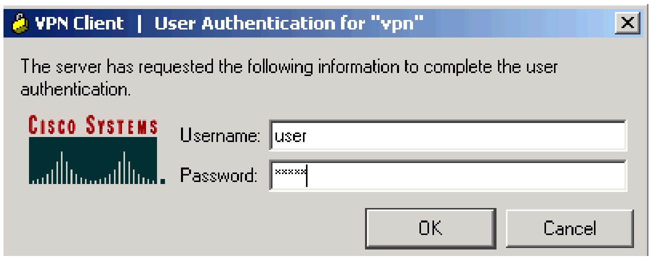


status: Disconnected | VPN Client - Version 4.8.01.0300

Connection Entry	Host	Transport
vpn	172.16.1.1	IPSec/UDP

Not connected.

After saving the configuration settings, return to the Connection Entries Tab as shown above. Press the "Connect" button to initiate the Remote Access VPN connection.



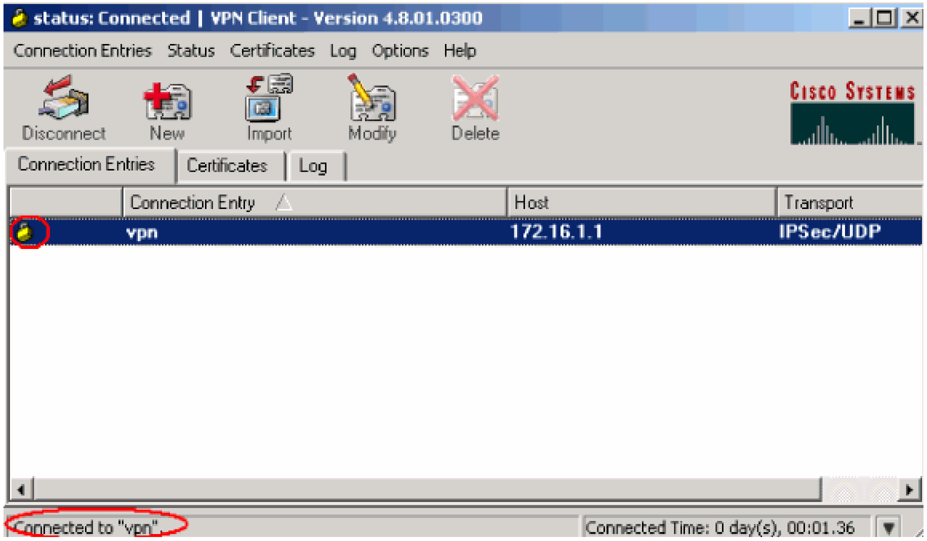
VPN Client | User Authentication for "vpn"

The server has requested the following information to complete the user authentication.

Username: user

Password: *****

After initiating the VPN communication, the remote user will be presented with a login screen in order to authenticate with the firewall. The credentials used in our example configuration (see Step 4 above) are Username: **user** and Password: **1234**



status: Connected | VPN Client - Version 4.8.01.0300

	Connection Entry	Host	Transport
	vpn	172.16.1.1	IPSec/UDP

Connected to "vpn".

Connected Time: 0 day(s), 00:01.36

After successfully authenticating with the firewall, the secure VPN Remote Access tunnel is established. If you use the **ipconfig/all** command on the remote user's computer, you will see an IP address in the range 192.168.20.0/24 assigned to the virtual VPN connection interface. This will enable the remote user to have full network access to the central corporate LAN.

CHAPTER 6:
CONFIGURING FIREWALL FAILOVER

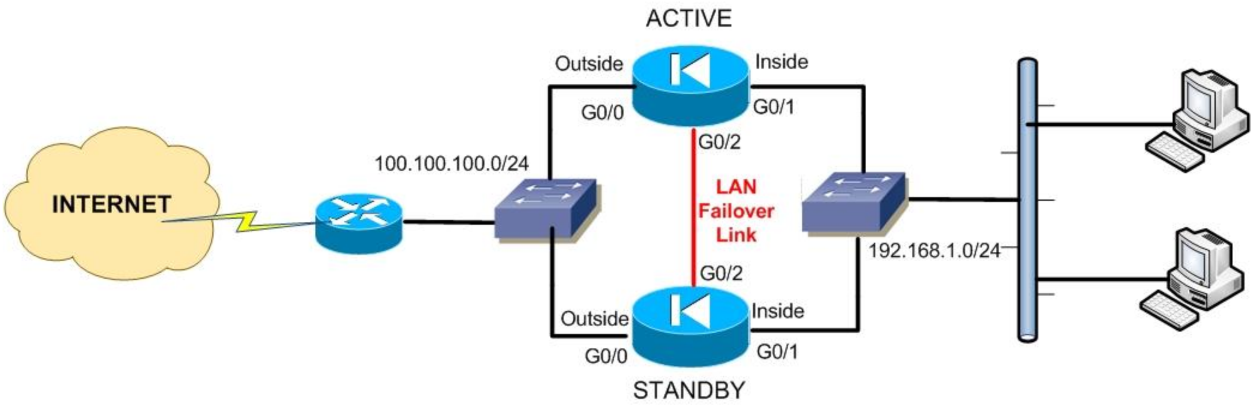
The Cisco ASA Firewall is a critical component of any network infrastructure and usually several essential enterprise services depend on the availability of the Firewall appliance. Firewall redundancy is therefore a must in many network topologies.

In this Chapter we will describe stateful failover in Active/Standby mode which is the most popular configuration in most networks. ASA supports also Active/Active failover mode which however requires special configuration using multiple firewall contexts.

UNDERSTANDING ACTIVE/STANDBY STATEFUL FAILOVER

In an Active/Standby (A/S) mode of operation, one of the firewall units in the failover pair is assigned the active role, handling all traffic and security functions. The other firewall unit in the pair remains in standby mode waiting to automatically take over all the traffic in the event of a failure.

The stateful failover feature passes connection state information from the active to the standby unit. After failover occurs, the same connection information is available at the standby unit, which automatically becomes active without any user traffic disconnection. The stateful connection information that is synchronized between active and standby units include global pool addresses and status, connection and translation information and status, TCP/UDP states, the translation table for NAT, the ARP table and many other details.



The network topology above shows a firewall failover pair in an Active/Standby setup. The "inside" interfaces are connected to the same internal switch and the "outside" interfaces to the same external switch. Also, a cross-over network cable is required between the two appliances as a LAN Failover Link. During normal operation, all traffic passes through the ACTIVE unit which controls all inbound and outbound communication. In the event of a failure of the active firewall (e.g interface failure, whole appliance failure etc), the STANDBY unit takes over by receiving the IP addresses of the ACTIVE unit so that traffic will continue to flow without interruption. All the connection state information is synchronized through the LAN Failover Link so that the STANDBY firewall unit has knowledge of the established flows when it takes over the traffic.

Failover Requirements:

There are several hardware and software requirements for the two firewall units in order to work in a failover configuration:

- Must be of the same platform model.
- Must have same hardware configuration (number and types of interfaces).
- Must be in the same operating mode (routed or transparent, single or multiple context).
- Must have same amount of Flash and RAM memory.
- Must have the same licensed features (e.g type of encryption supported, number of contexts, number of VPN peers supported etc).
- Proper Licensing. Cisco ASA models 5505 and 5510 support failover only with Security Plus license. With this license, the ASA 5505 supports stateless Active/Standby only. However, the ASA 5510 supports both Active/Standby and Active/Active failover modes. All the other

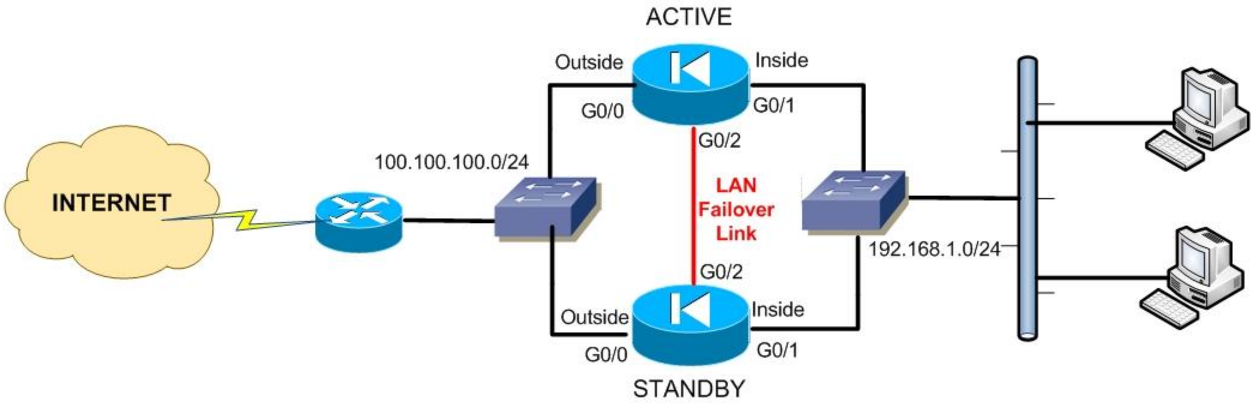
higher models (5520, 5540, 5550 etc) support both Active/Standby and Active/Active modes with the base license.

LAN Failover Link:

As shown on our example network schematic above, there is a dedicated LAN Failover Link between the two firewall appliances. This is a requirement for stateful failover configuration. A dedicated Ethernet interface must be reserved as a LAN Failover Link. This link can be a cross-over Ethernet cable connected directly between the two appliances.

In the next section we will discuss all technical details for configuring Stateful Active/Standby failover.

CONFIGURING ACTIVE/STANDBY STATEFUL FAILOVER



Returning to our example failover network topology, we will discuss the step-by-step process of configuring two ASA Firewalls in Active/Standby Stateful Failover setup.

- **STEP 1: Prepare the Primary (Active) Firewall**

Select one of the Firewall appliances to be the ACTIVE unit. Attach a network cable for each interface you plan to use on the Active Firewall unit and connect it to the appropriate switches. The Standby Firewall **must be disconnected** for now. Set the Active firewall interfaces to fixed speed and duplex mode. For example, use the commands **speed 100** and **duplex full** under Interface Configuration mode. Also, enable the PortFast feature on the switch ports connecting the Firewall interfaces.

Reserve **two** IP addresses for each Firewall network interface and decide which one will be assigned for the Active and which for the Standby unit. The two IP addresses for each interface must be in the same subnet. For example, in our network diagram above, assume that for the Inside interfaces we will use 192.168.1.1/24 for the ACTIVE firewall, and 192.168.1.2/24 for the STANDBY firewall. Also, for the Outside Interfaces we will use 100.100.100.1/24 for the ACTIVE and 100.100.100.2/24 for the STANDBY. Select also a private network subnet that will be used for the point-to-point Dedicated LAN Failover Link (Interface G0/2 in our example above). Assume that we will use 192.168.99.0/24.

- **STEP 2: Configure the LAN Failover Link on the Primary (Active) Firewall**

In our example topology, we will use the dedicated Gigabit Ethernet G0/2 as a LAN Stateful Failover Link. The command format for configuring the Failover link is the following:

ASA(config)# failover lan unit {*primary | secondary*} ←Set the unit as primary

ASA(config)# failover lan interface "*Failover Name*" "*Physical Interface*" ←Assign a physical interface as Failover link

ASA(config)# failover link "*Failover Name*" "*Physical Interface*" ←Enable the same Failover Link to be used for Stateful Failover as well.

ASA(config)# failover interface ip "*Failover Name*" "*ip_address*" "*netmask*" standby "*standby_ip_address*" ←Assign IP address to Active and Standby Failover interfaces

ASA(config)# failover ←Enable the failover mechanism

Example (for Primary Firewall):

```
ACTIVE-ASA(config)# interface GigabitEthernet0/2
ACTIVE-ASA(config-if)# no shut
ACTIVE-ASA(config)# failover lan unit primary
ACTIVE-ASA(config)# failover lan interface FAILOVER GigabitEthernet0/2
ACTIVE-ASA(config)# failover link FAILOVER GigabitEthernet0/2
ACTIVE-ASA(config)# failover interface ip FAILOVER 192.168.99.1 255.255.255.0 standby
192.168.99.2
ACTIVE-ASA(config)# failover
```

- **STEP 3: Configure Interface IP addresses on the Primary (Active) Firewall**

Each firewall interface in a failover pair must have two IP addresses assigned, one as the active address and another one as a standby address. Before configuring anything on the secondary firewall, we need to configure IP addresses on the Primary unit. The command format is:

```
ASA(config)# interface {Physical or Logical Interface}
ASA(config-if)# ip address "Active Unit IP" "netmask" standby "Standby Unit IP"
```

Example (for Primary Firewall):

```
ACTIVE-ASA(config)# interface GigabitEthernet0/1
ACTIVE-ASA(config-if)# nameif inside
ACTIVE-ASA(config-if)# security-level 100
ACTIVE-ASA(config-if)# ip address 192.168.1.1 255.255.255.0 standby 192.168.1.2
ACTIVE-ASA(config)# interface GigabitEthernet0/0
ACTIVE-ASA(config-if)# nameif outside
ACTIVE-ASA(config-if)# security-level 0
ACTIVE-ASA(config-if)# ip address 100.100.100.1 255.255.255.0 standby 100.100.100.2
```

- **STEP 4: Configure Monitoring on Interfaces of Primary (Active) Firewall**

One of the events that triggers the Failover mechanism is the failure of a Firewall Interface. We need to specify which Interfaces we want the appliance to monitor in order to switch over to the Standby unit when that Interface fails. In our example above we want to monitor both Inside and Outside firewall interfaces.

The command format is:
ASA(config)# monitor-interface "*Interface Name*"

Example (for Primary Firewall):

```
ACTIVE-ASA(config)# monitor-interface inside
ACTIVE-ASA(config)# monitor-interface outside
```

If either the "inside" or "outside" interfaces fail, the Active firewall will switch over to the Standby unit. You can exclude interfaces attached to less critical networks from affecting your failover mechanism by using the **no monitor-interface {*interface name*}** command.

- **STEP 5: Configure the LAN Failover Link on the Secondary (Standby) Firewall**

After the Primary security appliance is configured, we now need to configure the Secondary Unit. The only configuration required for the secondary appliance is the LAN Failover Link. Power on the secondary appliance and connect its interfaces to the appropriate switches. DO NOT connect the LAN Failover Link between the two firewalls yet. Connect with a console cable and setup the following commands:

Example (for Secondary Firewall):

```
ASA(config)# interface GigabitEthernet0/2
ASA(config-if)# no shut
ASA(config)# failover lan unit secondary
ASA(config)# failover lan interface FAILOVER GigabitEthernet0/2
ASA(config)# failover link FAILOVER GigabitEthernet0/2
ASA(config)# failover interface ip FAILOVER 192.168.99.1 255.255.255.0 standby
192.168.99.2
ASA(config)# failover
```

Notice that the only configuration difference we have with the Primary unit is the "**secondary**" keyword. Also, although we are configuring the Secondary unit, the IP address configuration for the failover interface must be the same as that of the Primary unit.

- **STEP 6: Reboot the Secondary (Standby) Firewall**

Use the **write memory** command to save the configuration of the Secondary Firewall. Connect the LAN Failover Link between the two Firewall appliances and use the **reload** command to reboot the secondary security appliance.

After the Secondary unit boots up, the Primary firewall configuration is replicated to the Secondary firewall. The following messages will appear on the Primary firewall:

Beginning Configuration Replication: Sending to Mate ←This denotes the start of the synchronization
End Configuration Replication to Mate ←This denotes the completion of synchronization

You need to enter the **write memory** command on the active firewall unit to save all the replicated configuration on both the active and standby units.

From now on, any additional configuration must be done only on the Primary Firewall unit, since it will be automatically replicated to the Secondary unit. The **write memory** command on the Primary firewall will save the configuration on both units.

Finally, use the **show failover** command to verify that the failover mechanism works as expected.

CHAPTER 7:
ADVANCED FEATURES OF DEVICE CONFIGURATION

CONFIGURING CLOCK AND NTP SUPPORT

The Cisco ASA appliance retains clock settings in memory via a battery on the device motherboard. Even if the device is turned off, the clock is retained in memory. Configuring accurate time settings on the appliance is important for logging purposes since syslog messages can contain a time stamp according to the device clock time setting. If you want the syslog messages to include a time-stamp value, you must first configure the clock (using **clock set** command) and then enable time-stamps using **logging timestamp** command (more on syslog configuration in later sections). Having a time-stamp value on log messages is important for event tracing and forensic purposes when a security incident occurs.

Another important reason for setting the correct time on the ASA firewall is when you use PKI (Public Key Infrastructure) with digital certificates for authentication of IPSEC VPN peers. The ASA firewall uses the local appliance clock to make sure that a Digital Certificate is not expired. When using PKI digital certificates, set the firewall clock to UTC time zone.

Configure Clock Settings:

To configure the clock settings of the ASA appliance, use the **clock set** command as shown below:

```
ciscoasa# clock set hh:mm:ss [day month | month day] year
```

Example:

```
ciscoasa# clock set 18:30:00 Apr 10 2009
```

To verify the correct clock on the appliance, use the **show clock** command.

Configure Time Zone and Daylight Saving Time:

To configure the time zone and the summer daylight saving time use the commands below:

```
ciscoasa# config t
ciscoasa(config)# clock timezone [zone name] [offset hours from UTC]
ciscoasa(config)# clock summer-time [zone name] recurring [week weekday month hh:mm week weekday month hh:mm] [offset]
```

Example:

```
ciscoasa(config)# clock timezone MST -7
ciscoasa(config)# clock summer-time MST recurring 1 Sunday April 2:00 last Sunday October 2:00
```

Configure Network Time Protocol (NTP):

If there is an NTP server in the network that provides accurate clock settings, then you can configure the firewall to synchronize its time with the NTP server. Both an authenticated and non-authenticated NTP is supported:

Non-Authenticated NTP:

```
ciscoasa(config)# ntp server [ip address of NTP] source [interface name]
```

Example:

```
ciscoasa(config)# ntp server 10.1.23.45 source inside
```

Authenticated NTP:

```
ciscoasa(config)# ntp authenticate
ciscoasa(config)# ntp authentication-key [key ID] md5 [ntp key]
ciscoasa(config)# ntp trusted-key [key ID]
ciscoasa(config)# ntp server [ip address of NTP] key [key ID] source [intf name]
```

Example:

```
ciscoasa(config)# ntp authenticate
ciscoasa(config)# ntp authentication-key 32 md5 secretkey1234
ciscoasa(config)# ntp trusted-key 32
ciscoasa(config)# ntp server 10.1.2.3 key 32 source inside
```

Configuring Logging (Syslog)

The Cisco ASA security appliance generates syslog messages for various events such as security alerts, resource depletion, traffic logs etc. You can configure what type of logging information (Logging Levels) will be generated by the firewall. Also, you can configure where the security appliance will sent syslog messages as we will see below. Before configuring anything else, use "**logging enable**" first to enable the firewall to generate syslog messages.

Configure Where the ASA will sent Syslog Messages:

Cisco ASA can send log messages to seven different destinations, either local or remote.

1. **Logging to SSH or Telnet session**:

If you want to monitor log messages while you are connected to the ASA via Telnet or SSH, use the "**logging monitor [logging level]**" command.

2. **Logging to Internal Buffer**:

There is a configurable internal log buffer memory where the security appliance can send syslog messages. Use the "**logging buffered [logging level]**" command to instruct the ASA to store log messages to its internal buffer. Use the "**logging buffer-size [bytes]**" to set the internal log buffer size in bytes. The default is 4KB. The following command sets the log buffer to 16KB: **logging buffer-size 16384.** To display the internal log buffer messages use the "**show logging**" command.

3. **Logging to Console**:

If you want to monitor syslog messages while you are connected to the ASA via the console port, use the "**logging console [logging level]**" command. You should be very careful with this logging option since the console is a slow-speed connection (9600 bps) and will degrade system performance if the ASA generates a lot of log messages.

4. **Logging to E-mail Address**:

The "**logging mail [logging level]**" command sends syslog messages to an email address.
Example:

```
ASA(config)# logging enable
ASA(config)# logging mail critical
ASA(config)# logging from-address ciscosecurityappliance@example.com
ASA(config)# logging recipient-address admin@example.com
```

5. **Logging to Adaptive Security Device Manager (ASDM)**:

The ASDM is the Graphical User Interface application to manage an ASA firewall. You can configure the appliance to send syslog messages to the ASDM GUI using “**logging asdm [logging level]**”.

6. **Logging to External Syslog Server**:

This is a great logging option since you can store and archive syslog messages for a longer period compared with the other options. Use the “**logging host [interface name] [syslog IP]**” to send log messages to an external syslog host. Use also the “**logging trap [logging level]**” command to specify the logging level.

Example:

```
ASA(config)# logging enable
ASA(config)# logging host inside 192.168.1.30
ASA(config)# logging trap errors
```

7. **Logging to SNMP Network Management System**:

If you have an NMS system in your network which collects SNMP alerts, you can send syslog messages as traps to the SNMP NMS system. Use the “**logging history [logging level]**” command.

Example:

```
ASA(config)# logging enable
ASA(config)# snmp-server host inside 10.1.1.100 trap community communityname
ASA(config)# snmp-server enable traps syslog
ASA(config)# logging history warnings
```

Configuring Logging Levels:

There are 8 configurable Logging Levels (0 to 7). For each command described above for the logging destination options, you should specify also a **logging level** after each command. Each logging level defines how much and what type of information will be logged by the appliance. The eight Logging Levels are:

0 – Emergencies: Generate System unusable messages.
1 – Alerts: Take immediate action messages.
2- Critical: Generate Critical condition messages.
3 – Errors: Generate Error messages.
4 – Warnings: Generate Warning messages.
5 – Notifications: Generate normal but significant condition messages.
6 – Informational: Generate information messages.
7 – Debugging: Generate debug messages and log FTP and WWW commands.

For each Logging Level that you configure, all lower number levels are enabled as well. For example if you enable Logging Level 4 (Warnings), then Logging Levels 0,1,2,3 are also enabled.

Example:

```
ASA(config)# logging enable
ASA(config)# logging timestamp ← attach timestamp to log messages
ASA(config)# logging buffer-size 8096 ← set log buffer to 8KB
ASA(config)# logging buffered warnings ← send syslog warning messages to log buffer
ASA(config)# logging asdm errors ← send syslog error messages to ASDM
```

Customize Syslog Output:

Sometimes there are unwanted syslog messages that flood the security appliance logs. You can block these unwanted syslog messages from appearing in the logs by using the "**no logging message [syslog_id]**" command. For example, assume the ASA is flooded with a syslog message id 710005 (NetBIOS traffic). To block the output of this syslog message configure the following:

```
ASA(config)# no logging message 710005
```

Displaying Syslog Settings:

To display the current syslog settings (Logging Levels, log destinations etc) and to also monitor the Internal Log Buffer messages, use the "**show logging**" command.

Configuring Device Access Authentication Using Local Username/Password

In this section we will examine how to configure the security appliance to require authentication for administrator users when they try to connect to the ASA firewall for management. You can configure usernames and passwords locally on the ASA or have an external AAA server (RADIUS or TACACS) which will hold the username/passwords database. In this section we will discuss only Local authentication. The next Chapter will describe Authentication using an external AAA server.

Authentication can be configured for all management access connections, i.e. **Telnet, SSH, Serial, and HTTP**. Also, the "**Enable**" option can be used to request a username and password before accessing Privileged Mode for Serial, Telnet, and SSH connections.

Configure Authentication using the Local username database:

1. First Configure a Local username/password pair:

```
ASA(config)# username [name of user] password [user password]
```

2. Then Configure the ASA firewall to request authentication from LOCAL Database

```
ASA(config)# aaa authentication [serial|telnet|ssh|http|enable] console LOCAL
```

Example:

```
ASA(config)# username admin password cisco123
ASA(config)# aaa authentication serial console LOCAL
ASA(config)# aaa authentication telnet console LOCAL
ASA(config)# aaa authentication ssh console LOCAL
ASA(config)# aaa authentication enable console LOCAL
```

NOTE: The "**console**" keyword in the commands above does not refer to the console cable that we use for serial access.

- **serial Parameter:** Causes the user to be prompted continually by default until that user successfully logs in with the correct username/password specified in the configuration. You can limit the maximum failed attempts as we will see below. The **serial** option is for users connecting with the serial console cable.
- **telnet Parameter:** Causes the user to be prompted continually by default until that user successfully logs in. You can limit the maximum failed attempts as we will see below. The **telnet** option requests a username/password for users connecting with telnet before the first command-line prompt.
- **ssh Parameter:** Allows three tries before stopping access attempts. The **ssh** option requests a username and password for users connecting with ssh before the first command-line prompt.
- **enable Parameter:** Allows three tries before stopping access attempts. The **enable** option requests a username and password before accessing privileged mode for Serial, Telnet, and SSH connections.

Configure Maximum Failed Attempts:

For Serial and Telnet connections, the ASA firewall will continually ask the user for username/password until the correct authentication is entered. This is a security problem since an attacker may use Brute Force attack to gain access to the appliance. It is strongly suggested to configure a limit on the maximum failed attempts, as shown below:

```
ASA(config)# aaa local authentication attempts max-fail [fail-attempts number]
```

Example:

```
ASA(config)# aaa local authentication attempts max-fail 5
```

Also:

Use the command "**show aaa local user**" to see if a specific user is locked out.

Use the command "**clear aaa local user lockout all**" to clear the lockout status.

CHAPTER 8:
AUTHENTICATION AUTHORIZATION ACCOUNTING (AAA)

AAA is a suit of control mechanisms that are used by network devices to control user access to the network. Authentication is the most common mechanism and is used to verify who the user is. Authorization is used to control what the user can do in the network, and Accounting is used to report what the user did in the network (audit-trail). In this Chapter we will focus mostly on Authentication using an External AAA Server.

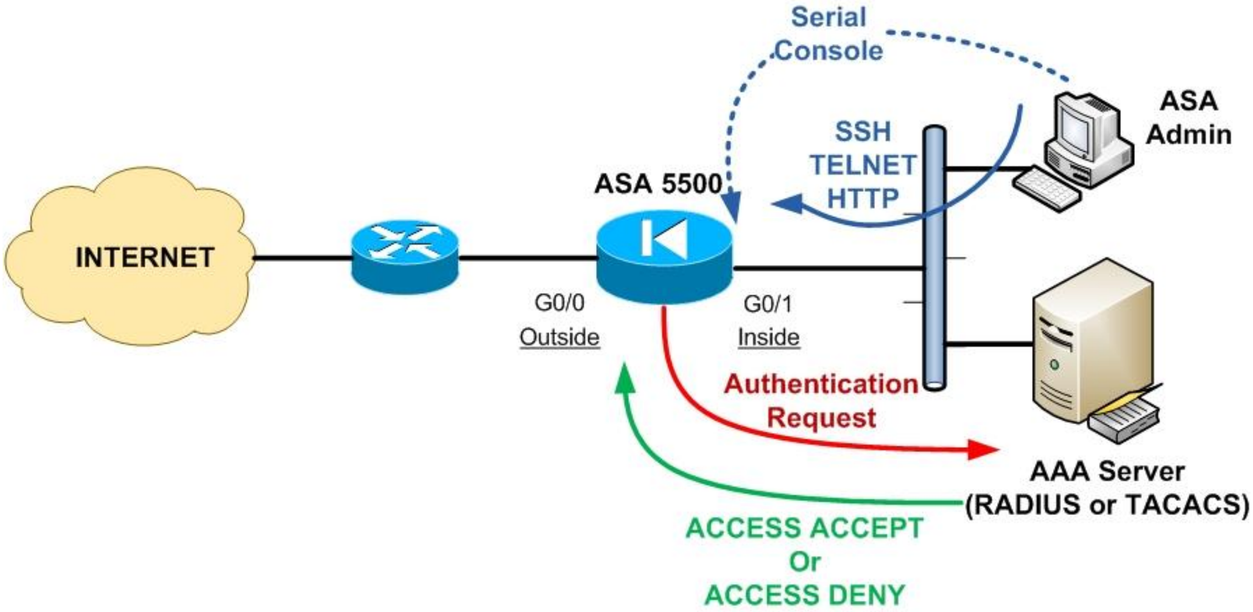
In the last section of the previous Chapter we described Authentication using the Local ASA user database. In this Chapter we will describe Authentication using an External AAA Server, such as the Cisco Access Control Server (ACS). That is, we will see how to configure the ASA firewall to Authenticate users utilizing an external AAA server.

For the Cisco ASA appliance we have three types of Authentication:

1. Authentication of users when accessing the security appliance itself (Device Access Authentication).
2. Authentication of users when accessing HTTP, HTTPs, Telnet and FTP services through the security appliance. This is called also cut-through proxy.
3. Authentication of users from remote access through an IPSEC or SSL VPN tunnel (Tunnel Access Authentication).

DEVICE ACCESS AUTHENTICATION USING EXTERNAL AAA SERVER

Next we will describe how to control Administrative access to the appliance using an external AAA Server. As mentioned above, an example of a AAA Server is the Cisco Secure ACS Server (Access Control Server) which supports both RADIUS and TACACS+ Authentication Protocols. A “AAA” server provides a centralized solution for offering authentication services to all of your network devices (ASA Firewalls, Routers, Switches etc). Basically the biggest advantage of a centralized AAA server is that you can keep a central database of username/passwords so that you don’t have to configure Local Usernames/Passwords on EACH of your network devices, thus minimizing administration effort and enhancing overall authentication security.



In the diagram above, the ASA Admin workstation can access the firewall using Serial Console cable, or through the network using SSH, TELNET, HTTP. Before allowing access, the ASA will prompt the admin user for his/her credentials. The username/password credentials supplied by the Admin will be sent by the ASA to the AAA Server as an **Authentication request**. If the credentials are valid, the AAA server will reply with "**ACCESS ACCEPT**" so that the ASA will allow access to the Admin user.

NOTE:

Before the ASA firewall can authenticate a TELNET, SSH, or HTTP access session, you must first configure the security appliance to allow those management protocols using the **telnet**, **ssh**, and **http** commands.

Example:

```
ASA(config)# ssh 10.1.1.0 255.255.255.0 dmz ← allow ssh from dmz subnet 10.1.1.0
ASA(config)# telnet 10.2.2.0 255.255.255.0 inside ← allow telnet from inside subnet 10.2.2.0
ASA(config)# http server enable
ASA(config)# http 10.2.2.50 255.255.255.255 inside ← allow http from inside host 10.2.2.50
```

SSH access can be used on all security level interfaces of the ASA (inside, outside, dmz etc). Telnet access is ONLY allowed on the inside interfaces.

Configure Authentication using an external AAA Server:

1. First specify a AAA server group:

```
ASA(config)# aaa-server [server-tag] protocol [radius|tacacs+]
```

2. Then designate an authentication server. You need to define the IP address of the AAA server and a pre-shared security key which must be configured also on the AAA server.

```
ASA(config)# aaa-server [server-tag] [ASA interface name] host [IP address of AAA]
ASA(config-aaa-server-host)# key [preshared secret key]
```

3. Then Configure the ASA firewall to request authentication from the AAA server

```
ASA(config)# aaa authentication [serial|telnet|ssh|http|enable] console [server-tag] [LOCAL]
```

Example:

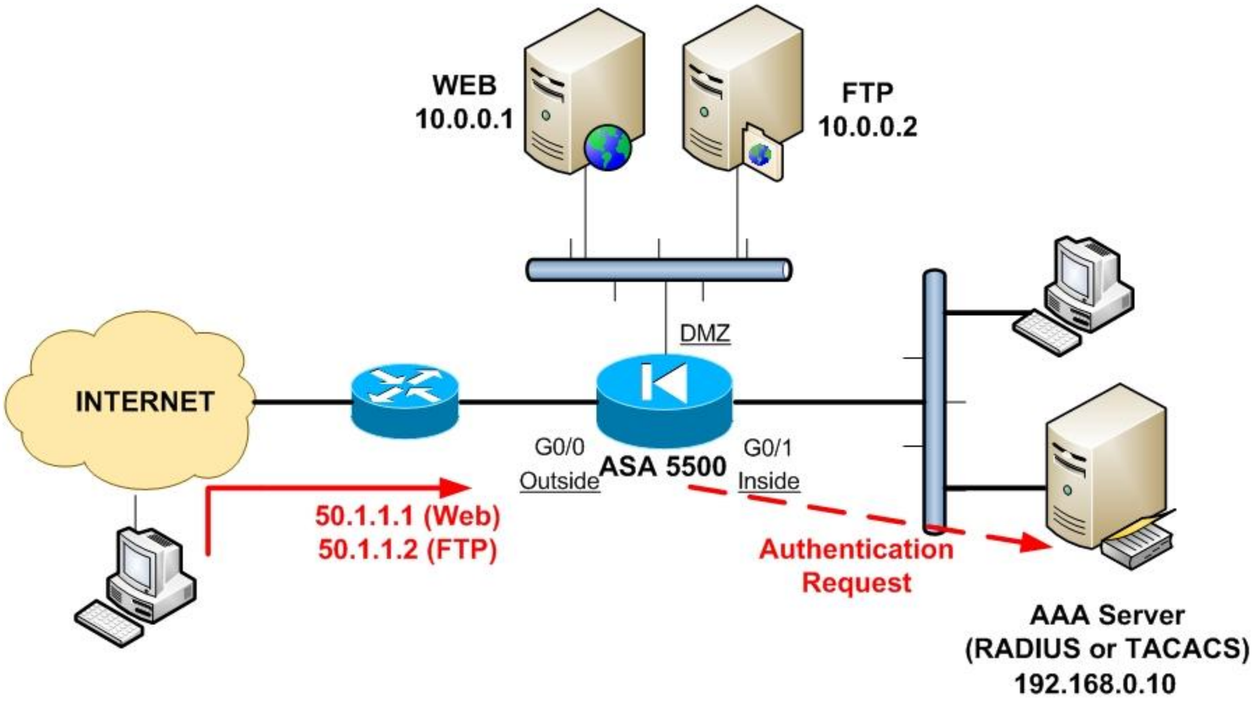
```
ASA(config)# username admin password cisco123 ← configure LOCAL username/password
ASA(config)# aaa-server ACSSRV protocol tacacs+ ← designate tacacs+ as auth. protocol
ASA(config)# aaa-server ACSSRV (inside) host 10.1.1.1
ASA(config-aaa-server-host)# key sharedsecret
ASA(config-aaa-server-host)# exit
ASA(config)# aaa authentication serial console ACSSRV LOCAL ← specify LOCAL as backup
ASA(config)# aaa authentication ssh console ACSSRV LOCAL
ASA(config)# aaa authentication enable console ACSSRV LOCAL
ASA(config)# ssh 10.1.1.0 255.255.255.0 inside ← enable ssh access on inside interface
```

NOTE:

It is strongly recommended to specify LOCAL authentication also in addition to the AAA server-tag. This means that if the AAA server is not available for any reason, the ASA firewall will use the LOCAL username/password as a backup authentication.

Cut-Through Proxy Authentication for TELNET,FTP,HTTP(s)

The Cut-through proxy feature of the security appliance allows the ASA to transparently verify the identity of users when accessing Telnet, FTP, HTTP and HTTPs services. The firewall first intercepts the Telnet/FTP/HTTP(s) session and authenticates the user identity against a AAA server. If the authentication is successful, the user session is redirected to the destination server. If the destination server has its own authentication, you must enter another username and password. I will not get into many details about the cut-through proxy feature because I have not seen it used very often in real networks, however it might be helpful in some situations (especially for HTTP authentication for example).



Let's see a scenario for cut-through proxy. From figure above, the Web Server (10.0.01) in DMZ is mapped (static nat) as 50.1.1.1 on the outside. Similarly, the FTP server (10.0.0.2) is mapped as 50.1.1.2 on the outside. When a user on the Internet tries to access either the Web or the FTP server, the ASA will generate an authentication prompt for the user. After the user enters his/her credentials, the ASA will query the AAA server for Authentication. If authentication is successful, the user session will be "cut-through" the security appliance and redirected to the destination server.

When using cut-through proxy, make sure that the inbound ACL allows the connection first. If the inbound ACL drops the connection from outside, then cut-through proxy authentication will not take place.

Configure cut-through proxy Authentication using an external AAA Server:

1. First specify a AAA server group:

ASA(config)# aaa-server *[server-tag]* **protocol** *[radius|tacacs+]*

2. Then designate an authentication server. You need to define the IP address of the AAA server and a pre-shared security key which must be configured also on the AAA server.

ASA(config)# aaa-server *[server-tag] [ASA interface name]* **host** *[IP address of AAA]*

ASA(config-aaa-server-host)# key *[preshared secret key]*

3. Then enable cut-through proxy authentication by specifying which traffic flow to authenticate.

ASA(config)# aaa authentication match *[ACL name] [interface name*] [AAA server-tag]*

**[interface name]* is where the connection originates

Let's see the following example which is based on the network diagram shown above.

Example:

```
ASA(config)# static (DMZ , outside) 50.1.1.1  10.0.0.1 netmask 255.255.255.255← Web
ASA(config)# static (DMZ , outside) 50.1.1.2  10.0.0.2 netmask 255.255.255.255← FTP

ASA(config)# access-list OUTSIDE-IN extended  permit tcp any host 50.1.1.1 eq 80 ← allow traffic to reach the web server from outside
ASA(config)# access-list OUTSIDE-IN extended  permit tcp any host 50.1.1.2 eq 21 ← allow traffic to reach the FTP server from outside
ASA(config)# access-group OUTSIDE-IN in interface outside

ASA(config)# aaa-server ACSSRV protocol radius ← designate radius as auth. protocol
ASA(config)# aaa-server ACSSRV (inside) host 192.168.0.10
ASA(config-aaa-server-host)# key sharedsecret
ASA(config-aaa-server-host)# exit
ASA(config)# access-list 101 permit tcp any host 50.1.1.1 eq www
ASA(config)# access-list 101 permit tcp any host 50.1.1.2 eq ftp
ASA(config)# aaa authentication match 101 outside ACSSRV ← enable cut-through proxy for traffic originating from “outside” and matching ACL 101. Use ACSSRV server for authentication.
```

CHAPTER 9:
ROUTING PROTOCOL SUPPORT

Firstly you need to know that the ASA appliance is not a full-functioning router. However, it still has a routing table which is used to select the best path to reach a certain destination network. After all, if a packet successfully passes all firewall rules, it needs to be routed by the firewall to its destination.

The Cisco ASA Firewall appliance supports both Static and Dynamic Routing. Three dynamic routing protocols are supported, namely RIP, OSPF, and EIGRP. It is highly recommended to prefer static routing configuration on the ASA firewall, instead of dynamic routing. This is because the usage of dynamic routing protocols might expose your internal network structure to the outside world. If you are not careful with dynamic routing configuration, it is possible to start advertising your internal network subnets to external untrusted networks, thus revealing your hidden networks to the outside world.

However, there are situations where dynamic routing configuration is necessary. Such a case would be a large network in which the ASA firewall is located within the internal network campus or data center. In such a case, you will benefit from using a dynamic routing protocol on the ASA since you will not have to configure tons of static routes, and also you will not run into the risk of revealing any hidden subnets to untrusted networks (since the ASA is located deep inside the campus network).

The following are some routing protocol best practices for the ASA:

- For small networks, use only static routes. Use a default static route pointing to the gateway connected to the outside interface (usually Internet), and also use static routes for internal networks which are more than one hop away (i.e not directly connected).
- Any network that is directly connected on an ASA interface DOES NOT need any static route configuration since the ASA firewall already knows how to reach this network.
- If the ASA is connected on the perimeter of the network (i.e border between trusted and untrusted networks), then configure a default route towards the outside untrusted zone, and then configure specific static routes towards the internal networks.

- If the ASA is located deep inside a large network campus with many internal network routes, then configure a dynamic routing protocol.

STATING ROUTING

There are three types of static routes:

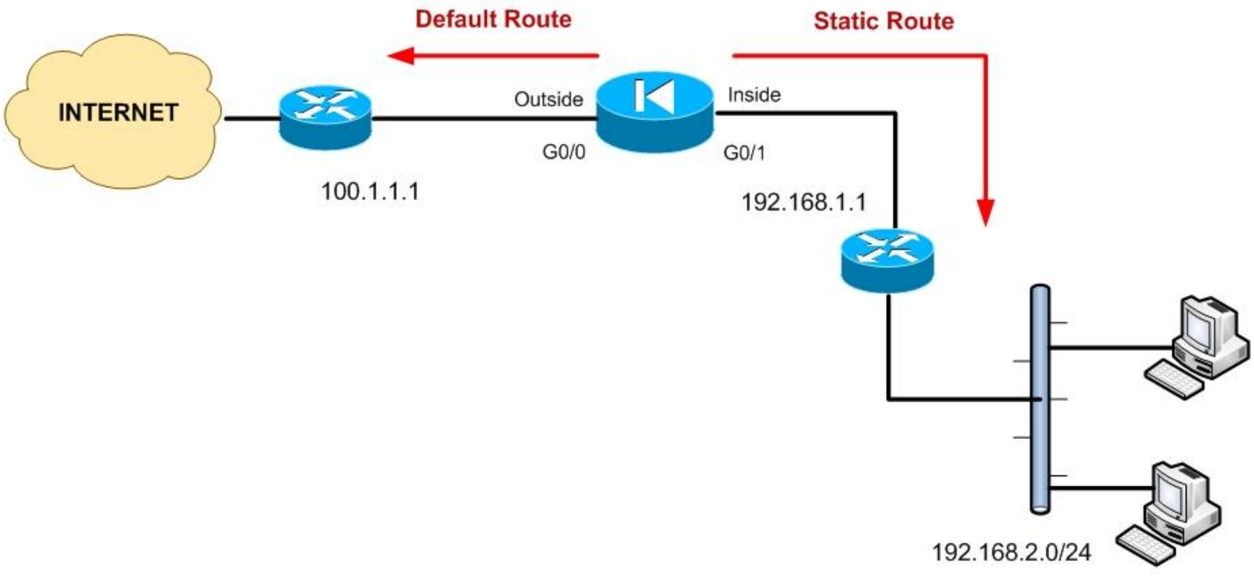
- Directly Connected Route
- Normal Static Route
- Default Route

Directly Connected Route

The Directly Connected Route is automatically created in the ASA routing table when you configure an IP address on an appliance interface. For example, if you configure the IP address 192.168.1.10/24 on the inside interface of ASA, then a Directly Connected Route of 192.168.1.0 255.255.255.0 will be automatically created.

Normal Static Route and Default Route

For configuring a Normal Static Route and Default Static Route refer to the diagram below.



A static route configuration on the ASA is like telling the appliance the following: "To send a packet to the specified network, give it to this router gateway".

Use the **route** command to enter either a static or default route. The command format is:

```
ASA(config)# route [interface-name] [destination-network] [netmask] [gateway]
```

[interface-name]: This is the ASA interface from which the packet will exit.
[destination-network] [netmask]: This is the destination network/mask we want to reach
[gateway]: Next hop device that ASA will send the packet to.

Let's see an example configuration below (refer to diagram above):

```
ASA(config)#  route outside 0.0.0.0 0.0.0.0 100.1.1.1 ← Default Route
ASA(config)#  route inside 192.168.2.0 255.255.255.0 192.168.1.1 ← Static Route. To reach
network 192.168.2.0 send the packets to 192.168.1.1
```

For the default route (usually towards the Internet), you set both the *destination-network* and *netmask* to 0.0.0.0. All traffic for which the ASA has no route in its routing table will be sent to 100.1.1.1 (the gateway in the default route).

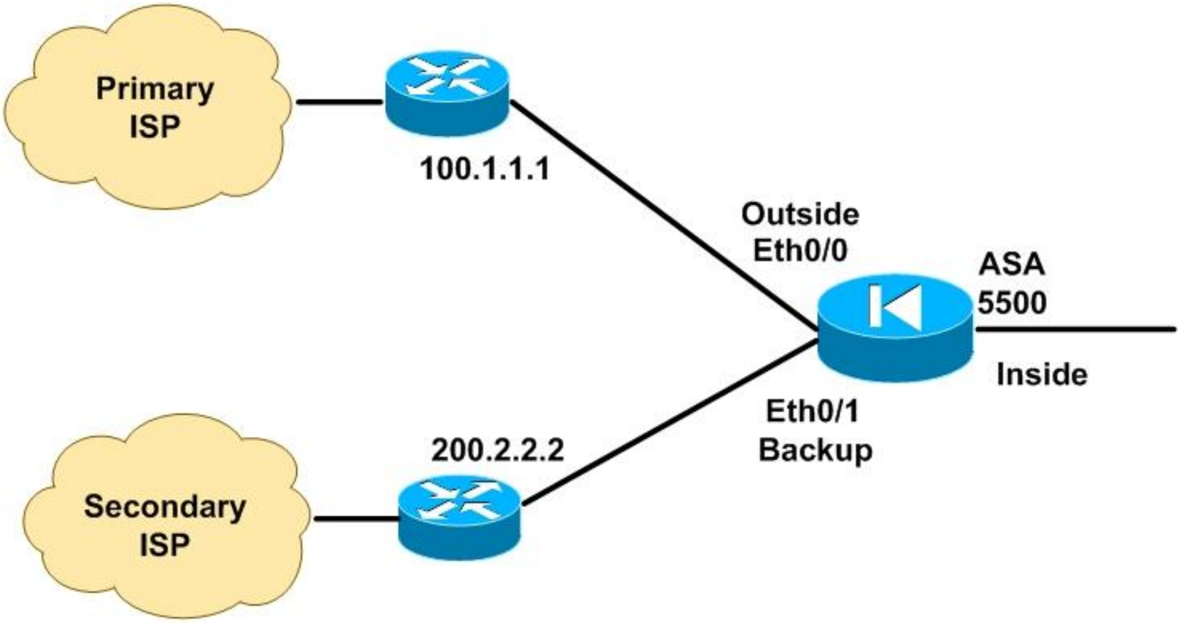
To see what is included in the appliance's routing table, use the "**show route**" command:

```
ASA# show route
S 0.0.0.0 0.0.0.0 [1/0] via 100.1.1.1, outside ← Default Static Route
C 192.168.1.0 255.255.255.0 is directly connected, inside ← Connected Route
C 100.1.1.0 255.255.255.0 is directly connected, outside ← Connected Route
S 192.168.2.0 255.255.255.0 [1/0] via 192.168.1.1, inside ← Static Route
```

Static Route Tracking

When you configure a static route on the security appliance, the route remains permanently in the routing table. The only way for the static route to get removed from the routing table is when the associated ASA interface goes physically down. In all other cases, such as for example when the remote default gateway goes down, the ASA will keep sending packets to its gateway router without knowing that it is actually down.

From ASA version 7.2 and upwards, the **Static Route Tracking** feature was introduced. The ASA tracks the availability of static routes by sending ICMP echo request packets through the primary static route path and waits for replies. If the primary path is down, a secondary path is used. This feature is useful when you want to implement Dual-ISP redundancy, as we will see in the scenario below.



In the network scenario above, interface Eth0/0 (outside) is connected to the Primary ISP and interface Eth0/1 (backup) is connected to the Secondary ISP. Two default static routes will be configured (one for each ISP) which will use the "**track**" feature. The primary ISP path will be tracked using ICMP echo requests. If an echo reply is not received within a predefined period, the secondary static route will be used. Note however that the scenario above is suitable only for outbound communication (that is, from the inside network towards the Internet).

Configuring Static Route Tracking

1. Use the "**sla monitor**" command to specify the monitoring protocol (e.g ICMP), the target address to track (e.g ISP gateway router) and the tracking timers.
2. Use the "**sla monitor schedule**" command to schedule the monitoring process (usually the monitoring process is configured to run "**forever**" but duration and start times are configurable).
3. Define the primary static route to be tracked using the "**route**" command with the "**track**" option.
4. Define the backup static route and set its metric higher than the primary static route.

Let’s see an example configuration below (related to the diagram shown above)

```
ASA(config)# global (outside) 1 interface
ASA(config)# global (backup) 1 interface
ASA(config)# nat (inside) 1 0.0.0.0 0.0.0.0
ASA(config)# sla monitor 100 ← Define SLA_ID 100
ASA(config-sla-monitor)# type echo protocol ipIcmpEcho 100.1.1.1 interface outside
ASA(config-sla-monitor)# timeout 3000 ← Define timeout 3000 milliseconds (3 sec)
ASA(config-sla-monitor)# frequency 5 ← track target 5 times
ASA(config-sla-monitor)# exit
ASA(config)# sla monitor schedule 100 life forever start-time now ← Schedule the
monitoring process SLA_ID 100 to start now and run forever
ASA(config)# track 10 rtr 100 reachability ← Associate a Track_ID 10 with the SLA_ID 100
ASA(config)# route outside 0.0.0.0 0.0.0.0 100.1.1.1 1 track 10 ← Associate the Track_ID 10
to the primary static route. Define also a metric 1 for this route.
ASA(config)# route backup 0.0.0.0 0.0.0.0 200.2.2.2 254 ← Define the backup static route
with a higher route metric of 254
```

In the scenario above, the firewall appliance will be tracking the primary ISP gateway router (100.1.1.1) . If an echo reply is not received within 3 sec (timeout 3000 milliseconds) and the process is repeated 5 times (frequency 5), the primary default route is considered down and therefore the secondary backup route will be used.

Dynamic Routing using RIP

RIP is one of the oldest dynamic routing protocols. Although it is not used a lot in modern networks, you still find it in some cases. Cisco ASA version 7.x supports RIP in a limited fashion. The ASA appliance (v7.x) can only accept RIP routes and optionally advertise a default route. However, it cannot receive RIP advertisements from one neighbor and then advertise these routes to another neighbor. From ASA version 8.x however, the security appliance supports full RIP functionality. Both RIPv1 and RIPv2 are supported. However, using RIPv1 is not recommended because it does not support routing updates authentication.

Configuring RIP

Configuration of RIP on the ASA appliance is similar with a Cisco router. RIP is configured using the "**router rip**" Global Configuration command. RIP authentication security is configured under Interface Configuration.

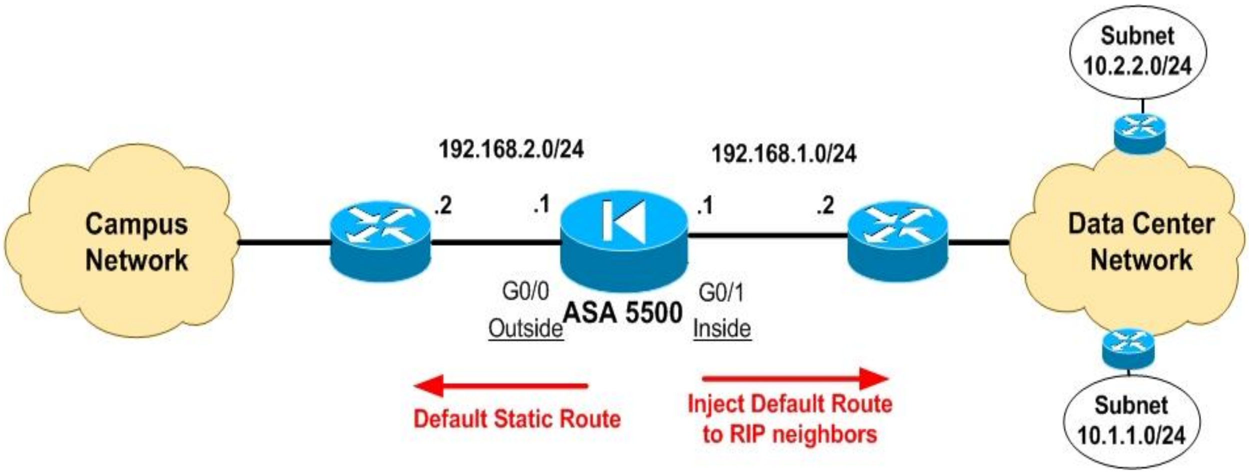
```
ASA(config)# router rip
ASA(config-router)# network [network-subnet] ← network to advertise via RIP
ASA(config-router)# version [1 | 2] ← select RIP version
ASA(config-router)# default-information originate ← Inject a default route into the network
ASA(config-router)# passive-interface [ASA interface name] ← disable RIP updates propagation on specified interface
ASA(config-router)# no auto-summarize ← disable automatic route summarization
```

The "**no auto-summarize**" command works only for RIPv2. It disables automatic route summarization to their network Class boundary. For example if you have a route 10.1.3.0/24 which you want to advertise via RIP, by default it will be advertised as 10.0.0.0/8 by the ASA. Using the "**no auto-summarize**" command, the route will be advertised as 10.1.3.0/24.

Regarding RIP updates authentication, this is configured on a per Interface basis:

```
ASA(config)# interface [interface number]
ASA(config-if)# rip authentication mode [text | md5] ← I suggest to always use md5 auth.
ASA(config-if)# rip authentication key [secret key] key-id [key ID number] ← Use the same secret authentication key to all neighbor devices running RIP. [secret key] can be up to 16 characters, and [key ID number] is a number between 0-255
```

The diagram below shows an example network topology with an ASA firewall running RIP within a network with other routers.



Assume the ASA is located between the Campus Network and the Data Center Network. All router neighbors behind the inside interface are running RIP.

Configuration Example:

```
ASA(config)# route outside 0.0.0.0 0.0.0.0 192.168.2.2
ASA(config)# router rip
ASA(config-router)# network 192.168.1.0
ASA(config-router)# version 2
ASA(config-router)# default-information originate
ASA(config-router)# exit
ASA(config)# interface GigabitEthernet0/1
ASA(config-if)# rip authentication mode md5
ASA(config-if)# rip authentication key somesecrethere key-id 10
```

DYNAMIC ROUTING USING OSPF

OSPF (Open Shortest Path First) is a dynamic routing protocol based on Link States rather than Distance Vectors (such as RIP) for optimal path selection. It is a much better and more scalable routing protocol compared to RIP, that's why is widely used in large Enterprise networks. OSPF can be very complex and one can write a whole book for it. In this section I will keep OSPF discussion as brief as possible, and I will try to discuss features and scenarios that are most commonly used in real networks. (Note: IPv6 is not currently supported on Cisco ASA running OSPF.)

Configuring OSPF

OSPF is based on Areas. In brief, to configure OSPF you need to create an OSPF routing process (up to two routing processes can be configured on ASA), specify the IP network addresses associated with the routing process, and then assign area IDs associated with each IP network address. Similarly with RIPv2, we can also configure MD5 authentication for OSPF updates security.

```
ASA(config)# router ospf [process ID] ← enable the ospf routing process
ASA(config-router)# network [IP address] [subnet mask] area [area ID] ← IP network address to advertise via OSPF. This network address must belong in a specific OSPF Area
```

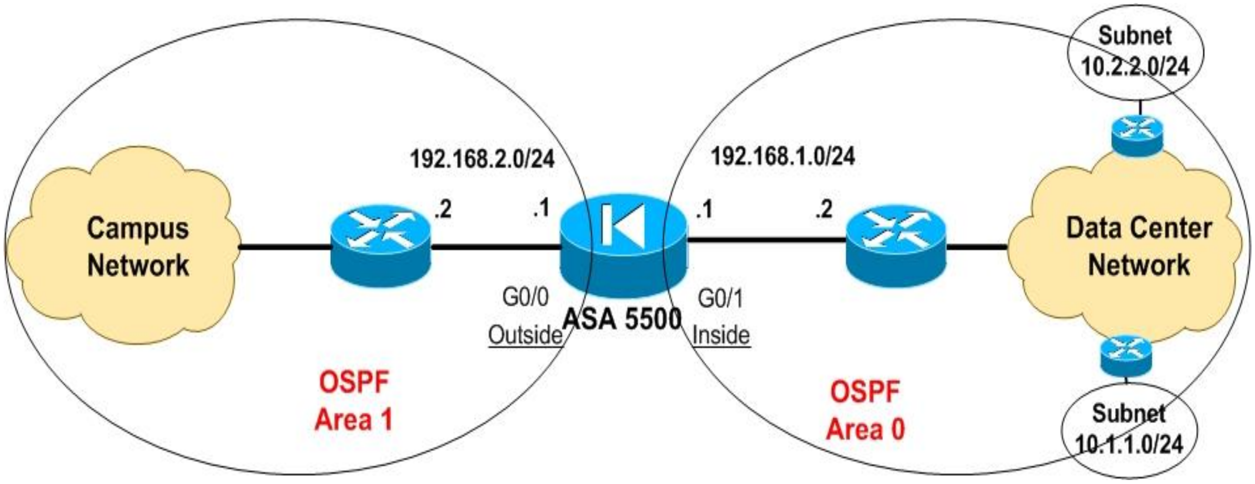
Note that "subnet mask" above must be a normal subnet mask (such as 255.255.255.0) and NOT an inverse (wildcard) subnet mask like we use in Cisco routers (such as 0.0.0.255).

To configure OSPF MD5 authentication, you need to enable authentication per Area (within the routing process) and also configure the MD5 authentication key under Interface configuration.

```
ASA(config)# router ospf [process ID]
ASA(config-router)# area [area ID] authentication message-digest ← Enable MD5 authentication in the specific Area
ASA(config-router)# exit
ASA(config)# interface [interface number]
ASA(config-if)# ospf authentication message-digest
ASA(config-if)# ospf message-digest-key [key ID] md5 [secret key]
```

We will see two OSPF example scenarios which are commonly used in real implementations. The first example depicts a Cisco ASA within an Enterprise network working as an ABR (Area Border Router), and the second example shows an ASA firewall injecting a default route into the internal network via OSPF.

Scenario 1: ASA working as OSPF ABR

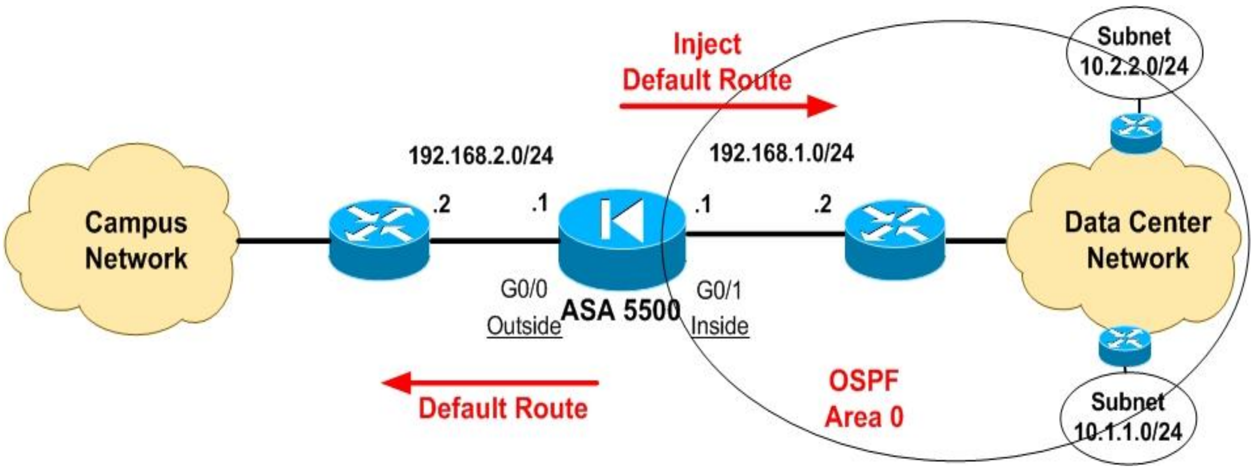


In the example above, the ASA5500 firewall is located between the Data Center and Campus Networks. All routers within the Data Center network are running OSPF in Area 0. On the other hand, all routers in Campus Network are running OSPF in Area 1. The ASA works as Area Border Router. We assume also that there is no NAT configured on the ASA ("**no nat-control**"). Firewall policies can be enforced using Access-Lists on both the Inside and Outside interfaces.

Configuration Example:

```
ASA(config)# router ospf 10
ASA(config-router)# network 192.168.1.0 255.255.255.0 area 0
ASA(config-router)# network 192.168.2.0 255.255.255.0 area 1
ASA(config-router)# area 0 authentication message-digest
ASA(config-router)# area 1 authentication message-digest
ASA(config-router)# exit
ASA(config)# interface GigabitEthernet0/0
ASA(config-if)# ospf authentication message-digest
ASA(config-if)# ospf message-digest-key 20 md5 somesecretkey
ASA(config-if)# exit
ASA(config)# interface GigabitEthernet0/1
ASA(config-if)# ospf authentication message-digest
ASA(config-if)# ospf message-digest-key 20 md5 somesecretkey
```

Scenario 2: ASA Injecting a default route in OSPF network



In the example above, the ASA has a default route towards the Campus Network and injects this default route in the inside network (Data Center). This means that all routers within the Data Center network (which will be running OSPF in Area 0) will acquire a default route which will point to their next-hop closest to the ASA.

Configuration Example:

```
ASA(config)# route outside 0.0.0.0 0.0.0.0 192.168.2.2
ASA(config)# router ospf 10
ASA(config-router)# network 192.168.1.0 255.255.255.0 area 0
ASA(config-router)# default-information originate always ← Inject default route
ASA(config-router)# area 0 authentication message-digest
ASA(config-router)# exit
ASA(config)# interface GigabitEthernet0/1
ASA(config-if)# ospf authentication message-digest
ASA(config-if)# ospf message-digest-key 20 md5 somesecretkey
ASA(config-if)# exit
```

Dynamic Routing using EIGRP

EIGRP is the enhanced version of the older IGRP. It is a Cisco proprietary protocol which runs only between Cisco devices. Support for EIGRP on Cisco ASA was included from version 8.0 and later. Although EIGRP is very easy to use and flexible, network designers and administrators hesitate to use it widely since it works only with Cisco equipment, so you are effectively dependent on a single vendor. I have not seen this protocol used a lot on Cisco ASA firewalls, so I will keep the discussion just to the basics. (Note: IPv6 is not currently supported on Cisco ASA running EIGRP.)

Configuring EIGRP

EIGRP configuration on a Cisco ASA is very similar with a Cisco router. Basically you just enable the EIGRP process by assigning it an AS number, and then configure the IP network ranges that will be advertised by the routing protocol to other EIGRP neighbors.

```
ASA(config)# router eigrp [AS Num] ← enable the EIGRP routing process
ASA(config-router)# network [IP address] [subnet mask] ← IP network address to advertise
```

MD5 authentication for EIGRP updates is configured under Interface config mode as shown below:

```
 ASA(config)# interface [interface number]
ASA(config-if)# authentication mode eigrp [AS-num] md5
ASA(config-if)# authentication key eigrp [AS-num] [key] key-id [key ID]
```

Note: All neighbor routers must belong in the same AS number and have the same MD5 key. The *[key ID]* is just a number between 0-255

Configuration Example:

```
ASA(config)# router eigrp 2 ← we are in Autonomous System 2
ASA(config-router)# network 192.168.1.0 255.255.255.0
ASA(config-router)# network 192.168.2.0 255.255.255.0
ASA(config-router)# exit
ASA(config)# interface Ethernet0/2
ASA(config-if)# authentication mode eigrp 2 md5
ASA(config-if)# authentication key eigrp 2 somesecretkey key-id 20
```

This concludes our discussion on routing protocol support.

CHAPTER 10: MODULAR POLICY FRAMEWORK CONFIGURATION

In this Chapter we will see the key concepts behind Modular Policy Framework (MPF). MPF is quite complex and extensive so I will only describe the basic features of it and the most useful concepts as implemented in real world networks.

MPF OVERVIEW

The Modular Policy Framework provides greater granularity and flexibility in implementing network and security policies with the ASA appliance. The MPF mechanism can be used for example to apply Quality of Service (prioritization) for voice traffic, to rate-limit specific remote access VPN connections, to apply TCP connection limits to specific traffic flows, to apply deep packet (Layer 7) inspection on specific flows of traffic etc.

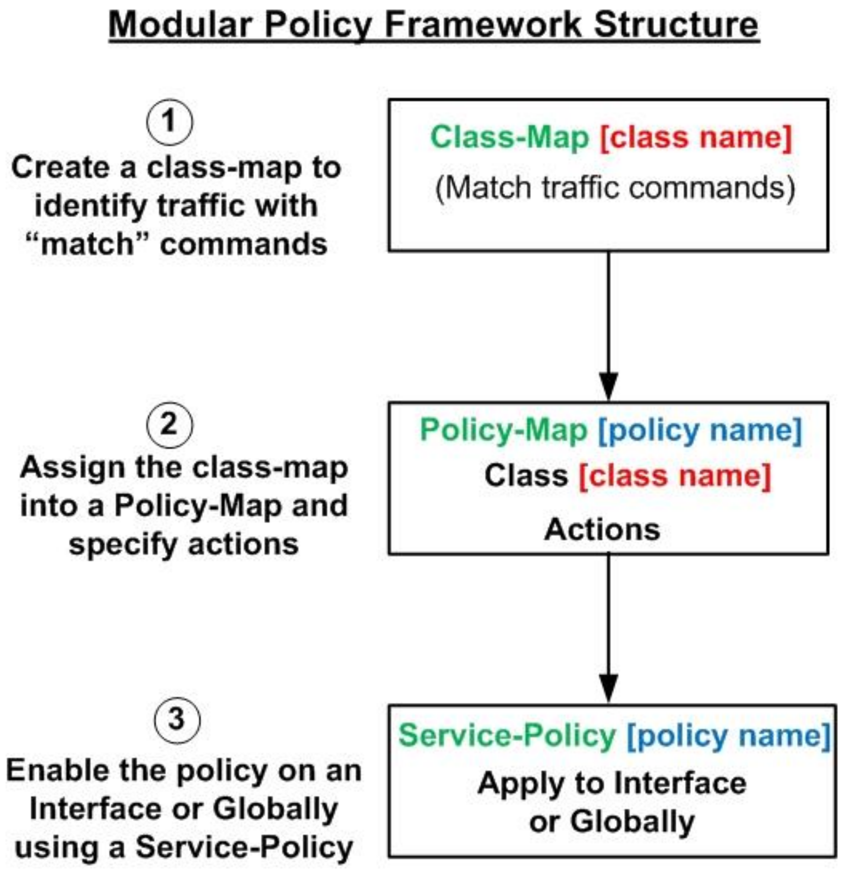
When configuring MPF, the traffic is first identified (traffic matching) with a Class-Map, then actions are applied to the matched traffic using a Policy-Map, and finally the whole policy is enabled on an interface or globally using a Service-Policy.

As described above, there are three main components of a Modular Policy Framework: A **Class-Map** component, a **Policy-Map** component and a **Service-Policy** component.

- **Class-Map**: This is used to identify a traffic flow that we want to apply policies on. You can create either a Layer3/4 Class Map or a Layer 7 Class Map. In this Chapter we will focus only on Layer3/4 class maps. This type of class map matches traffic based on protocols, ports, IP addresses and other Layer3/4 characteristics of the traffic flow. On the other hand, a Layer7 Class Map matches traffic based on application characteristics (for example a certain URL name in an HTTP traffic flow or even a certain FTP command in an FTP connection).
- **Policy-Map**: After the firewall appliance identifies the traffic flow with a Class-Map, a Policy-Map is used to apply certain actions (or policies) to the selected class of traffic. An example of a policy-map is to limit the maximum number of TCP connections towards a Web Server on the DMZ to a certain number. Another example of a policy-map is to apply high priority to voice packets between two sites. Similarly with Class-Maps, an administrator can create a Layer3/4 Policy-Map or a Layer 7 Policy-Map.

- **Service-Policy**: The Service-Policy component is used to apply the configured policy framework to an Interface or Globally on the appliance. The ASA appliance supports one Service-Policy per interface and one Globally.

The diagram below illustrates the structure of the Cisco ASA Modular Policy Framework. Keep this structure in mind to help you understand the various configuration examples and scenarios that we will describe later on.


Modular Policy Framework Structure
1
Create a class-map to identify traffic with "match" commands
Class-Map [class name]
(Match traffic commands)
2
Assign the class-map into a Policy-Map and specify actions
Policy-Map [policy name]
Class [class name]
Actions
3
Enable the policy on an Interface or Globally using a Service-Policy
Service-Policy [policy name]
Apply to Interface or Globally

Configuring Class-Maps

As stated above, in this Chapter we will focus only on Layer3/4 Class-Map. This type of class map classifies traffic based on Layer3 or Layer4 attributes, such as IP address, port number, DSCP values etc. The configuration involves two steps: First configure a name for the class-map and then use the "match" command under the class-map configuration mode in order to identify the traffic flow.

ASA(config)# class-map *[class name]* ← **assign a name to the class of traffic**
ASA(config-cmap)# match access-list *[ACL name]* ←**match traffic based on ACL**
ASA(config-cmap)# match port *[**tcp/udp**] [**eq** port_no |* ***range*** *port port]*←**match based on ports**
ASA(config-cmap)# match any ←**match any traffic**
ASA(config-cmap)# match default-inspection-traffic ←**match the default ports for the supported applications. More on this later**
ASA(config-cmap)# match dscp *[value]* ←**match specific dscp value(s) in the IP header. E.g dscp ef means "match expedited forwarding packets" which are usually voice packets.**
ASA(config-cmap)# match precedence *[value]* ←**match specific precedence value(s) in the IP header. Similar with dscp.**
ASA(config-cmap)# match tunnel-group *[tunnel name]*←**match specific site-to-site VPN tunnel or even remote access VPN group**
ASA(config-cmap)# match flow ip destination-address ←**this must be used together with the tunnel-group command above**
ASA(config-cmap)# match rtp *[start port-end port]* ←**match port range of RTP traffic**

Default Class-Map and default-inspection-traffic

By default, an out-of-the-box Cisco ASA appliance has a class-map already configured which matches the default-inspection-traffic. You can view this default class-map in the configuration by using the "**show run class-map**" command.

ASA(config)# show run class-map

```
class-map inspection_default
  match default-inspection-traffic
```

The keyword “**default-inspection-traffic**” is a special name which denotes matching of several default applications and protocols on their default ports, as shown on the table below.

Protocol/Application	**Protocol Type (tcp/udp)**	**Port**
CTIQBE (Computer Telephony Interface)	TCP	2748
DNS	UDP	53
FTP	TCP	21
GTP (GPRS Tunneling Protocol) *requires special license	UDP	2123 3386
H323 H225	TCP	1720
H323 RAS	UDP	1718-1719
HTTP	TCP	80
ICMP	N/A	N/A
ILS (LDAP)	TCP	389
IPSec Pass-Through	UDP	500
MGCP (Media Gateway Control Protocol)	UDP	2427,2727
NetBIOS Name Server	UDP	137,138 (source ports)
PPTP	TCP	1723
RADIUS Accounting	UDP	1646
RSH	TCP	514
RTSP	TCP	554
SIP	TCP/UDP	5060
SCCP (Cisco Skinny)	TCP	2000
SMTP-ESMTP	TCP	25
SNMP	UDP	161,162
SQL*Net	TCP	1521
SUN RPC	UDP	111
TFTP	UDP	69
XDMCP	UDP	177

Most of the applications and protocols shown above are inspected by the ASA in its default configuration. For example, an FTP communication through the ASA between an FTP client and server uses a Control connection on port 21 and a Data connection on port 20. Normally a stateful firewall would not allow such a communication to go through because the initial connection is on port 21 and the return FTP data traffic is on a different port (20). Using the "**default-inspection-traffic**" mechanism described above (together with the "inspect" command under Global policy-map configuration), the Cisco ASA will inspect the FTP traffic in order to allow both the control and the data connection flows to pass through with no problems. The rest of the protocols from the Table above either exhibit similar behavior with FTP or generally require some special "handling", therefore they are inspected by the firewall on the application layer for proper communication. For example, the voice signaling protocol H323 has to be inspected on the application layer in order for the firewall to allow the voice RTP (Real Time Protocol) traffic (which works on random range of UDP ports) to pass through the ASA for a successful VoIP communication.

Configuration Example for Class-Map

Consider a scenario where we want to apply some specific policies for the traffic reaching our company's Web Server from the Internet. Maybe we need to apply a restriction on the maximum number of simultaneous TCP connections allowed to reach our Web Server. Also, we want to prioritize voice traffic having a DSCP value of "ef" (expedited forwarding) that goes through a specific site-to-site IPSec VPN tunnel. We will create two class-maps which will classify the traffic that we described above:

```
ASA(config)# access-list websrv_traffic permit tcp any host 50.50.50.10 eq 80 ← assume our public web server is host 50.50.50.10
ASA(config)# class-map HTTP_To_Web_Server ← create a class-map for the http traffic
ASA(config-cmap)# match access-list websrv_traffic ←match traffic going to web server

ASA(config)# class-map L2L_Voice_Traffic ←create a class-map for the voice lan-to-lan traffic
ASA(config-cmap)# match tunnel-group SITE_B_VPN ←match IPSec tunnel group SITE_B_VPN
ASA(config-cmap)# match dscp ef ←match EF type traffic (i.e voice)
```

Keep in mind the configuration snapshot above because we will refer to it later on when we will describe Policy Maps.

Configuring Policy Maps

After classifying the traffic with a class-map, we need to assign this class-map into a Policy-Map which is responsible to apply some actions (policies) on the selected traffic (i.e traffic that matches a “**match**” statement in the class-map). We will focus only on Layer3/4 Policy Maps.

The security appliance supports one Policy-Map per interface and one Global Policy-Map. Also, each Policy-Map can support multiple Class-Maps and multiple actions on traffic. For instance, in the configuration example shown in the previous section for class-maps, we have configured two class-maps, namely “**HTTP_To_Web_Server**” and “**L2L_Voice_Traffic**“. We can assign both class-maps into a single Policy-Map and apply actions on them.

To configure a Policy-Map, first configure a name for it, then assign a class-map (using the “**class**” command) and then configure actions for the specific class-map.

```
ASA(config)# policy-map [policy name] ← assign a name to the policy map
ASA(config-pmap)# class [class-map name] ←assign a class-map
ASA(config-pmap-c)# [configure actions] ←here configure actions for the specific class-map
ASA(config-pmap-c)# exit
ASA(config-pmap)# class [class-map name] ←assign a second class-map on the same policy
ASA(config-pmap-c)# [configure actions] ←configure actions for the second class-map
```

The available categories of “***actions***” that can be configured on a policy-map are the following:

1. **CSC**: send the traffic to Content Security and Control service module.
2. **IPS**: send the traffic to the Intrusion Prevention System service module.
3. **set connection**: enforce connection limits on traffic.
4. **inspect**: apply protocol inspection services.
5. **police**: apply rate limiting for traffic
6. **priority**: apply priority for voice traffic (Low Latency Queuing-LLQ)

NOTE: The **CSC** (Content Security and Control) and **IPS** (Intrusion Prevention System) mentioned above are add-on card modules that can be inserted into the ASA firewall chassis to provide extra functionality (content inspection, antivirus, antispam, intrusion detection etc).

Now, to get a more complete picture of the usage of both class-maps and policy-maps, let's see some configuration examples below in various scenarios. The six example scenarios below will cover the six available "action" categories that can be applied from a policy-map to a class-map. The six action categories in a policy-map (as listed above) are: **CSC**, **IPS**, **set connection**, **inspect**, **police**, and **priority**.

Configuration Scenario 1: Send traffic to CSC ASA Module for inspection

The CSC module is an SSM card (Security Services Module) that is purchased separately and inserted into the ASA chassis to offer extra functionality such as antivirus, antispam, antispyware etc. The CSC module communicates with the ASA firewall via its backplane.
The CSC module can inspect and filter the following protocols (on their default port):

- HTTP traffic on TCP port 80
- POP3 traffic on TCP port 110
- SMTP traffic on TCP port 25
- FTP traffic on TCP port 21

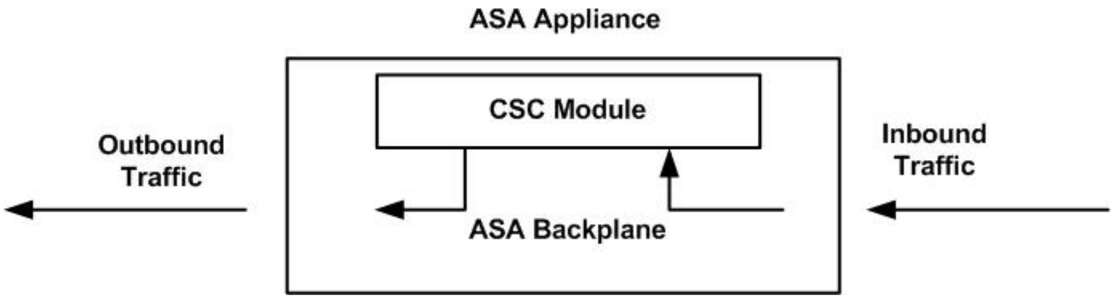
The Cisco ASA appliance can send HTTP, FTP, POP3 and SMTP traffic to the CSC module for inspection and filtering before allowing the traffic to continue to its destination. You can choose to scan traffic for all of these protocols or any combination of them. By default, the ASA does not sent any traffic to the CSC module. You must configure a class-map to identify traffic to be scanned, and then configure a policy-map with the "**csc**" command which will instruct the ASA firewall to send the traffic to CSC module for inspection. Here is how we configure CSC policy:

```
ASA(config)# policy-map [policy name]
ASA(config-pmap)# class [class name] ←first identify traffic to be scanned by CSC
ASA(config-pmap-c)# csc {fail-close | fail-open} ←send traffic to CSC card
```

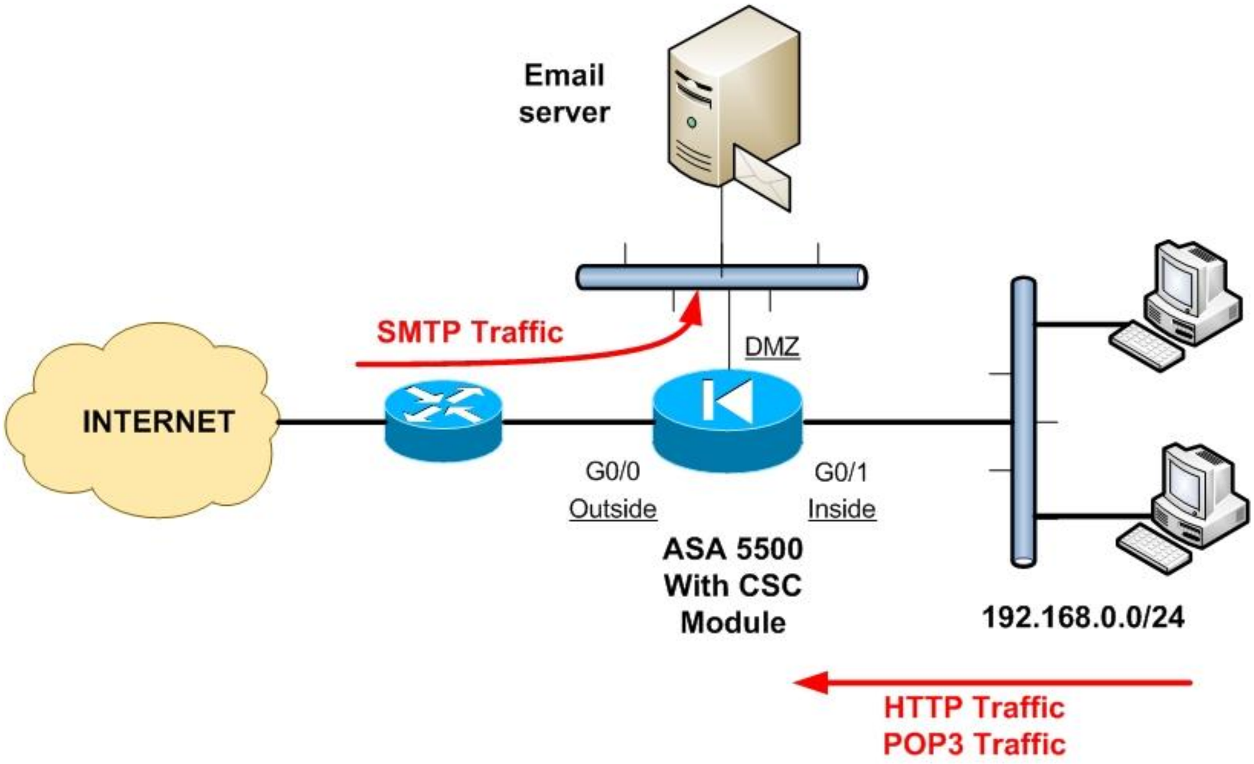
fail-close = if CSC card fails, traffic will be dropped

fail-open = if CSC card fails, traffic will be forwarded

Using the CSC module with the ASA



In our example below we want to scan and inspect HTTP and POP3 traffic from our internal network users towards the Internet, and also scan and inspect SMTP traffic coming from the Internet towards our company's mail server located on DMZ.



Assume that our SMTP mail server listens on IP address 50.50.50.1 (port 25). The policy will be applied globally, which means it will affect ingress traffic on all interfaces.

```
ASA(config)# access-list CSC_traffic permit tcp 192.168.0.0 255.255.255.0 any eq 80
ASA(config)# access-list CSC_traffic permit tcp 192.168.0.0 255.255.255.0 any eq 110
ASA(config)# access-list CSC_traffic permit tcp any host 50.50.50.1 eq 25

ASA(config)# class-map CSC_class ← create a class-map for traffic towards CSC
ASA(config-cmap)# match access-list CSC_traffic ←identify traffic to be inspected
ASA(config-cmap)# exit
ASA(config)# policy-map global_policy ← get into the default global policy
ASA(config-pmap)# class CSC_class ← attach the CSC class-map in global policy
ASA(config-pmap-c)# csc fail-open ←send traffic to CSC
ASA(config-pmap-c)# exit
ASA(config-pmap)# exit
ASA(config)# service-policy global_policy global ←attach the policy globally (this line should be already configured in the default ASA configuration)
```

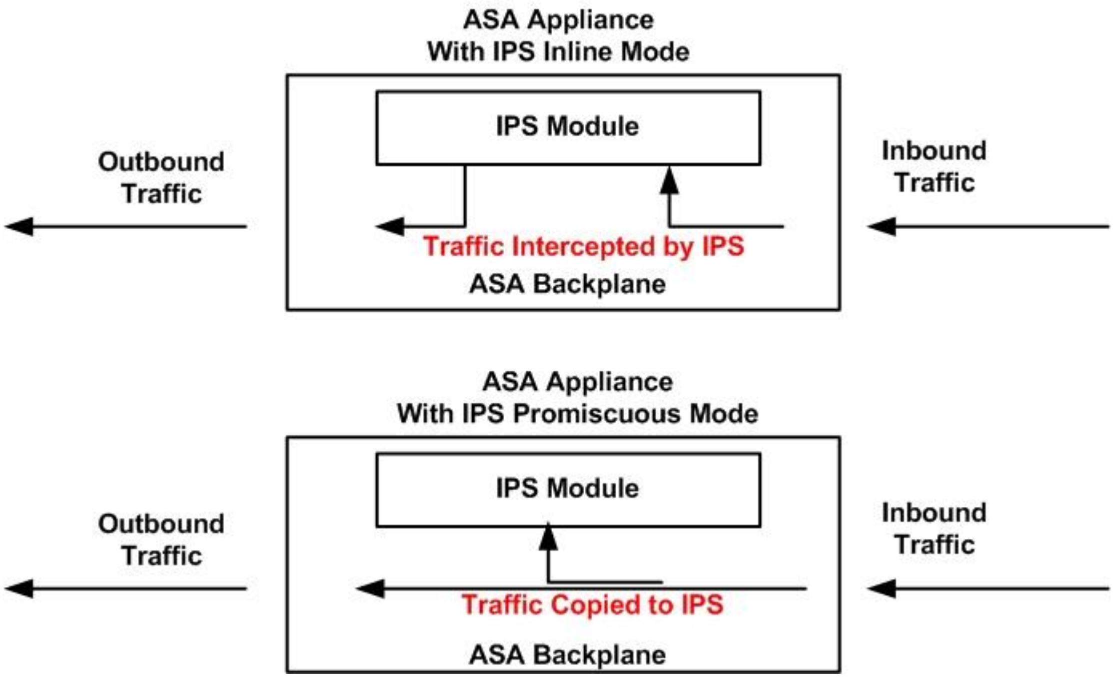
Configuration Scenario 2: Send traffic to IPS Module for inspection

Similarly with a CSC card described above, an Intrusion Prevention System (IPS) module card can also be used in an ASA chassis to provide intrusion detection and prevention functionality. The IPS module is loaded with specialized intrusion detection software which uses "signatures" to identify patterns of malicious traffic in order to block it. Only one module can be used in an ASA though, either a CSC or an IPS module. The IPS module (also called AIP-SSM) can operate in two modes:

- **IPS Inline Mode**: In inline mode, the IPS sits in the traffic path and therefore the traffic is fully intercepted and inspected by the IPS before being sent back to the ASA firewall. The traffic that passes through the IPS is the traffic that matches a class-map configured with the "**ips**" command. In inline mode, the IPS is capable to block attacks by itself..
- **IPS Promiscuous Mode**: In promiscuous mode, the IPS does not intercept traffic that passes through the ASA. Instead, the ASA firewall sends a copy of each packet to the IPS for

inspection. If the packet is identified as malicious by the IPS, it issues an alarm or instructs the ASA firewall (using the "shun" command) to block the traffic. In this mode the IPS does not block attacks by itself.

Using the IPS module with the ASA



By default, the ASA does not sent any traffic to the IPS module. You must configure a class-map to identify traffic to be inspected by IPS, and then configure a policy-map with the "**ips**" command which will instruct the ASA firewall to send the traffic to IPS module for inspection. Here is how we configure the IPS policy:

```
ASA(config)# policy-map [policy name]
ASA(config-pmap)# class [class name] ←first identify traffic to be inspected by IPS
ASA(config-pmap-c)# ips {inline | promiscuous} {fail-close | fail-open} ←send traffic to IPS
```

inline = the IPS will be working in inline mode

promiscuous = the IPS will be working in promiscuous mode

fail-close = if IPS card fails, traffic will be dropped

fail-open = if IPS card fails, traffic will be forwarded

Let’s see a more complete example below. Assume that we have a DMZ zone with public servers on a class C subnet 50.50.50.0/24. We want all traffic coming from Internet towards our DMZ servers to be inspected by the IPS in inline mode.

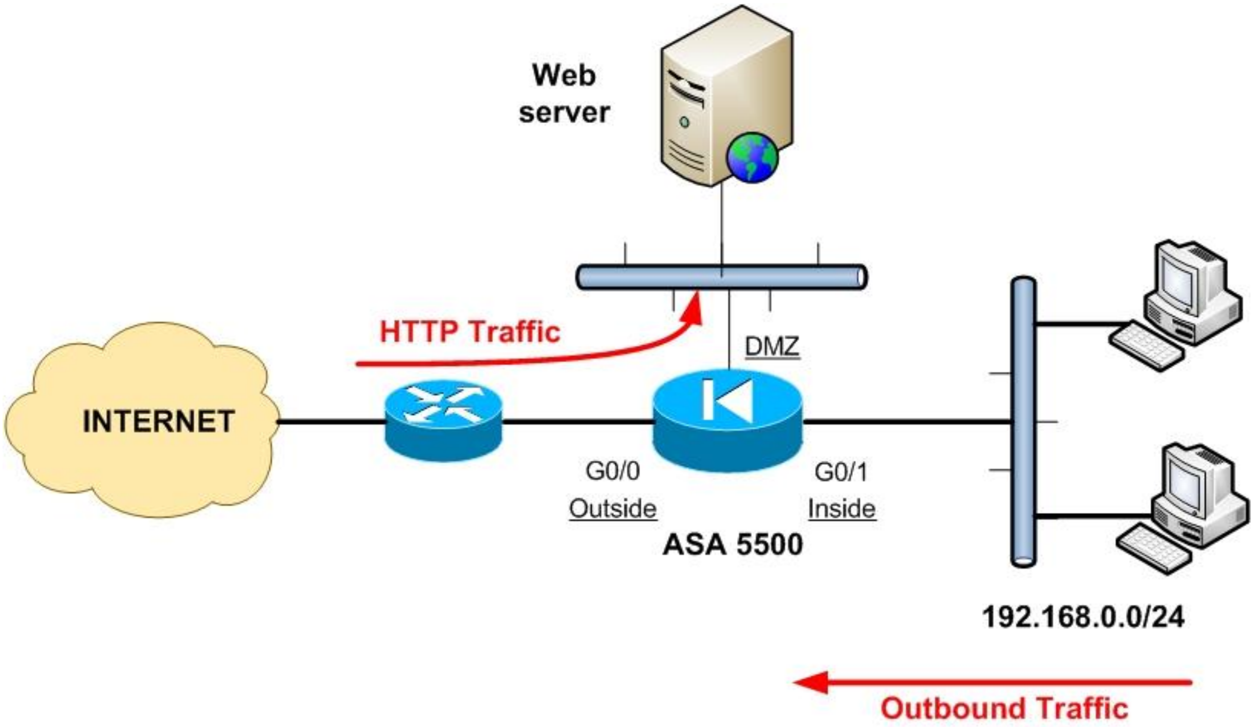
```
ASA(config)# access-list DMZ_traffic permit ip any 50.50.50.0 255.255.255.0

ASA(config)# class-map IPS_class ← create a class-map for traffic towards IPS
ASA(config-cmap)# match access-list DMZ_traffic ←identify traffic to be inspected
ASA(config-cmap)# exit
ASA(config)# policy-map outside_ips_policy ← create a policy-map for IPS
ASA(config-pmap)# class IPS_class ← attach the IPS class-map in the IPS policy
ASA(config-pmap-c)# ips inline fail-open ←send traffic to IPS in inline mode
ASA(config-pmap-c)# exit
ASA(config-pmap)# exit
ASA(config)# service-policy outside_ips_policy interface outside ←attach the policy on the outside interface
```

Configuration Scenario 3: Set Connection Limits Policy

The “**set connection**” command used under a policy-map configuration is used to enforce connection limits for specific traffic flows. When a connection matches the associated match criteria in the class-map, the ASA appliance sets the specified connection limits to the traffic. You can use the “set connection” command to configure the following:

- **conn-max**: Maximum number of simultaneous connections allowed. Can help to protect against Denial of Service attacks.
- **per-client-max**: Maximum number of connections allowed per client. Can help to restrict internal users from opening excessive connections (e.g when using torrent or peer-to-peer)
- **embryonic-conn-max**: Maximum numbers of TCP “half-open” (embryonic) connections allowed. Protects against “SYN” attacks.
- **per**-client-embryonic-max: Maximum number of TCP embryonic allowed per client.

Let’s see an example scenario below.



We want to apply connection limit policies for HTTP inbound traffic (from Internet to DMZ Web Server) and also for users' outbound traffic (on a per user basis). Assume that our Web Server listens on public IP address 50.50.50.1.

```
ASA(config)# access-list HTTP_traffic permit tcp any host 50.50.50.1 eq 80
ASA(config)# access-list outbound_traffic permit ip 192.168.0.0 255.255.255.0 any

ASA(config)# class-map Web_SRV_Class ← create a class-map for DMZ Web Server
ASA(config-cmap)# match access-list HTTP_traffic ←identify HTTP traffic to web srv
ASA(config-cmap)# exit

ASA(config)# class-map Outbound_Class ← create a class-map for Outbound traffic
ASA(config-cmap)# match access-list outbound_traffic ←identify outbound traffic
ASA(config-cmap)# exit
ASA(config)# policy-map Web_SRV_policy ← create a policy-map for Web Server
ASA(config-pmap)# class Web_SRV_Class ← attach the class-map on the policy
ASA(config-pmap-c)# set connection conn-max 3000 ←limit max connections to web srv
ASA(config-pmap-c)# set connection per-client-max 100 ←limit max per client connections to web server to 100
ASA(config-pmap-c)# set connection embryonic-conn-max 1500 ←limit max half-open connections to web server to 1500
ASA(config-pmap-c)# exit
ASA(config-pmap)# exit

ASA(config)# policy-map outbound_policy ← create a policy-map for outbound traffic
ASA(config-pmap)# class Outbound_Class ← attach the class-map on the policy
ASA(config-pmap-c)# set connection per-client-max 70 ←limit max simultaneous connections for each internal user to 70
ASA(config-pmap-c)# exit
ASA(config-pmap)# exit

ASA(config)# service-policy Web_SRV_policy interface outside ←attach the Web server policy on the outside interface
ASA(config)# service-policy outbound_policy interface inside ←attach the outbound policy on the inside interface
```

Configuration Scenario 4: Traffic Inspection Policy

An out-of-the-box Cisco ASA firewall has a global default inspection policy which applies inspection policies on several applications and protocols which are matched by the default class map. If you view the running configuration of the ASA (using the "show run" command) you will notice the following default configuration commands:

```
class-map inspection_default
 match default-inspection-traffic
!
policy-map type inspect dns preset_dns_map
 parameters
  message-length maximum 512

policy-map global_policy
 class inspection_default
  inspect dns preset_dns_map
  inspect h323 h225
  inspect h323 ras
  inspect rsh
  inspect rtsp
  inspect sqlnet
  inspect skinny
  inspect sunrpc
  inspect xdmcp
  inspect sip
  inspect netbios
  inspect tftp
  inspect ils
  inspect ftp
  inspect http
!
service-policy global_policy global
```

From the default configuration shown above, you can observe that there is a default class-map (***class-map inspection_default***) and a default policy-map (***policy-map global_policy***). The default policy is applied globally on the appliance (*service-policy* ***global_policy*** *global*).

Notice that we use the "inspect" command to apply application layer inspection on several protocols. We can add or remove protocols from the global policy accordingly. You can go under the ***policy-map global_policy*** > ***class inspection_default*** and type "**inspect ?**" to see which other protocols are supported for inspection. Then you can add more protocols for inspection as needed.

The "inspect" command for each protocol helps the security appliance to do the following:

- Look for common security issues in the application layer and prevent them.
- Look for additional connections that need to be opened (e.g for FTP or voice traffic) and open those connections as well.
- Look for embedded addressing information inside packets that will be translated with NAT.

Configuration Scenario 5: Apply Bandwidth Policy to traffic using the Police Command

The "**police**" command under a policy-map configuration is used to apply rate-limiting to traffic flow. You need to specify the direction of the traffic to be policed, the rate limit bandwidth (in bps) and optionally the burst size and the actions to be taken for conforming or non-conforming burst traffic.

The "police" mechanism is configured as below:

```
ASA(config)# policy-map [policy name]
ASA(config-pmap)# class [class name] ←first identify traffic to be policed
ASA(config-pmap-c)# police {input|output} conform-rate-in-bps [burst size in bytes] conform-action {drop|transmit} exceed-action {drop|transmit}
```

The "input" keyword applies traffic limiting to packets entering an interface and the "output" keyword applies traffic limiting to packets leaving an interface. The burst size indicates the maximum size in bytes of an instantaneous burst of traffic allowed before the traffic is capped to get it back to the policing rate. A formula to calculate a good maximum burst size, according to the maximum rate limit applied, is the following:

Burst Size = (conform rate in bps)/8 * 1.5

Assume that we want to apply rate limiting to a specific IPSec remote access user group (with name "Remote_VPN") as following:

- Maximum allowed bandwidth of 512kbps
- Burst Size = (512000/8)*1.5 = 96000 bytes

Let's see the configuration snapshot below:

```
ASA(config)# class-map VPN_Users_Class ← create a class-map for VPN remote users
ASA(config-cmap)# match tunnel-group Remote_VPN ←identify the VPN tunnel group
ASA(config-cmap)# exit
ASA(config)# policy-map VPN_policy ← create a policy-map for VPN remote access
ASA(config-pmap)# class VPN_Users_Class ← attach the class-map on the policy
ASA(config-pmap-c)# police input 512000 96000 conform-action transmit exceed-action drop ←limit the traffic to 512kbps and 96000 bytes of burst size
ASA(config-pmap-c)# police output 512000 96000 conform-action transmit exceed-action drop ←do the same for outgoing traffic
ASA(config-pmap-c)# exit
ASA(config-pmap)# exit

ASA(config)# service-policy VPN_policy interface outside ←attach the VPN policy on the outside interface
```

Configuration Scenario 6: Setting Prioritization for traffic

The last configuration example that we will see here has to do with priority and queuing. We can use the "**priority**" command under the policy-map configuration to enable Low Latency Queuing for traffic that is delay-sensitive (mainly voice). Together with the "**priority**" command we must also use the "**priority-queue**" command in order to enable the priority **queue** on the interface on which we want to apply high priority for traffic. Each interface of the security appliance has two queues: A priority queue which is used to transmit delay-sensitive traffic and a default queue which transmits all other traffic. Priority queuing is applied ONLY on egress traffic (packets that exit from an interface).

The "**priority**" mechanism is configured as following:

```
ASA(config)# policy-map [policy name]
ASA(config-pmap)# class [class name] ←first identify traffic to apply priority on
ASA(config-pmap-c)# priority
ASA(config-pmap-c)# exit
ASA(config-pmap)# exit
ASA(config)# priority-queue logical_if_name ←enable the priority queue on an interface
```

In the following example we will apply high priority for voice traffic that passes through a specific Lan-to-Lan IPSec tunnel between two sites.

```
ASA(config)# class-map Voice_L2L_Class ← create a class-map for voice VPN traffic
ASA(config-cmap)# match tunnel-group L2L_VPN ←identify the IPSec VPN tunnel
ASA(config-cmap)# match dscp ef ←match the expedited forwarding "ef" voice traffic
ASA(config-cmap)# exit

ASA(config)# policy-map voice_policy ← create a policy-map for VPN voice traffic
ASA(config-pmap)# class Voice_L2L_Class ← attach the class-map on the policy-map
ASA(config-pmap-c)# priority ← apply high priority to voice traffic
ASA(config-pmap-c)# exit
ASA(config-pmap)# exit
ASA(config)# priority-queue outside ←enable the priority queue on the outside interface
```

Applying The Policy Using a Service-Policy

So far we have seen the two (out of three) components of a Modular Policy Framework (MPF) configuration. That is, we have described **Class-Maps** and **Policy-Maps**. The third and last component of an MPF is a **service-police**. A service-policy is used to attach the policy-map either Globally or on a specific interface.

- If the policy-map is applied globally, actions are applied to traffic in the ingress direction only.
- If the policy-map is applied to a specific interface, actions are applied to traffic bidirectionally.
- Exception to the above is the Priority Queuing policy which is always applied to traffic on the egress direction.

The service-policy is used as following:

ASA(config)# service-policy {*policy-map name}* **{global | interface** *if_name***}**

To verify that your policies are being enforced use the "**show service-policy**" command.

CHAPTER 11:
CONFIGURING ANYCONNECT WEBVPN

In this Chapter we will describe the newest VPN functionality supported by the Cisco ASA, the **AnyConnect WebVPN,** which uses SSL and a special Java client software to offer tunnel mode remote access VPN for users. Before moving on to the details of AnyConnect WebVPN, let's first refresh our memory about the VPN technologies supported by Cisco ASA firewalls.

OVERVIEW OF CISCO ASA VPN TECHNOLOGIES

Cisco supports several types of VPN implementations on the ASA but they are generally categorized as either "**IPSec Based VPNs**" or "**SSL Based VPNs**". The first category uses the IPSec protocol for secure communications while the second category uses SSL. SSL Based VPNs are also called **WebVPN** in Cisco terminology. The two general VPN categories supported by Cisco ASA are further divided into the following VPN technologies.

- **IPSec Based VPNs**:
  - ***Lan-to-Lan IPSec VPN***: Used to connect remote LAN networks over unsecure media (e.g Internet). It runs between ASA-to-ASA or ASA-to-Cisco Router.
  - ***Remote Access with IPSec VPN Client***: A VPN client software is installed on user's PC to provide remote access to the central network. Uses the IPSec protocol and provides full network connectivity to the remote user. The users use their applications at the central site as they normally would without a VPN in place.

- **SSL Based VPNs (WebVPN)**:
  - ***Clientless Mode WebVPN:*** This is the first implementation of SSL WebVPN supported from ASA version 7.0 and later. It lets users establish a secure remote access VPN tunnel using just a Web browser. There is no need for a software or hardware VPN client. However, only limited applications can be accessed remotely.
  - ***AnyConnect WebVPN:*** A special Java based client is installed on the user's computer providing an SSL secure tunnel to the central site. Provides full network connectivity (similar with IPSec remote access client). All applications at the central site can be accessed remotely.

COMPARISON BETWEEN WEBVPN TECHNOLOGIES

In this Chapter we will focus only on AnyConnect WebVPN. I decided not to bother with the Clientless WebVPN because I believe that the benefits of using AnyConnect instead of Clientless are much more. To justify what I'm saying, let's see the differences between the two WebVPN modes and I'm sure you will understand why I focus only on AnyConnect!

Clientless WebVPN does not require any VPN client to be installed on user's computer. It uses a normal web browser. By pointing the browser to **https://[outside address of ASA]** the user authenticates with the firewall and gets access to a Web Portal. Through this Web Portal, the user can then access a limited number of internal applications. Specifically, only internal Web applications (HTTP, HTTPs), email servers (POP3, SMTP, IMAP), Windows file shares and a small number of TCP legacy applications (e.g Telnet) can be accessed. That is, there is no full network connectivity with Clientless WebVPN.

AnyConnect WebVPN, on the other hand, provides FULL network connectivity to the remote user. The ASA firewall, working as AnyConnect WebVPN server, assigns an IP address to the remote user and attaches the user to the network. Thus, all IP protocols and applications function across the SSL VPN tunnel without any problems. For example, a remote user, after successfully authenticated with AnyConnect VPN, can open a Remote Desktop connection and access a Windows Terminal Server inside the central network. Although a special Java-based client is required to be installed on the user's desktop, this client can be supplied dynamically to the user from the ASA. The user can connect with a browser to the ASA firewall and download the Java client on demand. The Java client can remain installed or even get removed from the user's desktop when disconnected from the ASA appliance. This Java client is small in size (around 3MB) and is stored on the ASA flash memory.

ANYCONNECT WEBVPN OVERVIEW

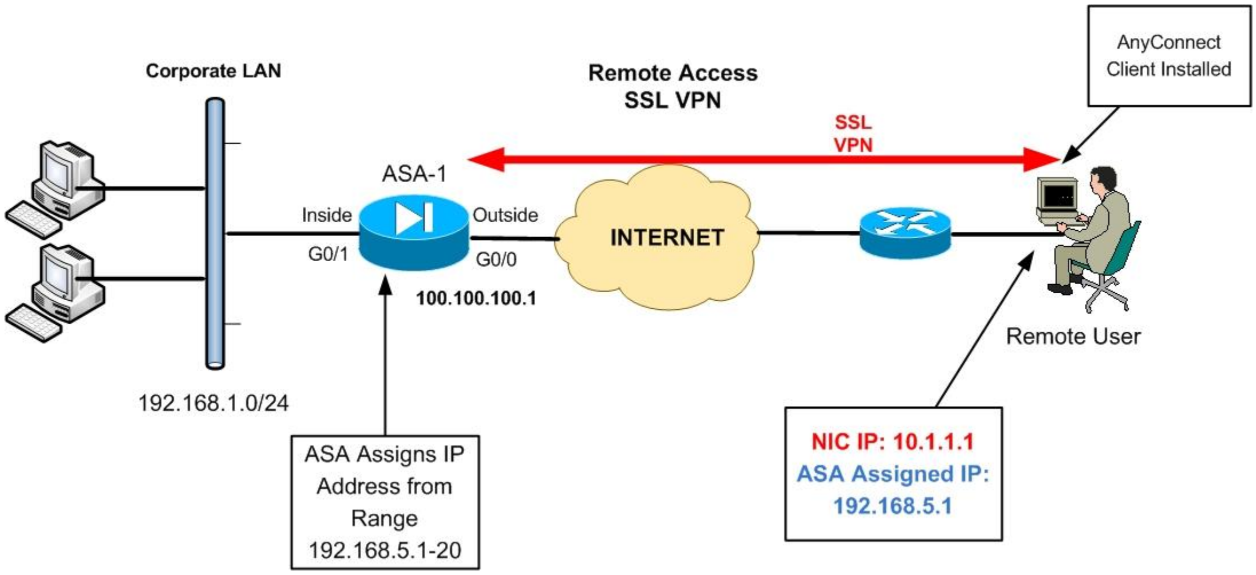
The AnyConnect WebVPN client protects traffic at the network layer and above (tunnel-mode). It provides the same remote access functionality as the Cisco IPSec VPN client. There are two versions of the tunnel-mode WebVPN client as shown below:

WebVPN Client	Operating System Supported	ASA version
SSL VPN Client (SVC)	Windows 2000 and XP	7.0-7.2
AnyConnect Client	Windows 2000, XP, VISTA, MAC OS X, Linux	8.0+

In the older ASA versions 7.0 up to 7.2, the WebVPN client was called SVC (SSL VPN Client). From ASA version 8.0 and later, the client is called AnyConnect WebVPN client. Although we will focus only on the AnyConnect client, the configuration for both client versions (SVC and AnyConenct) is the same on the ASA.

Overview of AnyConnect VPN operation:

The diagram below shows a network topology with ASA and a remote user with AnyConnect VPN.



From the diagram above, the ASA firewall is configured as AnyConnect WebVPN server. A remote user has access to the Internet and has an IP address on his/her laptop interface card of 10.1.1.1 (NIC IP). The user can also be behind a router doing NAT/PAT and have his private IP address translated to a public IP by the NAT router. When the remote user connects and successfully authenticates to the ASA with the AnyConnect client, the ASA will assign an internal IP address to the user from a preconfigured IP address range (in our example above, this address range is 192.168.5.1 up to 192.168.5.20). From the diagram above, the ASA assigns IP 192.168.5.1 to the remote user. This means that the remote user is virtually attached to the corporate LAN behind the ASA firewall.

The operation overview described above assumes that the AnyConnect client is already installed on the user's laptop. Let's see below the available options how to initially install the AnyConnect client.

There are two Initial Installation options for AnyConnect client:

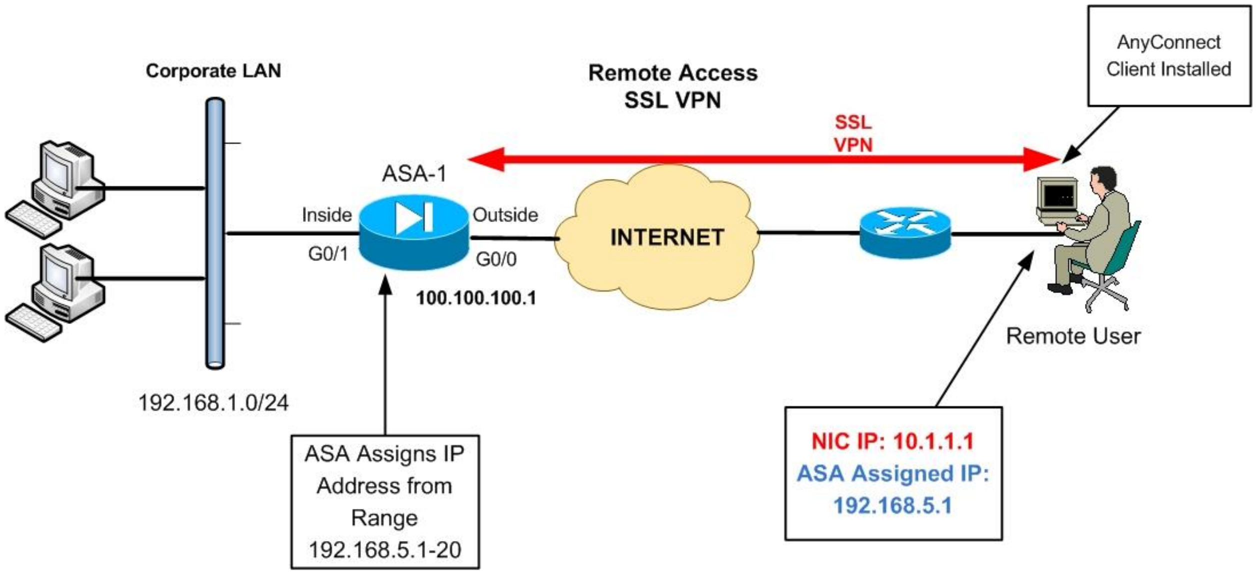
- Using clientless WebVPN portal.
- Manual installation by the user

Using the clientless Web portal, the user first connects and authenticates to the ASA with a secure web browser and the Java Anyconnect client is automatically downloaded and installed on the user's computer (the user can also click the "AnyConnect" Tab on the WebVPN portal to download the client). This means that the Java client (**.pkg extension**) is already stored on the ASA flash memory by the administrator (you need to download it from Cisco site). This is the preferred method in my opinion because it automates the distribution of the client to the remote users.

With the manual installation method, the network administrator must download the appropriate Java client (Microsoft MSI package installer or one of the other OS versions) from Cisco site and provide the file to the users for manual installation on their laptop. With this method, the user does not need to log in via clientless mode to start the SSL VPN tunnel. Instead, the users can start up the AnyConnect client manually from their desktop and provide their authentication credentials.

ANYCONNECT CONFIGURATION STEPS

We will focus on the automatic Anyconnect installation option, i.e the AnyConnect client is located on the ASA flash memory and is downloaded by the remote users. The diagram below will be used to describe the configuration:



STEP1:

Transfer the PKG file to flash on the ASA. First you need to download one of the **.pkg** files from Cisco website. An example Windows client file has the format "**anyconnect-win-x.x.xxxx-k9.pkg**".

To copy the PKG file to ASA flash:

ASA# copy {tftp|ftp|scp}://*[ip address]/anyconnect-win-x.x.xxxx-k9.pkg* **disk0:**

Assume we have downloaded the Anyconnect client file on our computer with IP address 192.168.1.1. We will use a TFTP server on our PC to transfer the file to ASA.

ASA# copy tftp://192.168.1.1/anyconnect-win-2.3.2016-k9.pkg disk0:

```
Address or name of remote host [192.168.1.1]?
Source filename [anyconnect-win-2.3.2016-k9.pkg]?
Destination filename [anyconnect-win-2.3.2016-k9.pkg]?
Accessing tftp://192.168.1.1/anyconnect-win-2.3.2016-k9.pkg...!!!!!!
```

STEP2:

Identify the PKG image file on flash by telling the ASA where the image file is located. Also, enable the webvpn Anyconnect service on the outside ASA interface.

```
ASA# configure terminal
ASA(config)# webvpn
ASA(config-webvpn)# svc image disk0:/anyconnect-win-2.3.2016-k9.pkg 1
ASA(config-webvpn)# enable outside
ASA(config-webvpn)# svc enable
```

Note: The number 1 at the end of the package file is the file order. It is used when you have more than one images stored on the ASA flash (e.g Anyconnect client images for Windows and MAC).

STEP3:

Exempt the SSL WebVPN traffic from Access List checks on the outside interface. By default, WebVPN traffic is not exempted from Access List checks after terminated on the outside interface; once the traffic is decrypted, it is checked by the inbound ACL applied on outside interface. You must either include **permit** statements for the decrypted traffic in the ACL, or use the "**sysopt connection permit-vpn**".

```
ASA(config)# sysopt connection permit-vpn
```

STEP4:

This step is optional but it is really helpful. All SSL VPN communication between remote users and ASA works with secure HTTPs (port 443). This means that users have to use "**https://[ASA public IP]"** on their browsers. Since most users will forget to use "https://", you can set up port redirection which means that if the user connects to port 80 ("http://"), the ASA will automatically redirect the browser to port 443.

```
ASA(config)# http redirect outside 80
```

STEP5:

Create an IP address pool from which the ASA will assign addresses to remote users. From the diagram above we see that after the remote user gets authenticated, the ASA assigns an IP address to the remote user from a predefined pool 192.168.5.1 up to 192.168.5.20.

```
ASA(config)# ip local pool VPNpool 192.168.5.1-192.168.5.20 mask 255.255.255.0
```

STEP6:

Create a NAT exemption for traffic between the corporate LAN network behind the ASA (192.168.1.0/24) and the remote user's address pool (VPNpool). We do this exemption because the encrypted traffic must not go through a NAT operation. This step is of course required only of we do NAT on the ASA.

```
ASA(config)# access-list NONAT extended permit ip 192.168.1.0 255.255.255.0 192.168.5.0 255.255.255.0
ASA(config)# nat (inside) 0 access-list NONAT
ASA(config)# nat (inside) 1 0.0.0.0 0.0.0.0
ASA(config)# global (outside) 1 interface ← We assume that we do PAT on the outside interface
```

STEP7:

Create a Group Policy for the AnyConnect WebVPN users. The Group Policy allows you to separate different remote access users into groups with different attributes. The Group Policy attributes that can be configured include DNS server addresses, split-tunneling settings, how the client will be downloaded (automatically or after prompting the user), if the client software will remain permanently on the user's computer etc.

The command format is as following:

```
 ASA(config)# group-policy "policy name" internal
ASA(config)# group-policy "policy name" attributes
ASA(config-group-policy)# vpn-tunnel-protocol {[svc] [webvpn][ipsec] [l2tp-ipsec]}
ASA(config-group-policy)# webvpn
ASA(config-group-webvpn)# svc keep-installer {installed | none}
ASA(config-group-webvpn)# svc ask {none | enable [default {webvpn | svc} timeout value]}
```

Let's clarify some of the Group Policy commands shown above:

svc keep-installer {installed | none} ← "installed" means that the client remains installed permanently on the user's computer even after disconnection. The default is that the client gets uninstalled after the user disconnects from the Anyconnect session.

svc ask {none | enable [default {webvpn | svc} timeout *value*]} ← This command has to do with how AnyConnect client will be downloaded to user's computer.

- svc ask none default webvpn ← The ASA immediately displays the WebPortal. This is the default configuration.
- svc ask none default svc ← Download the AnyConnect client automatically.
- svc ask enable default svc timeout 20 ← The user will get a prompt to install the AnyConnect client. If nothing is done within 20 seconds, the client will be downloaded and installed automatically.

Example:

```
ASA(config)# group-policy Anyconnect-Policy internal
ASA(config)# group-policy Anyconnect-Policy attributes
ASA(config-group-policy)# vpn-tunnel-protocol svc webvpn
ASA(config-group-policy)# dns-server value 192.168.1.15
ASA(config-group-policy)# webvpn
ASA(config-group-webvpn)# svc keep-installer installed
ASA(config-group-webvpn)# svc ask enable default svc timeout 20
```

STEP8:

Create a Tunnel Group. The tunnel group must incorporate the Group Policy configured above. It also binds the Group Policy with the IP address pool that we have already configured for remote users.

The command format is as following:

```
 ASA(config)# tunnel-group “tunnel  name” type remote-access
ASA(config)# tunnel-group “tunnel  name” general-attributes
ASA(config-tunnel-general)# default-group-policy “group policy name” ←Assign the Group Policy configured in Step7 above.
ASA(config-tunnel-general)# address-pool “IP Pool for VPN” ← Assign the IP address pool configured in Step5 above.
ASA(config-tunnel-general)# exit
ASA(config)# tunnel-group “tunnel  name” webvpn-attributes
ASA(config-tunnel-webvpn)# group-alias “group_name_alias” enable ← Create an alias name for the tunnel group which will be listed on the log on screen of the Anyconnect client.
ASA(config-tunnel-webvpn)# exit
ASA(config)# webvpn
ASA(config-webvpn)# tunnel-group-list enable ← Enable the listing of the alias name on the log on screen of the AnyConnect client.
```

Example:

```
ASA(config)# tunnel-group telecommuters  type remote-access
ASA(config)# tunnel-group telecommuters general-attributes
ASA(config-tunnel-general)# default-group-policy Anyconnect-Policy
ASA(config-tunnel-general)# address-pool VPNpool
ASA(config-tunnel-general)# exit
ASA(config)# tunnel-group telecommuters webvpn-attributes
ASA(config-tunnel-webvpn)# group-alias sslgroup_users enable
ASA(config-tunnel-webvpn)# exit
ASA(config)# webvpn
ASA(config-webvpn)# tunnel-group-list enable
```

STEP9:

Create a local user on ASA which will be used for AnyConnect authentication.

```
ASA(config)# username ssluser1 password secretpass
ASA(config)# username ssluser1 attributes
ASA(config-username)# service-type remote-access
```

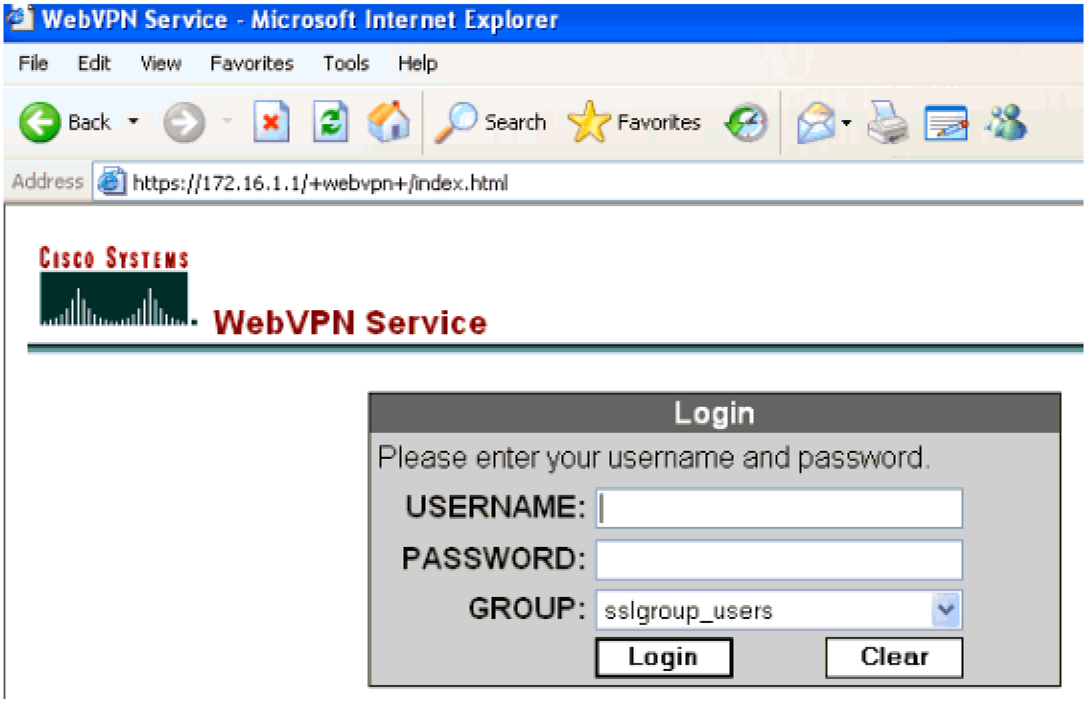
Complete Configuration of AnyConnect WebVPN:

```
ASA# configure terminal
ASA(config)# webvpn
ASA(config-webvpn)# svc image disk0:/anyconnect-win-2.3.2016-k9.pkg 1
ASA(config-webvpn)# enable outside
ASA(config-webvpn)# svc enable
ASA(config-webvpn)# exit
ASA(config)# sysopt connection permit-vpn
ASA(config)# http redirect outside 80
ASA(config)# ip local pool VPNpool 192.168.5.1-192.168.5.20 mask 255.255.255.0
ASA(config)# access-list NONAT extended permit ip 192.168.1.0 255.255.255.0 192.168.5.0
255.255.255.0
ASA(config)# nat (inside) 0 access-list NONAT
ASA(config)# nat (inside) 1 0.0.0.0 0.0.0.0
ASA(config)# global (outside) 1 interface
ASA(config)# group-policy Anyconnect-Policy internal
ASA(config)# group-policy Anyconnect-Policy attributes
ASA(config-group-policy)# vpn-tunnel-protocol svc webvpn
ASA(config-group-policy)# dns-server value 192.168.1.15
ASA(config-group-policy)# webvpn
ASA(config-group-webvpn)# svc keep-installer installed
ASA(config-group-webvpn)# svc ask enable default svc timeout 20
ASA(config-group-webvpn)# exit
ASA(config-group-policy)# exit
ASA(config)# tunnel-group telecommuters  type remote-access
ASA(config)# tunnel-group telecommuters general-attributes
ASA(config-tunnel-general)# default-group-policy Anyconnect-Policy
ASA(config-tunnel-general)# address-pool VPNpool
ASA(config-tunnel-general)# exit
ASA(config)# tunnel-group telecommuters webvpn-attributes
ASA(config-tunnel-webvpn)# group-alias sslgroup_users enable
ASA(config-tunnel-webvpn)# exit
ASA(config)# webvpn
ASA(config-webvpn)# tunnel-group-list enable
ASA(config-webvpn)# exit
ASA(config)# username ssluser1 password secretpass
ASA(config)# username ssluser1 attributes
ASA(config-username)# service-type remote-access
ASA(config)# wr mem
```

To establish an AnyConnect WebVPN

1. Connect to ASA on its public outside address: **https://[outside ASA Address]**

You will get the following screen:


WebVPN Service - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Cisco Systems
Login
Clear

Address https://172.16.1.1/+webvpn+/index.html

WebVPN Service

Login

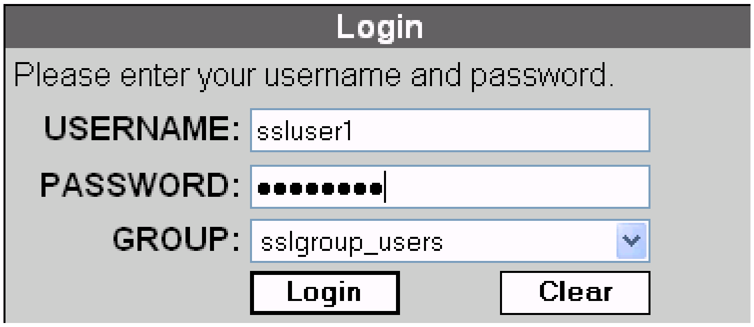
Please enter your username and password.

USERNAME:

PASSWORD:

GROUP: sslgroup_users

2. Enter your username and password (ssluser1). Also, choose your respective group from the drop down list as shown. Note that the group name in the drop down is the group-alias name configured in Step8 (**sslgroup_users**).



Login

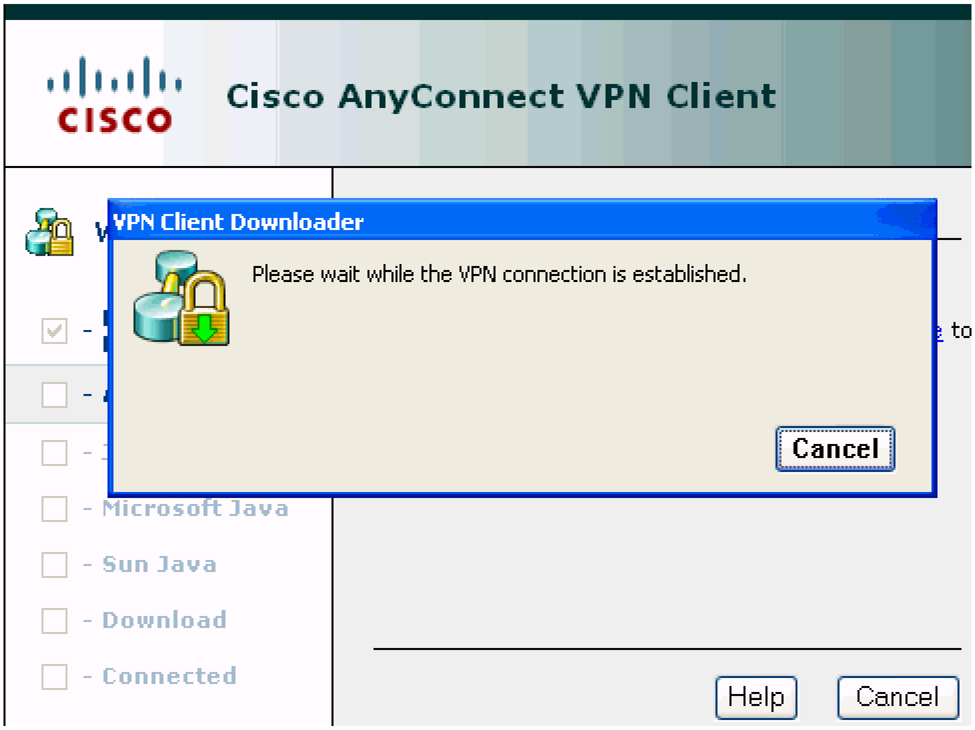
Please enter your username and password.

USERNAME: ssluser1

PASSWORD: ●●●●●●●●

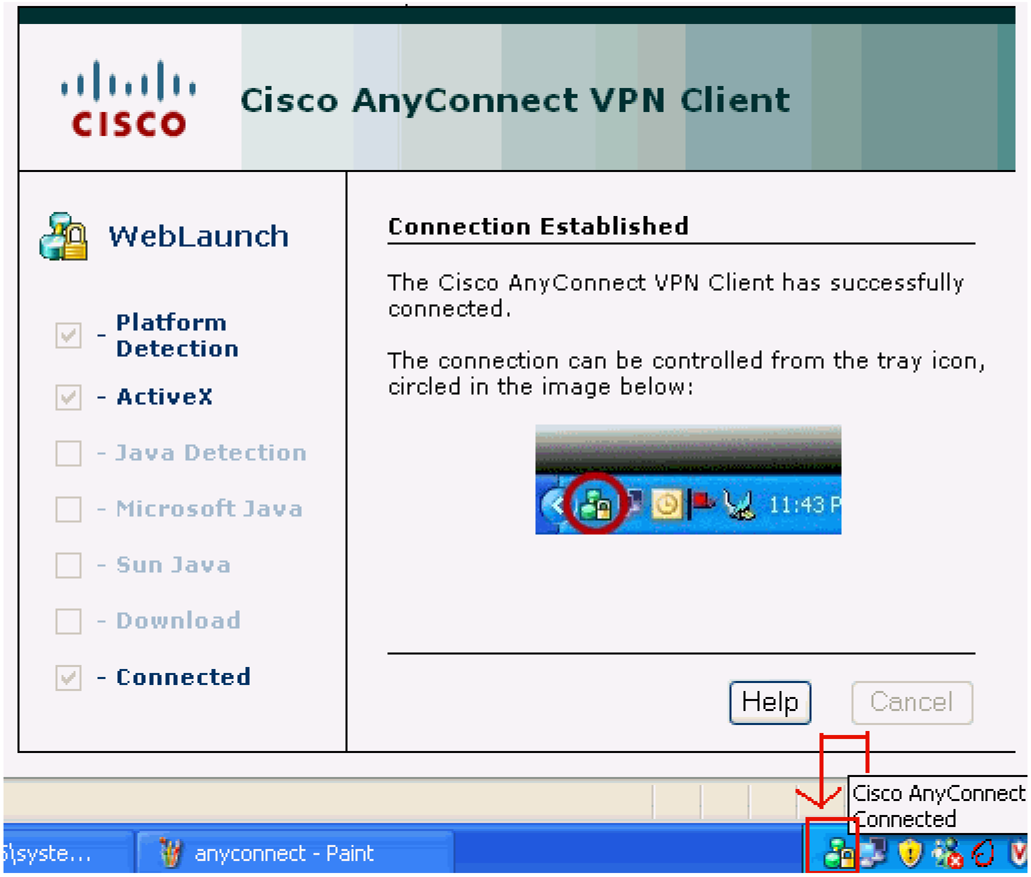
GROUP: sslgroup_users

3. This following window appears before the SSL VPN connection is established.



Please wait while the VPN connection is established.

4. ActiveX software must be installed in your computer before you download the Anyconnect client. You receive the following window once the connection is established.



Cisco AnyConnect VPN Client

WebLaunch

☑ - Platform Detection

☑ - ActiveX

□ - Java Detection

□ - Microsoft Java

□ - Sun Java

□ - Download

☑ - Connected

Connection Established

The Cisco AnyConnect VPN Client has successfully connected.

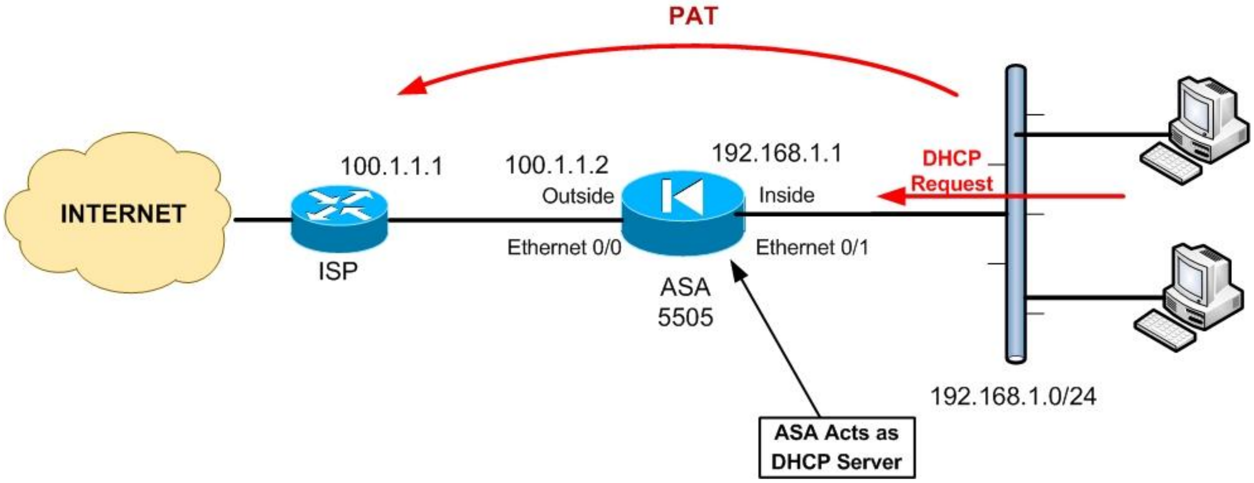
The connection can be controlled from the tray icon, circled in the image below:

CHAPTER 12: CONFIGURATION EXAMPLES

So far we covered the Fundamentals you need to implement a Cisco ASA Firewall in the most common network scenarios.
In this Chapter we will provide real world complete configuration examples of Cisco ASA Firewalls. These configurations will bind together all the example pieces we described before, in order to give you a complete picture of an ASA Configuration in different network topologies.

CONFIGURATION EXAMPLE 1: ASA 5505 BASIC INTERNET ACCESS WITH DHCP

The ASA 5505 (the smallest ASA model) is ideal for small businesses or small branch offices with approximately 50 internal users (recommended maximum). This model comes with 8 port 10/100 switch, with port Ethernet0/0 used for the Public/Outside zone and ports Ethernet0/1 up to 0/7 for the Inside zone. The difference of this model compared with the rest ASA models is that its network ports are pure Layer 2 switch ports. This means you cannot configure IP addresses directly on the physical interfaces. Instead, you have to assign the interface port in a VLAN, and then configure all Firewall Interface parameters using the **interface VLAN** command.
In this scenario the 5505 is used for basic internet access using PAT, with a static Public IP address on the outside (100.1.1.2). The Firewall will act also as a DHCP server for assigning IP addresses to inside hosts.



The complete configuration follows below. See the Blue Color comments for clarifications.

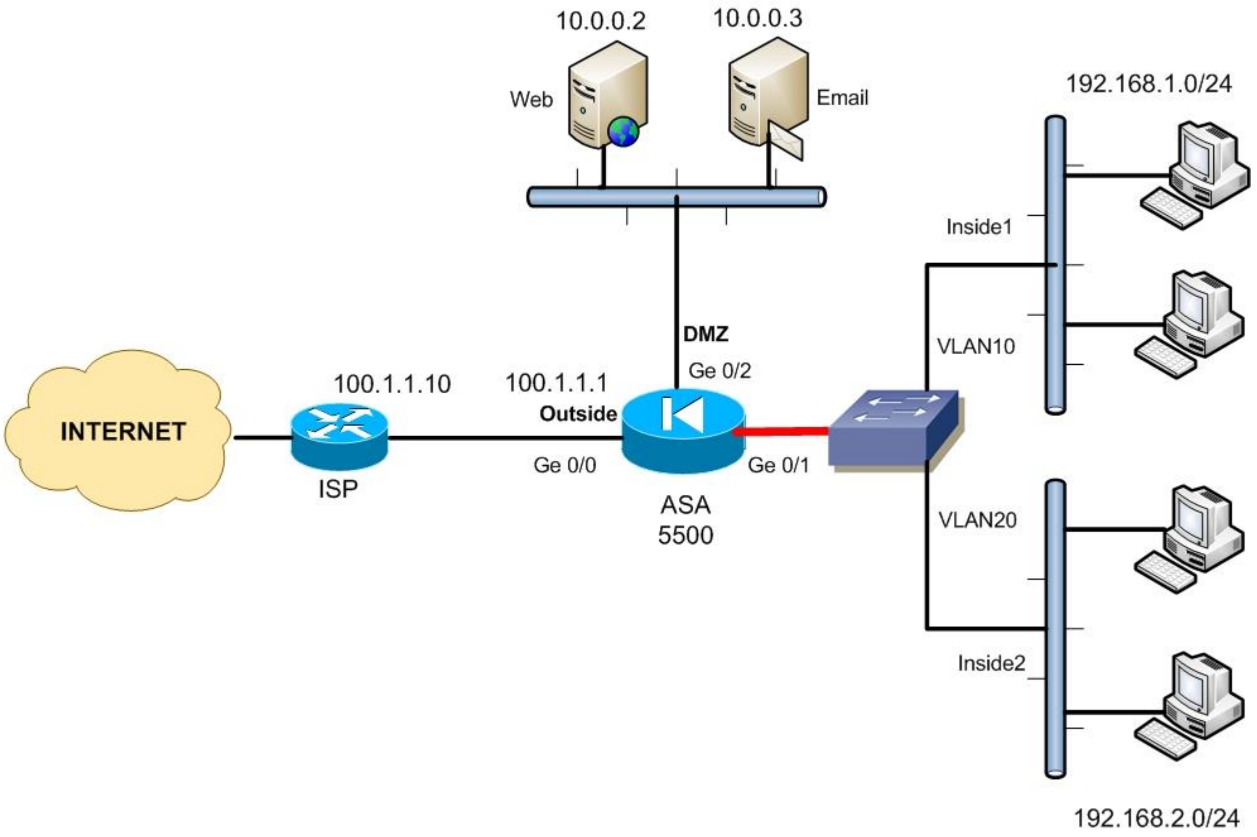
```
ASA-5505# show run
: Saved
:
ASA Version 7.2(3)
!
hostname ASA-5505
domain-name test.com
enable password xxxxxxxxxxxxxxxx encrypted
names
!
! Vlan 1 is assigned by default for all ports Ethernet0/1 to 0/7 which belong to the inside zone.
interface Vlan1
 nameif inside
 security-level 100
 ip address 192.168.1.1 255.255.255.0
!
! Vlan 2 is assigned to port Ethernet0/0 which belongs to the outside zone.
interface Vlan2
 nameif outside
 security-level 0
 ip address 100.1.1.2 255.255.255.252
!
! Assign Eth0/0 to vlan 2.
interface Ethernet0/0
 switchport access vlan 2
!


! By default, Eth0/1 to 0/7 are assigned to vlan 1. No need to change anything.
interface Ethernet0/1
!
interface Ethernet0/2
!
interface Ethernet0/3
!
interface Ethernet0/4
!
interface Ethernet0/5
!
interface Ethernet0/6
!
interface Ethernet0/7
!
passwd xxxxxxxxxxxxxxxxxx encrypted
ftp mode passive
dns server-group DefaultDNS
 domain-name test.com
```

```
! Create an ACL on the outside that will allow only echo-reply for troubleshooting purposes. Use a
!deny all with log at the end to monitor any attacks coming from outside.
access-list outside_in extended permit icmp any any echo-reply
access-list outside_in extended deny ip any any log
pager lines 24
logging asdm informational
mtu inside 1500
mtu outside 1500
icmp unreachable rate-limit 1 burst-size 1
asdm image disk0:/asdm-523.bin
no asdm history enable
arp timeout 14400
! Do PAT using the outside interface address
global (outside) 1 interface
! Translate ALL inside addresses
nat (inside) 1 0.0.0.0 0.0.0.0
access-group outside_in in interface outside
route outside 0.0.0.0 0.0.0.0 100.1.1.1 1
timeout xlate 3:00:00
timeout conn 1:00:00 half-closed 0:10:00 udp 0:02:00 icmp 0:00:02
timeout sunrpc 0:10:00 h323 0:05:00 h225 1:00:00 mgcp 0:05:00 mgcp-pat 0:05:00
timeout sip 0:30:00 sip_media 0:02:00 sip-invite 0:03:00 sip-disconnect 0:02:00
timeout uauth 0:05:00 absolute
! Configure Local authentication for firewall management (For accessing the Firewall you need to
!use the username/password configured later).
aaa authentication serial console LOCAL
aaa authentication telnet console LOCAL
aaa authentication ssh console LOCAL
no snmp-server location
no snmp-server contact
snmp-server enable traps snmp authentication linkup linkdown coldstart
! Allow internal hosts to telnet to the device
telnet 192.168.1.0 255.255.255.0 inside
telnet timeout 5
! Allow an external management host to ssh from outside for firewall management
ssh 100.100.100.1 255.255.255.255 outside
ssh timeout 5
console timeout 0
! Assign a DNS server to internal hosts
dhcpd dns 200.200.200.1
!
! Assign IP addresses to internal hosts
dhcpd address 192.168.1.10-192.168.1.40 inside
dhcpd enable inside
!
class-map inspection_default
 match default-inspection-traffic
!
policy-map type inspect dns preset_dns_map
 parameters
```

```
  message-length maximum 512
policy-map global_policy
 class inspection_default
  inspect dns preset_dns_map
  inspect ftp
  inspect h323 h225
  inspect h323 ras
  inspect netbios
  inspect rsh
  inspect rtsp
  inspect skinny
  inspect esmtp
  inspect sqlnet
  inspect sunrpc
  inspect tftp
  inspect sip
  inspect xdmcp
!
service-policy global_policy global
username admin password xxxxxxxxxxxxxx encrypted
prompt hostname context
: end
```

Configuration Example 2: ASA Firewall with DMZ and Two Internal Zones

In this scenario we will illustrate an ASA 5500 series Firewall (any model except 5505) with four security zones. One Outside, one DMZ, and two Internal Zones. The two Internal zones will be implemented on the same physical interface (Ge0/1) using two subinterfaces (Ge0/1.10 and Ge0/1.20). The DMZ zone will host a Web Server and an Email Server. We will use static NAT for the DMZ servers to translate their private IP addresses to public. Also we will impose traffic restrictions to the two Internal Zones. Inside1 users will be allowed to access only Web and Email, and Inside2 user will have unrestricted Internet access.



The complete configuration follows below. See the Blue Color comments for clarifications.

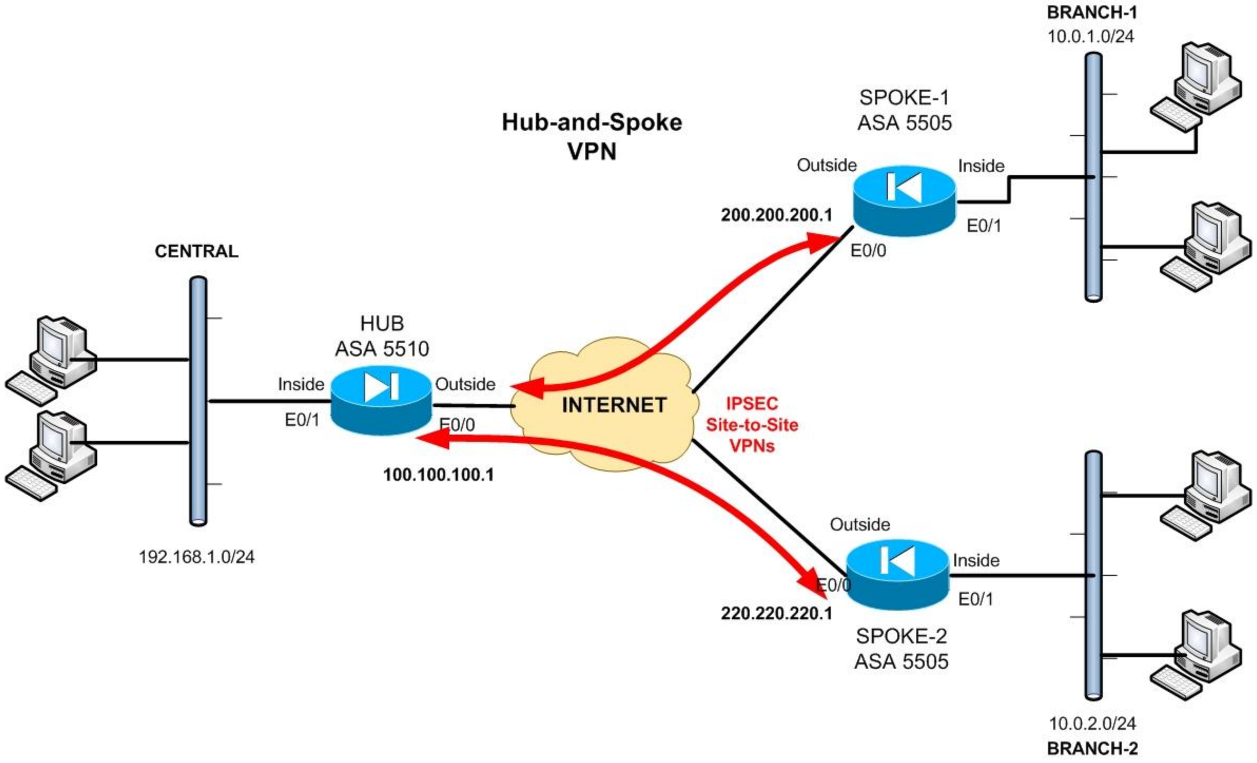
```
ASA-5500# show run
: Saved
:
ASA Version 7.2(2)
!
hostname ASA-5500
domain-name test.com
enable password xxxxxxxxxxxxxxxxx encrypted
names
dns-guard
!
interface GigabitEthernet0/0
 description CONNECTION TO OUTSIDE INTERNET
 speed 100
 duplex full
 nameif outside
 security-level 0
 ip address 100.1.1.1 255.255.255.0
!
! Use the same Physical Interface Ge0/1 to create two internal zones using Vlans
interface GigabitEthernet0/1
 no nameif
 no security-level
 no ip address
!
interface GigabitEthernet0/1.10
 description CONNECTION TO INSIDE 1
 vlan 10
 nameif inside1
 security-level 80
 ip address 192.168.1.1 255.255.255.0
!
interface GigabitEthernet0/1.20
 description CONNECTION TO INSIDE 2
 vlan 20
 nameif inside2
 security-level 90
 ip address 192.168.2.1 255.255.255.0
!
interface GigabitEthernet0/2
 description CONNECTION TO DMZ
 nameif DMZ
 security-level 50
 ip address 10.0.0.1 255.255.255.0
!
interface GigabitEthernet0/3
 shutdown
 no nameif
 no security-level
 no ip address
```

```
!
interface Management0/0
 shutdown
 no nameif
 no security-level
 no ip address
!
passwd xxxxxxxxxxxxxxxxxx encrypted
!
banner motd           ** W A R N I N G **
banner motd Unauthorized access prohibited. All access is
banner motd monitored, and trespassers shall be prosecuted
banner motd to the fullest extent of the law.
no ftp mode passive
dns server-group DefaultDNS
 domain-name test.com
object-group service WEB-PORTS tcp
 port-object eq 80
 port-object eq 443

! Allow access from Internet to our Web Server and Email Server
access-list OUTSIDE_IN extended permit tcp any host 100.1.1.2 object-group WEB-PORTS
access-list OUTSIDE_IN extended permit tcp any host 100.1.1.3 eq 25

! Inside1 zone is only allowed to access web and email
access-list INSIDE1_IN extended permit tcp 192.168.1.0 255.255.255.0 any eq http
access-list INSIDE1_IN extended permit tcp 192.168.1.0 255.255.255.0 any eq https
access-list INSIDE1_IN extended permit tcp 192.168.1.0 255.255.255.0 any eq smtp
access-list INSIDE1_IN extended permit tcp 192.168.1.0 255.255.255.0 any eq pop3
access-list INSIDE1_IN extended permit udp 192.168.1.0 255.255.255.0 any eq dns

! Inside2 zone is allowed to access all protocols
access-list INSIDE2_IN extended permit ip 192.168.2.0 255.255.255.0 any

pager lines 24
mtu outside 1500
mtu inside1 1500
mtu inside2 1500
mtu DMZ 1500
icmp unreachable rate-limit 1 burst-size 1
asdm image disk0:/asdm-522.bin
no asdm history enable
arp timeout 14400
nat-control
! Do PAT on the Outside and DMZ interfaces for internal hosts
global (outside) 1 interface
global (DMZ) 1 interface
nat (inside1) 1 192.168.1.0 255.255.255.0
nat (inside2) 1 192.168.2.0 255.255.255.0
```

```
! Create permanent static NAT mappings for our DMZ servers
static (DMZ,outside) 100.1.1.2 10.0.0.2 netmask 255.255.255.255
static (DMZ,outside) 100.1.1.3 10.0.0.3 netmask 255.255.255.255
access-group OUTSIDE_IN in interface outside
access-group INSIDE1_IN in interface inside1
access-group INSIDE2_IN in interface inside2
route outside 0.0.0.0 0.0.0.0 100.1.1.10 1
timeout xlate 3:00:00
timeout conn 1:00:00 half-closed 0:10:00 udp 0:02:00 icmp 0:00:02
timeout sunrpc 0:10:00 h323 0:05:00 h225 1:00:00 mgcp 0:05:00 mgcp-pat 0:05:00
timeout sip 0:30:00 sip_media 0:02:00 sip-invite 0:03:00 sip-disconnect 0:02:00
timeout uauth 0:05:00 absolute
username admin password xxxxxxxxxxxxx encrypted privilege 15
aaa authentication serial console LOCAL
aaa authentication ssh console LOCAL
aaa authentication telnet console LOCAL
ssh 192.168.1.0 255.255.255.0 inside1
ssh timeout 20
ssh version 2
console timeout 0
!
class-map inspection_default
 match default-inspection-traffic
!
!
policy-map type inspect dns migrated_dns_map_1
 parameters
  message-length maximum 512
policy-map global_policy
 class inspection_default
  inspect dns migrated_dns_map_1
  inspect ftp
  inspect h323 h225
  inspect h323 ras
  inspect netbios
  inspect rsh
  inspect rtsp
  inspect skinny
  inspect esmtp
  inspect sqlnet
  inspect sunrpc
  inspect tftp
  inspect sip
  inspect xdmcp

!
service-policy global_policy global
prompt hostname context
: end
```

CONFIGURATION EXAMPLE 3: HUB-AND-SPOKE IPSEC VPN WITH THREE ASA

This is a very common and useful scenario which you can scale it to a bigger number of Spokes depending on your network topology. Many Enterprises usually have a big Central site which shares data resources with several remote Branches. You can build a WAN data network between your Central and Branch sites using dedicated communication lines (very expensive) or use cheap Internet connectivity to build a private IPSEC Hub-and-Spoke VPN, as illustrated in the example network below. The Central site is equipped with an ASA 5510 firewall, while the Branch sites with 5505 models. To setup our Hub-and-Spoke VPN, we need to create two Site-to-Site VPN tunnels between Central – Branch1 and Central – Branch2.



The complete configurations follow below. See the Blue Color comments for clarifications.

CENTRAL SITE HUB CONFIGURATION

```
HUB# show run
: Saved
:
ASA Version 7.2(3)
!
hostname HUB
domain-name test.com
enable password xxxxxxxxxxxxxx encrypted
names
dns-guard
!
interface Ethernet0/0
 nameif outside
 security-level 0
 ip address 100.100.100.1 255.255.255.248
!
interface Ethernet0/1
 nameif inside
 security-level 100
 ip address 192.168.1.1 255.255.255.0
!
interface Ethernet0/2
 shutdown
 no nameif
 no security-level
 no ip address
!
interface Ethernet0/3
 shutdown
 no nameif
 no security-level
 no ip address
!
interface Management0/0
 shutdown
 no nameif
 no security-level
 no ip address
 management-only
!
passwd xxxxxxxxxxxxxxxxxx encrypted
boot system disk0:/asa723-k8.bin
ftp mode passive
dns server-group DefaultDNS
 domain-name test.com

access-list outside_in extended permit icmp any any echo-reply
access-list outside_in extended deny ip any any log
```

```
! Select which traffic must be excluded from NAT.
access-list nat0_acl extended permit ip 192.168.1.0 255.255.255.0 10.0.1.0 255.255.255.0
access-list nat0_acl extended permit ip 192.168.1.0 255.255.255.0 10.0.2.0 255.255.255.0
! Select the Interesting Traffic to be encrypted
access-list BRANCH1-VPN extended permit ip 192.168.1.0 255.255.255.0 10.0.1.0 255.255.255.0

access-list BRANCH2-VPN extended permit ip 192.168.1.0 255.255.255.0 10.0.2.0 255.255.255.0

pager lines 24
logging enable
logging trap debugging
mtu outside 1500
mtu inside 1500
icmp unreachable rate-limit 1 burst-size 1
asdm image disk0:/asdm521.bin
no asdm history enable
arp timeout 14400
nat-control
global (outside) 1 interface
! Do not translate Interesting Traffic
nat (inside) 0 access-list nat0_acl
nat (inside) 1 192.168.1.0 255.255.255.0
access-group outside_in in interface outside
route outside 0.0.0.0 0.0.0.0 100.100.100.2 1
timeout xlate 3:00:00
timeout conn 1:00:00 half-closed 0:10:00 udp 0:02:00 icmp 0:00:02
timeout sunrpc 0:10:00 h323 0:05:00 h225 1:00:00 mgcp 0:05:00 mgcp-pat 0:05:00
timeout sip 0:30:00 sip_media 0:02:00 sip-invite 0:03:00 sip-disconnect 0:02:00
timeout uauth 0:05:00 absolute
aaa authentication ssh console LOCAL
aaa authentication serial console LOCAL
aaa authentication telnet console LOCAL
no snmp-server location
no snmp-server contact
snmp-server enable traps snmp authentication linkup linkdown coldstart
! Create a Phase 2 transform set for encryption and authentication protocols.
crypto ipsec transform-set espSHA3DESproto esp-3des esp-sha-hmac
! Create a crypto map with two entries (10 and 20) for the two IPSEC VPNs with the Branches
crypto map outside_map 10 match address BRANCH1-VPN
crypto map outside_map 10 set peer 200.200.200.1
crypto map outside_map 10 set transform-set espSHA3DESproto
crypto map outside_map 20 match address BRANCH2-VPN
crypto map outside_map 20 set peer 220.220.220.1
crypto map outside_map 20 set transform-set espSHA3DESproto
! Attach the crypto map to the outside interface
crypto map outside_map interface outside
! Enable also the Phase 1 isakmp to the outside interface
crypto isakmp enable outside
```

```
! Create the Phase 1 isakmp policy
crypto isakmp policy 10
 authentication pre-share
 encryption 3des
 hash sha
 group 2
 lifetime 86400

telnet 192.168.1.0 255.255.255.0 inside
telnet timeout 5
console timeout 0
!
class-map inspection_default
 match default-inspection-traffic
!
!
policy-map type inspect dns migrated_dns_map_1
 parameters
  message-length maximum 512
policy-map global_policy
 class inspection_default
  inspect dns migrated_dns_map_1
  inspect ftp
  inspect h323 h225
  inspect h323 ras
  inspect netbios
  inspect rsh
  inspect rtsp
  inspect skinny
  inspect esmtp
  inspect sqlnet
  inspect sunrpc
  inspect tftp
  inspect sip
  inspect xdmcp
!
service-policy global_policy global
username admin password xxxxxxxxxxxxxx encrypted
! Create tunnel groups for the two IPSEC VPNs
tunnel-group 200.200.200.1 type ipsec-l2l
tunnel-group 200.200.200.1 ipsec-attributes
 pre-shared-key branch1vpnkey
 isakmp keepalive threshold 30 retry 5

tunnel-group 220.220.220.1 type ipsec-l2l
tunnel-group 220.220.220.1 ipsec-attributes
 pre-shared-key branch2vpnkey
 isakmp keepalive threshold 30 retry 5

prompt hostname context
```

BRANCH-1 SPOKE CONFIGURATION

```
SPOKE-1# show run
: Saved
:
ASA Version 7.2(3)
!
hostname SPOKE-1
domain-name test.com
enable password xxxxxxxxxxxxxx encrypted
names
!
interface Vlan1
 nameif inside
 security-level 100
 ip address 10.0.1.1 255.255.255.0
!
interface Vlan2
 nameif outside
 security-level 0
 ip address 200.200.200.1 255.255.255.0
!
interface Ethernet0/0
 switchport access vlan 2
!
interface Ethernet0/1
!
interface Ethernet0/2
!
interface Ethernet0/3
!
interface Ethernet0/4
!
interface Ethernet0/5
!
interface Ethernet0/6
!
interface Ethernet0/7
!
passwd xxxxxxxxxxxxxxxx encrypted
ftp mode passive
dns server-group DefaultDNS
 domain-name test.com
access-list VPN-TO-HUB extended permit ip 10.0.1.0 255.255.255.0 192.168.1.0 255.255.255.0
access-list NONAT extended permit ip 10.0.1.0 255.255.255.0 192.168.1.0 255.255.255.0
access-list OUTSIDE_IN extended permit icmp any any echo-reply
access-list OUTSIDE_IN extended deny ip any any log
pager lines 24
logging asdm informational
```

```
mtu inside 1500
mtu outside 1500
icmp unreachable rate-limit 1 burst-size 1
asdm image disk0:/asdm-523.bin
no asdm history enable
arp timeout 14400
global (outside) 1 interface
nat (inside) 0 access-list NONAT
nat (inside) 1 10.0.1.0 255.255.255.0
access-group OUTSIDE_IN in interface outside
route outside 0.0.0.0 0.0.0.0 200.200.200.2 1
timeout xlate 3:00:00
timeout conn 1:00:00 half-closed 0:10:00 udp 0:02:00 icmp 0:00:02
timeout sunrpc 0:10:00 h323 0:05:00 h225 1:00:00 mgcp 0:05:00 mgcp-pat 0:05:00
timeout sip 0:30:00 sip_media 0:02:00 sip-invite 0:03:00 sip-disconnect 0:02:00
timeout uauth 0:05:00 absolute
aaa authentication serial console LOCAL
aaa authentication telnet console LOCAL
aaa authentication ssh console LOCAL
no snmp-server location
no snmp-server contact
snmp-server enable traps snmp authentication linkup linkdown coldstart
crypto ipsec transform-set espSHA3DESproto esp-3des esp-sha-hmac
crypto map IPSEC 10 match address VPN-TO-HUB
crypto map IPSEC 10 set peer 100.100.100.1
crypto map IPSEC 10 set transform-set espSHA3DESproto
crypto map IPSEC interface outside
crypto isakmp identity address
crypto isakmp enable outside
crypto isakmp policy 10
 authentication pre-share
 encryption 3des
 hash sha
 group 2
 lifetime 86400
telnet 10.0.1.0 255.255.255.0 inside
telnet timeout 5
ssh 100.100.100.1 255.255.255.255 outside
ssh timeout 5
console timeout 0
!
class-map inspection_default
 match default-inspection-traffic
!
!
policy-map type inspect dns preset_dns_map
 parameters
  message-length maximum 512
policy-map global_policy
 class inspection_default
```

```
  inspect dns preset_dns_map
  inspect ftp
  inspect h323 h225
  inspect h323 ras
  inspect rsh
  inspect rtsp
  inspect sqlnet
  inspect skinny
  inspect sunrpc
  inspect xdmcp
  inspect sip
  inspect netbios
  inspect tftp
  inspect pptp
!
service-policy global_policy global
username admin password xxxxxxxxxxxxxxxx encrypted

tunnel-group 100.100.100.1 type ipsec-l2l
tunnel-group 100.100.100.1 ipsec-attributes
 pre-shared-key branch1vpnkey
 isakmp keepalive threshold 30 retry 5

prompt hostname context
: end
```

BRANCH-2 SPOKE CONFIGURATION

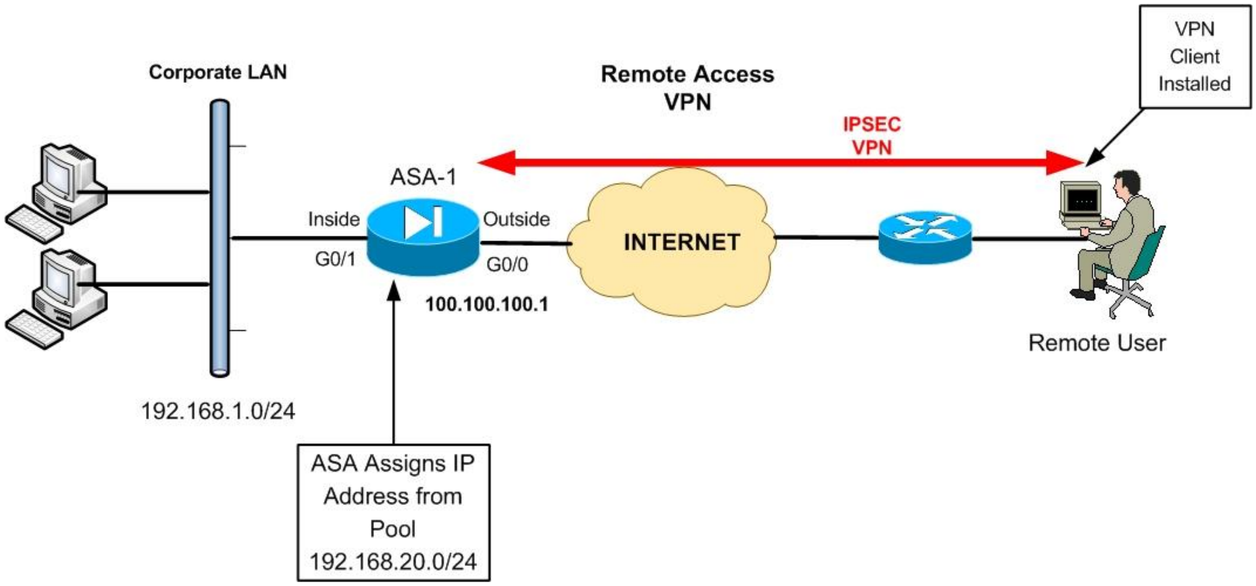
```
SPOKE-2# show run
: Saved
:
ASA Version 7.2(3)
!
hostname SPOKE-2
domain-name test.com
enable password xxxxxxxxxxxxxx encrypted
names
!
interface Vlan1
 nameif inside
 security-level 100
 ip address 10.0.2.1 255.255.255.0
!
interface Vlan2
 nameif outside
 security-level 0
 ip address 220.220.220.1 255.255.255.0
!
interface Ethernet0/0
 switchport access vlan 2
!
interface Ethernet0/1
!
interface Ethernet0/2
!
interface Ethernet0/3
!
interface Ethernet0/4
!
interface Ethernet0/5
!
interface Ethernet0/6
!
interface Ethernet0/7
!
passwd xxxxxxxxxxxxxxxx encrypted
ftp mode passive
dns server-group DefaultDNS
 domain-name test.com
access-list VPN-TO-HUB extended permit ip 10.0.2.0 255.255.255.0 192.168.1.0 255.255.255.0
access-list NONAT extended permit ip 10.0.2.0 255.255.255.0 192.168.1.0 255.255.255.0
access-list OUTSIDE_IN extended permit icmp any any echo-reply
access-list OUTSIDE_IN extended deny ip any any log
pager lines 24
logging asdm informational
```

```
mtu inside 1500
mtu outside 1500
icmp unreachable rate-limit 1 burst-size 1
asdm image disk0:/asdm-523.bin
no asdm history enable
arp timeout 14400
global (outside) 1 interface
nat (inside) 0 access-list NONAT
nat (inside) 1 10.0.2.0 255.255.255.0
access-group OUTSIDE_IN in interface outside
route outside 0.0.0.0 0.0.0.0 220.220.220.2 1
timeout xlate 3:00:00
timeout conn 1:00:00 half-closed 0:10:00 udp 0:02:00 icmp 0:00:02
timeout sunrpc 0:10:00 h323 0:05:00 h225 1:00:00 mgcp 0:05:00 mgcp-pat 0:05:00
timeout sip 0:30:00 sip_media 0:02:00 sip-invite 0:03:00 sip-disconnect 0:02:00
timeout uauth 0:05:00 absolute
aaa authentication serial console LOCAL
aaa authentication telnet console LOCAL
aaa authentication ssh console LOCAL
no snmp-server location
no snmp-server contact
snmp-server enable traps snmp authentication linkup linkdown coldstart
crypto ipsec transform-set espSHA3DESproto esp-3des esp-sha-hmac
crypto map IPSEC 10 match address VPN-TO-HUB
crypto map IPSEC 10 set peer 100.100.100.1
crypto map IPSEC 10 set transform-set espSHA3DESproto
crypto map IPSEC interface outside
crypto isakmp identity address
crypto isakmp enable outside
crypto isakmp policy 10
 authentication pre-share
 encryption 3des
 hash sha
 group 2
 lifetime 86400
telnet 10.0.2.0 255.255.255.0 inside
telnet timeout 5
ssh 100.100.100.1 255.255.255.255 outside
ssh timeout 5
console timeout 0
!
class-map inspection_default
 match default-inspection-traffic
!
!
policy-map type inspect dns preset_dns_map
 parameters
  message-length maximum 512
policy-map global_policy
 class inspection_default
```

```
  inspect dns preset_dns_map
  inspect ftp
  inspect h323 h225
  inspect h323 ras
  inspect rsh
  inspect rtsp
  inspect sqlnet
  inspect skinny
  inspect sunrpc
  inspect xdmcp
  inspect sip
  inspect netbios
  inspect tftp
  inspect pptp
!
service-policy global_policy global
username admin password xxxxxxxxxxxxxxx encrypted

tunnel-group 100.100.100.1 type ipsec-l2l
tunnel-group 100.100.100.1 ipsec-attributes
 pre-shared-key branch2vpnkey
 isakmp keepalive threshold 30 retry 5

prompt hostname context
: end
```

CONFIGURATION EXAMPLE 4: REMOTE ACCESS VPN

Continuing our VPN examples, we will configure here a Remote Access VPN scenario for providing secure connectivity to remote users over the Internet, as we have described in more detail in Chapter 5. Moreover, in this configuration example we will setup the "**split-tunneling**" feature which allows remote users to browse the Internet while connected with the IPSEC VPN. Because "split-tunneling" is not considered safe, it is disabled by default. This means that once the remote users initiate a Remote Access VPN with the central site, they can ONLY access the Corporate LAN network and nothing else. In order for the users to simultaneously access Internet resources and the Corporate LAN, then split-tunneling must be configured.



The complete configuration follows below. See the Blue Color comments for clarifications.

```
ASA-1# sh run
: Saved
:
ASA Version 7.2(3)
!
hostname ASA-1
domain-name test.com
enable password xxxxxxxxxxxxxxxxxx encrypted
names
dns-guard
!
interface GigabitEthernet0/0
 nameif outside
 security-level 0
 ip address 100.100.100.1 255.255.255.248
!
interface GigabitEthernet0/1
 nameif inside
 security-level 100
 ip address 192.168.1.1 255.255.255.0
!
interface GigabitEthernet0/2
 shutdown
 no nameif
 no security-level
 no ip address
!
interface GigabitEthernet0/3
 shutdown
 no nameif
 no security-level
 no ip address
!
interface Management0/0
 shutdown
 no nameif
 no security-level
 no ip address
!
passwd xxxxxxxxxxxxxxxxxxxxxx encrypted
boot system disk0:/asa723-k8.bin
ftp mode passive
dns server-group DefaultDNS
 domain-name test.com

access-list outside-in extended permit icmp any any echo-reply
access-list outside-in extended deny ip any any log
```

```
! Traffic between internal LAN and Remote Access clients must not be translated
access-list nat0_acl extended permit ip 192.168.1.0 255.255.255.0 192.168.20.0 255.255.255.0


! Remote Access client traffic destined to the internal LAN is permitted for split tunneling (i.e to
!access the Internet simultaneously)
access-list splittunnel standard permit 192.168.1.0 255.255.255.0

pager lines 24
logging enable
logging trap debugging
mtu outside 1500
mtu inside 1500

! Create a pool of addresses to assign for the remote access clients
ip local pool vpnpool 192.168.20.1-192.168.20.254

icmp unreachable rate-limit 1 burst-size 1
asdm image disk0:/asdm521.bin
no asdm history enable
arp timeout 14400
nat-control
global (outside) 1 interface
nat (inside) 0 access-list nat0_acl
nat (inside) 1 192.168.1.0 255.255.255.0

access-group outside-in in interface outside
route outside 0.0.0.0 0.0.0.0 100.100.100.2 1
timeout xlate 3:00:00
timeout conn 1:00:00 half-closed 0:10:00 udp 0:02:00 icmp 0:00:02
timeout sunrpc 0:10:00 h323 0:05:00 h225 1:00:00 mgcp 0:05:00 mgcp-pat 0:05:00
timeout sip 0:30:00 sip_media 0:02:00 sip-invite 0:03:00 sip-disconnect 0:02:00
timeout uauth 0:05:00 absolute
aaa authentication ssh console LOCAL
aaa authentication serial console LOCAL
aaa authentication telnet console LOCAL
no snmp-server location
no snmp-server contact
snmp-server enable traps snmp authentication linkup linkdown coldstart

crypto ipsec transform-set ESP-3DES-MD5 esp-3des esp-md5-hmac

! Create a dynamic crypto map for the remote VPN clients
crypto dynamic-map outside_dyn_map 20 set transform-set ESP-3DES-MD5
! Attach the dynamic crypto map to a static crypto map
crypto map outside_map 65535 ipsec-isakmp dynamic outside_dyn_map
crypto map outside_map interface outside
```

```
! Create a Phase 1 isakmp policy for the remote VPN clients
crypto isakmp enable outside
crypto isakmp policy 20
 authentication pre-share
 encryption 3des
 hash md5
 group 2
 lifetime 86400

! nat-traversal allows remote clients behind a NAT device to connect without problems.
crypto isakmp nat-traversal  20

telnet 192.168.1.0 255.255.255.0 inside
telnet timeout 5
console timeout 0
!
class-map inspection_default
 match default-inspection-traffic
!
!
policy-map type inspect dns migrated_dns_map_1
 parameters
  message-length maximum 512
policy-map global_policy
 class inspection_default
  inspect dns migrated_dns_map_1
  inspect ftp
  inspect h323 h225
  inspect h323 ras
  inspect netbios
  inspect rsh
  inspect rtsp
  inspect skinny
  inspect esmtp
  inspect sqlnet
  inspect sunrpc
  inspect tftp
  inspect sip
  inspect xdmcp
!
service-policy global_policy global

! Configure a group-policy and associate the split tunnel network list configured before
group-policy remotevpn internal
group-policy remotevpn attributes
 vpn-idle-timeout 30
 split-tunnel-policy tunnelspecified
 split-tunnel-network-list value splittunnel
username admin password xxxxxxxxxxxxxxxxxxxx encrypted
```

```
! Create a tunnel group with type “ipsec-ra” and associate the vpn pool configured before
tunnel-group remotevpn type ipsec-ra
tunnel-group remotevpn general-attributes
 address-pool vpnpool
 default-group-policy remotevpn
! The group name “remotevpn” and the pre-shared-key value must be configured also on the Cisco
!VPN client software
tunnel-group remotevpn ipsec-attributes
 pre-shared-key some-strong-key-here
prompt hostname context
: end
```

CONCLUSION

If you have studied carefully the information presented in this eBook, I'm confident that you will be able to tackle the most common ASA configuration scenarios that you will encounter in your professional career. The purpose of this eBook was to provide you the Foundation concepts for designing and implementing one of the most popular hardware firewalls in the market, the Cisco Adaptive Security Appliance. I know that the features, concepts and configuration capabilities that the Cisco ASA Firewall supports are much more than what is presented here. However, with the foundation base that this eBook provided you, it's fairly easy from now on to build up your knowledge with extra information provided from other Cisco documents for the ASA firewall.

Again, thank you for purchasing and reading this eBook. It has been a pleasure writing this handbook, and I really hope that you enjoyed it as well.

Please check out my Cisco Blog http://www.cisco-tips.com for technical tips and tutorials about Cisco products and solutions.

I will be glad to answer any questions you may have at asaebook@cisco-tips.com

GOOD LUCK TO YOUR PROFESSIONAL CAREER