

# Understanding Security Policies

---

## Introduction

Cisco Centri Firewall provides secure access and internetwork communications between private trusted networks and untrusted and unknown networks, such as the Internet, as well as among the subnets of a private network. It dynamically constructs a secure TCP/IP-based stack using kernel-level proxy services for all incoming and outgoing communications. These dynamic stacks enforce downloadable security policies that are based on your site's existing network security practices and security policies. We discuss these dynamic stacks in detail in the next chapter.

Several concepts, however, are key to understanding how these dynamic stacks work as well as what security policy is applied in the event that no security policies are selected by the administrator. In this chapter, we introduce you to the security stance of Cisco Centri Firewall and explain how it uses security policies to enforce communication rules.

The next section begins our discussion with the security stance of the Cisco Centri Firewall and explains its approach to security enforcement.

## Security Stance

Cisco Centri Firewall follows a minimalist and reductionist approach. This approach dictates that simplicity is best, and it follows the paradigm of “that which is not expressly permitted is prohibited.”

Cisco Centri Firewall uses the configurable Kernel Proxy technology that allows you to design your firewall to meet your own specifications for speed, reliability, and security. Because you must explicitly allow each user to have access to the firewall's services, a user cannot enable new network services or use new network applications to slip through the firewall server.

Knowing that you will want to set up certain services initially, the Centri Setup program prompts you with questions about which network services you would like to enable and how you would like to initialize your new security system.

---

**Note** You must understand the network services that you do allow through the firewall, and you must stay abreast of the security concerns related to those services. Almost daily, the media reports new weaknesses found within one network service or another. Your network users or users who are external to your network may attempt to exploit these advertised weaknesses and compromise the security of your network. The World Wide Web provides excellent resources for staying abreast of security advisories and issues, and we encourage you to visit these resources often. The Centri home page at [www.cisco.com/centri](http://www.cisco.com/centri) provides you a list of the most respected security advisories and their web pages.

---

The next section describes security policies and their role in protecting your network. With this introduction to security policies, you will be able to plan how you intend to use the Cisco Centri Firewall to protect your network.

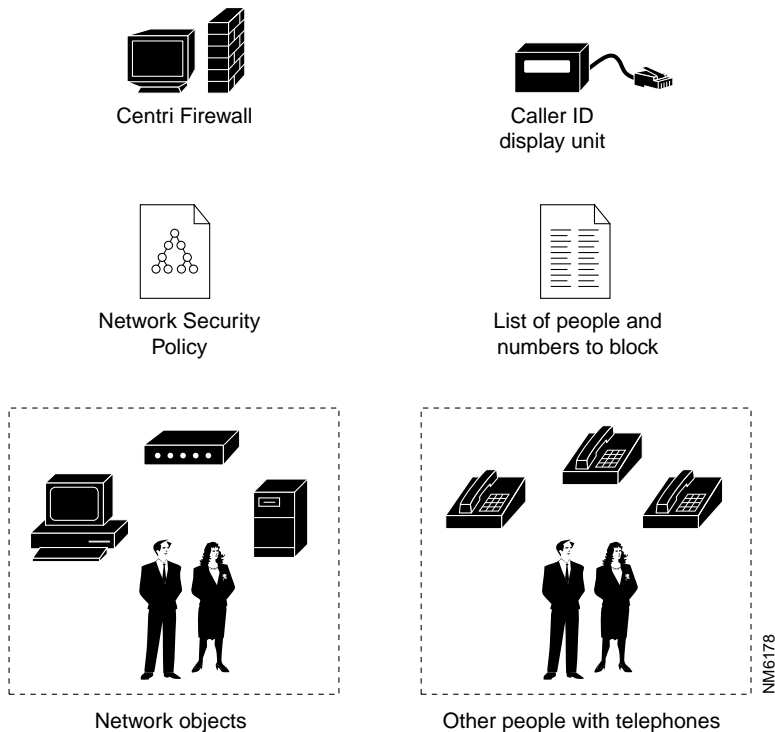
## How Security Policies Work

Security policies instruct Cisco Centri Firewall as to how it should control the traffic that traverses between internal trusted networks and external untrusted and unknown networks. By controlling which “sessions” can occur between two network objects, Cisco Centri Firewall secures the flow of network traffic. A *session* is simply a single act of communication between two network objects—much like a phone call between two people—with an explicit start and an explicit end.

To explain five concepts that are critical to writing strong security policies, we compare the similarities between the Cisco Centri Firewall and today’s Caller ID systems. The five concepts that we need to understand before writing a security policy are *sessions*, *session controls*, *actions*, *policy inheritance*, and *network objects*.

The relationships that exist among Cisco Centri Firewall, a security policy, and network objects are similar to the relationships among a Caller ID display unit, a list of people with whom you do and do not want to speak, and other people with telephones. These relationships are depicted in Figure 4-1.

**Figure 4-1**      **Analogy between Cisco Centri Firewall and Caller ID**



The Caller ID display unit lets you filter calls similar to the way that Cisco Centri Firewall filters sessions—it either accepts or rejects a request for a session. One notable difference is that Cisco Centri Firewall can also restrict which internal network objects can initiate sessions to external network objects. Although “call blocking” services do allow you to

block outgoing calls, such as 900 numbers, the Caller ID service by itself does not allow you to restrict who can be called from your phone. Basically, Cisco Centri Firewall lets you restrict who you can call as well as who can call you.

When you want to prevent a specific person from calling you, you enter the phone numbers from which the person typically calls and the response that the person should get when he or she attempts to call you. When you associate a response with the data about the caller, you are defining rules by which incoming calls should be processed.

Similarly, Cisco Centri Firewall filters session attempts according to the rules defined in a security policy. A security policy specifies which network objects are allowed to communicate with each other, and each security policy is designed to enforce some part of the overall network security policy defined by an organization. You can specify which internal network objects can communicate with which external network objects and vice versa. Other options exist by which you can filter communications, such as time of day, destination, and type of protocol being used to conduct the communication.

When a phone call comes in, the Caller ID unit displays information about the incoming call, such as the person's name, the originating phone number, and the time and date of the call. Using this information, you can determine whether to accept or ignore the call.

In networking terminology, we would say that the name and phone number act as *session control* criteria. These criteria provide information that we can use as the basis for acting upon a session, as well as information that is interesting to record and evaluate later (like a bi-directional phone bill that includes who called you and who you called). By evaluating incoming requests to start a new session against the session controls and responses defined in a security policy, Cisco Centri Firewall can determine whether to allow that session. If it does allow a session, Cisco Centri Firewall also determines how to modify the data that is transferred during that session.

Session controls are predominately specific to a network service and are used to act upon a session to provide stricter control over what is and what is not allowed during that session. Within Cisco Centri Firewall, two types of session controls exist: run-time and static.

*Run-time session controls* are those session controls that can be modified at the time the session request is received by the firewall. Run-time session controls are defined using security policies and can either apply to all communications or to a specific network service. *Static session controls* are those controls that are fixed for all sessions of a particular type. Most static session controls are defined when a network service is created

under the Network Services branch of these user interface. However, additional static session controls, such as HTTP filtering, are defined within the CentriFirewall property panel under the Networks tree.

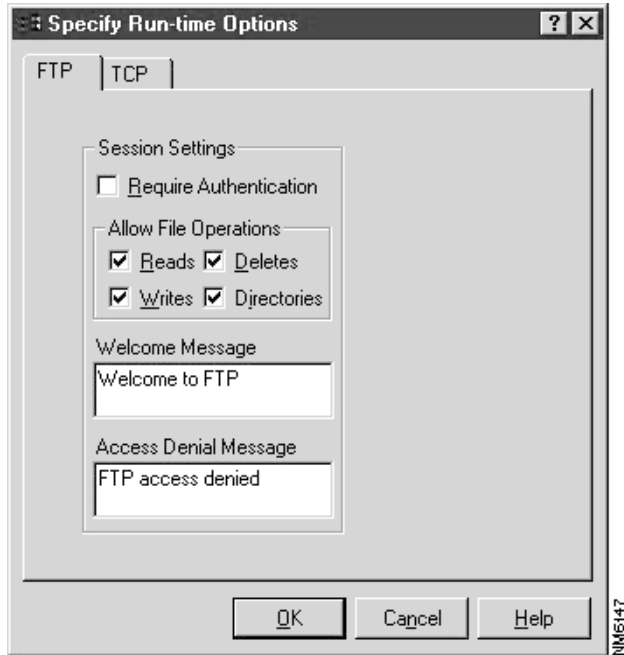
Table 4-1 identifies the types of session controls used by Cisco Centri Firewall, provides examples of the different controls, and highlights those controls that are not specific to a network service.

**Table 4-1           Types of Session Controls and Purpose**

Session Control Type	Purpose of Session Controls
Common Run-Time Controls	These session controls are common to all network services. They define the basic elements of any session, such as its time of day, date, User ID, Host ID, and type of service. These controls are defined using security policies.
Session-Specific Run-Time Controls	These session controls are specific to a network service. They define what is allowed during a session that uses a particular network service. For example, FTP, users can retrieve files but not write files to a destination server. Or when using HTTP, filter Java applets. These controls are defined using security policies.
Static Controls	Static session controls define a network service using non-variable information, such as the protocol number of the network service and the IP port number on which that service operates. These controls are defined when a network service is defined.

Figure 4-2 depicts the FTP run-time session controls, while Figure 4-3 depicts the TCP static session controls.

**Figure 4-2** Example Run-Time Session Controls



---

**Note** You can determine which options should be accessible to users of the FTP kernel proxy by defining them on-the-fly within a security policy. On-the-fly definitions allow you to modify specific settings on a per-condition basis—for each condition you define, you can specify different settings for the same network service. If you prefer, you can define the service settings semi-statically. To do so, you modify the service settings within the Network Services branch of the Services tree, and when you refer to that network service within a security policy, you accept the default values.

---

**Figure 4-3** Example Static Session Controls

The screenshot shows a dialog box titled "Proxy Settings" with a sub-header "TCP Proxy" and "Transport Layer". It is divided into two main sections: "Instance Settings" and "Session Settings".

**Instance Settings:**

- Port ID: 9999

**Session Settings:**

**Timeouts**

- Idle: 120 minutes (Use 0 to disable)
- Connect: 75 seconds

At the bottom of the dialog are four buttons: "< Back", "Next >", "Cancel", and "Help". The "Next >" button is highlighted. On the right side of the dialog, there is a vertical label "NM5146".

---

**Note** The TCP kernel proxy includes only one static session control. It identifies the port number on which this proxy will listen for incoming and outgoing session requests.

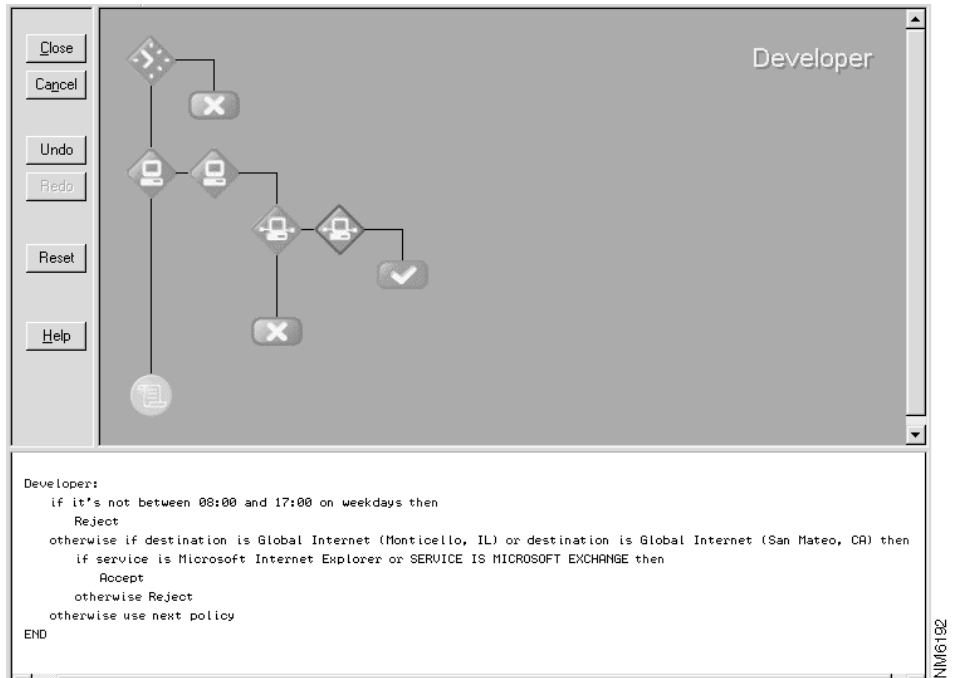
---

Unlike within the Caller ID analogy, some session controls allow you to modify the data of network packets that are part of a particular session while others allow you to determine whether to allow a session. Example session controls are start date, start time, session length, type of service, port number, user authentication requirements, and Java and ActiveX filtering.

Within the user interface, a security policy is represented as a decision tree of session controls for controlling network sessions and associated data. These decision trees specify the conditions that a session must satisfy and the responses, or *actions*, to apply to those sessions that do satisfy the conditions. The two actions that exist within network security policies are `Accept` and `Reject`. By accepting a session, you are stating that it is OK for the two specified network objects to communicate. By rejecting a session, you are stating that the two network objects are *not* allowed to communicate.

Figure 4-4 depicts an example decision tree within the Policy Builder control of the Centri Firewall user interface.



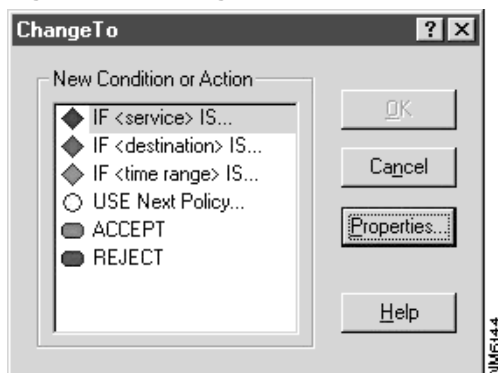
**Figure 4-4 A Decision Tree within Policy Builder**

**Note** Decision trees are collections of conditions and actions that represent security policies, which enforce part of the overall network security policy defined by an organization. Cisco Centri Firewall uses decision trees to enforce an administrator's policy for accepting and rejecting network traffic that traverses the firewall server. Security policies are constructed using the graphical Policy Builder interface within the Cisco Centri Firewall user interface.

Within Centri, each condition does not have to terminate in an action. It can continue with another condition branch in the same security policy (logically, it is an *else if* statement) or with another security policy altogether. We can direct the evaluation process to another

security policy by instructing the condition branch to `Use Next Policy`. The `Use Next Policy` statement simply says that you want to evaluate session requests that do not satisfy any condition branches within the current policy using the next security policy that is higher up in the direct path of the current node within the Networks tree. Unlike actions, the `Use Next Policy` statement does not terminate the evaluation process. Figure 4-5 depicts the conditions and actions that are available within the Policy Builder control of the Centri Firewall user interface.

**Figure 4-5** High-Level Conditions and Actions



---

**Note** Policy Builder presents the user with three condition types, two action types, and one statement. If the parameters of a network session satisfy a condition, an action (or statement) is applied to that session.

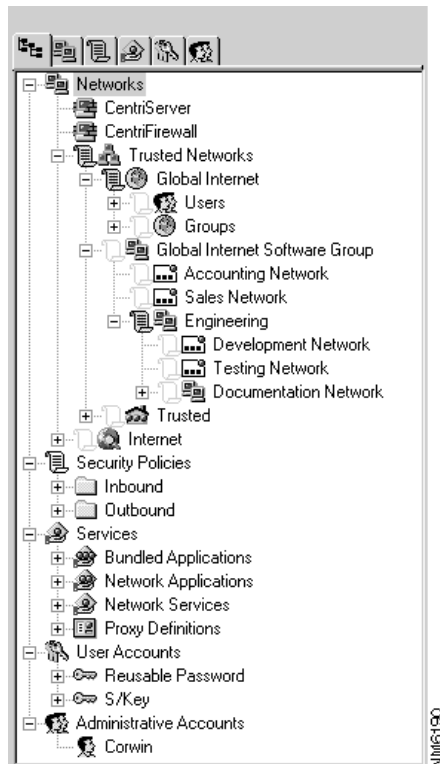
---

The `Use Next Policy` statement brings us to the important concept of policy inheritance. Policy inheritance does not have a counterpart within our Caller ID analogy, but it is an important concept to understand before you can develop effective network security policies.

*Policy inheritance* refers to Cisco Centri Firewall's ability to use recursive lists of security policies. If a security policy on a lower node of the Networks tree has the statement "Use Next Policy" applied to a condition branch, then the next policy up and in the direct path of that node is applied. This ability is transferred all the way up to the Trusted Networks,

Domains, or Internet branch if the policies below those branches use the “Use Next Policy” action. Dominance is an attribute of the lowest node to which a security policy is applied. If the parameters of a session request match two security policies within a direct path, the one applied to the lowest node in that path is applied to that session.

**Figure 4-6 Security Policy Inheritance in Cisco Centri Firewall**



**Note** Policy inheritance provides for flexible design and enforcement of an organization’s network security policy. It also eases the burden of managing large networks.

Within the Cisco Centri Firewall user interface, the rules for policy inheritance are processed according to the layout of the Networks tree. Security policies are processed from the top of the Networks tree down to the bottom and from the children nodes to parent nodes (top-down, inside-out).

The benefit of policy inheritance is most obvious when managing large networks, multiple departments, or numerous users and workstations requiring special privileges. Using policy inheritance allows you to tune your network's security based on the "exceptional" needs of specific users—needs that do not apply to most users of your network. More general policies, which are policies that are applicable to most of your network, are applied to the higher-level nodes within the Networks tree. On the lower-level nodes of the Networks tree, you can append exceptions to those general policies using more specific policies.

As you may have guessed, network objects are similar to the different people whom can call you. Within Centri, you can have multiple trusted network objects that which normally reside on your internal network much like having multiple phones within your home. Because you can only control communications between internal network objects and external network objects, trusted network objects normally communicate with multiple untrusted and unknown network objects, similar to all of the phones outside of your home.

A *trusted network object* is a network object over which you have administrative control of its security policy. It represents or exists on a trusted network. An *untrusted network object* is a network object that you know about but one for which you cannot specify and administer the security policy. It represents or exists on an untrusted network. An *unknown network object* is simply one that is unknown to you; you do not know of its existence or its IP address.

We can identify network objects by IP addresses and subnetwork masks (192.168.1.1, 255.255.255.0) or by familiar hostnames, such as `\\trumpet`. These grouping mechanisms are similar to phone numbers, (217) 555-1212, and quick access numbers on your telephone, such as a button with a fireman's hat that you program with the number of the fire department. In the Caller ID example, we can filter out all 900 numbers by blocking the 900 area code. We might also want to be more specific in our filtering by only blocking an area code and a prefix in combination, such as (900) 555-####.

With Cisco Centri Firewall, we can also identify network objects by Windows NT Domain names, Group accounts, or individual User accounts. Windows NT is responsible for mapping these network objects to the addresses that it understands. This ability makes it

possible to apply security policies to familiar network objects on homogeneous Windows-based networks, and it allows you to apply security policies based on users no matter where they log into your network.

Because you can apply security policies to different types of network objects that can actually represent the same network object (such as a Windows NT user on a specific host), you must understand the order in which security policies are applied. Cisco Centri Firewall assumes that if you want to apply a security policy to a specific user, then the security policy applied to a user should take precedence over a security policy applied to the computer from which the user is logging in. Security policies are evaluated in the following order:

- 1 Windows NT Domains, Group accounts, and User accounts
- 2 Logical Networks
- 3 Physical Networks

---

**Note** Within the user interface, *logical network objects* represent network objects in a manner that provides meaning to the administrator. For example, administrators can define logical network objects that organize network objects in groups that mimic the administrator's own organizational models.

*Physical network objects* represent specific network objects as they appear within the topology of the installed network. Actual physical layout is important because it helps the firewall server automatically derive routing rules for logical network objects and identifies where security policies should be applied by identifying where ranges of network objects exist.

---

We can summarize this section by stating that security policies control the access into and out of your network, much as a Caller ID display unit allows you to filter phone calls to your home. You can use security policies to evaluate network sessions using conditions consisting of session controls and to specify what actions should be applied to sessions satisfying those conditions. Security policies are applied to the network objects that you want to protect, and policy inheritance allows you to apply more general rules to higher-level network abstractions while still applying case-specific security policies to those network objects requiring special permissions.

The next chapter describes how the Cisco Centri Firewall enforces security policies, identifies and defines the major components of the architecture, and explains how it prevents common attacks through detailed evaluation of network packets and intelligent countermeasures.