Chapter 9

Authentication

Solutions in this chapter:

- Authentication Overview
- Users and Administrators
- User Authentication
- Session Authentication
- Client Authentication

☑ Summary
☑ Solutions Fast Track
☑ Frequently Asked Questions
Introduction

Using Check Point NGX, you can control the traffic coming into or going out of your networks. A good definition of your networks, hosts, gateways, and services allows you to have granular control of traffic through the Security Gateway. However, there are times when you will need or want to authenticate specific users who are accessing your resources.

For example, an administrator might have to download privileged files using a restricted user's workstation, and would need to be granted special privileges for a specific amount of time. Networks that use DHCP with different classes of users in the same network would need to authenticate privileged users to grant them access to the resources they need. Enterprises might have a need for registering in the log the specific user accessing a specific Web site.

With authentication, Check Point NGX's features are greatly expanded and complement already strong security with the ability to implement security on a per user basis. Once you understand how NGX Authentication works, you will probably find many uses for it in your environment.

Authentication Overview

Check Point NGX works based on the information it has to permit or deny a connection. The firewall has no knowledge of which user is logged into a Microsoft Active Directory, or if a user is moving among different machines. To be able to authenticate a particular user that is crossing the firewall, it needs additional information to match the user and the connection. The main topic of this chapter addresses the best way to authenticate users so they can access privileged resources. There have not been many changes between how authentication works in NG with Application Intelligence and how it works in NGX, and we list the major changes in this chapter.

We will first address the issue of which users can authenticate. Check Point NGX is flexible enough to authenticate users created in various sources, databases, or external directory servers. We will then examine the different types of authentication that NGX allows, which are called User, Session, and Client Authentication. These authentication schemes are for unencrypted authentication.

Using Authentication in Your Environment

Using Authentication involves additional configuration of the firewall and planning an environment that allows users to access the resources they need. Some of the environments that can benefit from authentication include the following.
You use DHCP without IP reservations in your network, but you need to give a few of your users access to special resources.

Your CIO wants strict logging of the traffic habits of your users, and so you need your log to contain the username of every connection from your internal networks.

A support technician needs to download drivers and antivirus programs from the Internet, on machines in a restricted segment that are not allowed to access the Internet.

You have an extranet site and want to add an additional layer of password security via your firewall.

Users and Administrators

Think of a user as an entity: Bob, Peter, and so forth. To recognize (or authenticate) a user, the user either needs to know something (a password) or have something (a digital certificate). This chapter focuses on passwords, since Digital Certificates are for VPNs only. Most companies already have some sort of user database (MS Active Directory, a RADIUS server, etc.), and would like to integrate this database with their firewall, through the use of an Authentication Scheme.

Managing Users and Administrators

Before you can authenticate users, you need to define them and place them in groups. Check Point NGX is very flexible in this sense. You can use NGX’s built-in user database, as well as external user directories. Let’s first focus on the built-in database, which you’ll probably use the most or at least have to interact with most often.

There are two ways you can access and edit the user database. You can access the Manage Users and Administrators dialog from the Manage | Users and Administrators menu (look at Figure 9.1). This dialog includes a listing of all user-related objects: users, groups, templates, administrators, external users groups, LDAP, and so on.
You can also select the Users tab in the Object Tree, and then expand the different entity classes to edit their objects (as in Figure 9.2). You can right-click on any entity class to add new objects.

**Figure 9.2 Object Tree Listing of User Entities and Their Icons**
The first item you will see in the Manage Administrators dialog box will be a yellow icon named `cpconfig_administrators`. It represents the administrators configured by the `cpconfig` utility in the SmartCenter server. In NGX, you can define only one administrator via `cpconfig` (in the `cpconfig` menu, it now says Administrator instead of previous versions’ Administrators). If you have upgraded from NG, you can migrate the existing `cpconfig` administrators to the SmartDashboard by using the `cp_admin_convert` command in the SmartCenter server (you need to use expert mode in SecurePlatform).

Each entity that you can create in the Users and Administrators dialog box is represented by a different icon. Administrators have crowns over them, groups are represented by two users, templates are outlines, and external users have a circle around them. Look at Figure 9.2 to identify the different icons.

Permission Profiles

Before you create an administrator, you need to create a Permissions Profile. Go to the Manage | Permission Profiles dialog, and select New... | Permissions Profile... (see Figure 9.3).

Figure 9.3 The Permissions Profiles Dialog

In the General tab, you can name the profile, select a color and enter a comment. In the Permissions tab, you allow access via two methods:
SmartPortal and Console Applications  Administrators can use SmartDashboard, and all the Smart management applications, as well as the SmartPortal web interface.

SmartPortal only  Administrators can access only the SmartPortal web interface, which is read-only and designed for auditors and restricted administrators.

In the Permissions tab, you also can choose the following profiles (see Figure 9.4):

- **None**  Use this to disable an administrator's permissions.
- **Read/Write All**  Allows full access to all NGX management applications. A Read/Write All profile can also select Manage Administrators, which will allow the administrator to create, modify, and delete other administrators from the SmartDashboard. It will also grant access to the Permission to Install setting for objects.
- **Read-Only All**  Administrators will be able to read every configuration, but won’t be able to change anything.
- **Customized**  Here you can create a personalized profile for administrators with very specific functions. The permissions for each option can be None (disable the option with the Check Box next to the item), Read Only, and Read/Write.

You can select the following specific functions for a customized profile:

- **SmartUpdate**  Administrators can use SmartUpdate for managing product updates and assigning licenses. An administrator with Read/Write SmartUpdate access will automatically have Read/Write access to the Objects, LDAP and Users databases, the Security and QoS policies, and the Log Consolidator, Eventia Reporter, and UserAuthority Web Access.
- **Objects Database**  Working with the networks objects and services in the SmartDashboard interface.
- **Check Point Users Database**  Working with the internal user database.
- **LDAP Users Database**  Working with an external LDAP database using the SmartDirectory functionality (which requires a separate license).
- **Security Policy**  Working with the Security and Address Translation rules and installing a policy (with Read/Write access).
- **QoS Policy**  Working with the QoS rules and installing a policy (with Read/Write access).
- **Log Consolidator**  Working with the Consolidation Policy. Eventia Reporter uses that policy for compiling information from the logs.
- **Eventia Reporter**  Working with the Eventia Reporter tables.
- **Monitoring**  Access to the SmartView Monitor database for statuses.
- **UserAuthority Web Access**  Working with the UserAuthority Web Access product.
- **ROBO Gateways Database**  Working with Remote Office/Branch Office Gateways, using the SmartLSM (Large Scale Manager) application.
- **Events Database**  Working with the Eventia Analyzer database.
- **Event Correlation Policy**  Working with the Eventia Analyzer Events database.
- **Track Logs**  Accessing the Traffic Log and Active sessions in the SmartView Tracker. Users with Read/Write permissions can purge and switch the logs, and to Block Intruders from the Active Sessions page.
- **Audit Logs**  Accessing the Active sessions and Audit Logs in the SmartView Tracker. Users with Read/Write permissions can purge and switch the logs, and block intruders from the Active Sessions page.
Administrators

The administrators are the users who have access to the configuration of the firewall. Depending on the Permissions Profile assigned to the administrators, they may or may not have permission to read and write to different parts of the security policy. Once you create or edit an administrator, you’ll see the following tabs.

**General Tab**

In the **General** tab of the Administrator you can give a name to the administrator and select a previously created Permissions Profile. You can also click **New...** and directly create a new profile. **View Permissions Profile** allows you to view and edit existing profiles. Look at Figure 9.5.
You will find a Personal tab for both administrators and users. The Expiration Date field (in dd-mm-yyyy format) is used to set a valid time period for an administrator. For example, you can use an administrator with a set expiration date for an auditor that needs to review your policies during the next month. By default in NGX, the expiration date is December 31, 2008.

Enter a Comment for the Administrator and select a Color for the display of its icon. These fields are for informational use only.

Groups
You can select which Administrators Groups this administrator belongs to. You can Add and Remove from the Available Groups and the Belongs to Groups boxes.

Admin Auth
Here you can select what Authentication Scheme will be used for administrators, basically what you check the password against. The options are Undefined, SecurID, Check Point Password, OS Password, RADIUS, and TACACS. If you select Undefined, then the administrator will not be able to authenticate using a password, only a digital certificate.
Chapter 9 • Authentication

**Admin Certificates**

One of the advantages of using SmartDashboard administrators is that you can implement authentication via digital certificates generated by the Internal Certificate Authority (ICA). In this tab, you will see the certificate **State** (there is no certificate for this object, or Object has a certificate), and the Distinguished Name if it has a certificate, as in Figure 9.6.

**Figure 9.6 The Certificates Tab**

If there is no certificate for the administrator, you can click **Generate and Save**. You will see a prompt warning that the generation cannot be undone unless revoking the certificate, and then you can enter and verify the password for the certificate. Finally, you will select where to save the .P12 file in your hard drive. This file you can distribute to the appropriate administrators, save on a USB device, and so on. Once the administrator has a certificate, you use View… to see its details, or Revoke to eliminate the certificate.

**Administrator Groups**

You can create Administrator Groups and place administrators in them. You can use Administrator Groups by editing a Check Point Gateway, using the **Advanced | Permissions to Install** tab (see Figure 9.7). Here you can **Remove** the Any group...
from the selected groups and add the specific Administrator Groups you want to grant install access to. Only administrators with Manage Administrators permission can modify these properties.

**Figure 9.7 The Permissions to Install Tab**

![Permissions to Install Tab](image)

**User Templates**

Before you create users, you have to understand templates. Since there are many options to configure on users, templates save you time by preconfiguring options at the time of user creation. If you change an option on a template, it does not affect users already created from that template, only future users. The Standard template is preconfigured for quickly adding a user if you don’t want to go into details. Let’s look at the different Tabs you see once you select **New… | Template…**

**General**

Select the **Name** the template will have. This is the name you will use when selecting **New… | User from Template…**
282 Chapter 9 • Authentication

Personal

The Expiration Date field (in dd-mm-yyyy format) is used to set a valid time period for users. For example, you can use a template for a group of temporary employees that will leave the company at a specific date. By default in NGX, the expiration date is December 31, 2008.

Enter a Comment for the Template and select a Color for the display of its icon. These fields are for informational use only.

Groups

You can select which User Groups the users will belong to. You can Add and Remove from the Available Groups and the Belongs to Groups boxes.

Authentication

Here you can select what form of Authentication Scheme will be used for these users. The options are Undefined, SecurID, Check Point Password, OS Password, RADIUS, and TACACS. If you select Undefined, then the user will not be able to authenticate using a password, only a digital certificate. You will only be able to select the Authentication Scheme, but won’t be able to enter a Password for the Template, for security reasons.

Location

Location refers to the users’ allowed sources and destinations. You will be able to select Network Objects and move them to either the Source or Destination boxes, or leave them as Any, as in Figure 9.8. The location then becomes a restriction as to where the users can connect from (i.e., IT_Users restricted to IT_Networks source location), and where the users can connect to (i.e., Extranet_Users restricted to Extranet_Servers destination). This field can give you flexibility in having few authentication rules that behave differently for specific users and groups. It’s a bit complex to keep track of, but if you need it, it’s very useful. When you configure an authentication rule, you can decide whether the rule has to intersect with the location of the users, or it can ignore it (we’ll look at that later).
Figure 9.8 The Location Tab

Time
You can select what day of the week and range of time **User may connect at**. Although you can select several **Day in week options**, you are limited to a single range for **Time of Day**.

Encryption
Encryption is used for VPN Remote Access, and will be covered in Chapter 12, “SecuRemote, SecureClient, and Integrity.” If you select **IKE** for **Client Encryption**, the user will be able to participate in the Remote Access community.

User Groups
You can create User Groups and place users in them. You can select a **Name**, **Comment**, and **Color** for the group, and **Add** and **Remove** from the **Not in Group** to the **In Group** boxes, as in Figure 9.9.
User Groups can also contain other groups in a nested fashion. When you add a group to another group, NGX will ask **Would you like to add each member of the group separately?** and each group would be expanded in the new group. With nested groups, if you change a group the change will be reflected in the parent group, but that will not happen if you expand the group.

Check Point NGX does not reference individual users directly in rules or object properties, so if you have a user, you will want to place them in the appropriate group. If they’re not in any group, they’re still part of the All Users group.

**Users**

When you want to create a user, you have to work based on an existing template. Once you select **New… | User from Template…** (see Figure 9.10), you can select the initial template you want and then you will see a dialog box with many tabs. Let’s look at them.
**Figure 9.10** Creating a User from a Template

**General**
Select the **Login Name** for the users. It can have special characters in it, as well as spaces and periods, and long names.

**Personal**
The **Expiration Date** field (in dd-mm-yyyy format) is used to set a valid time period for a user. By default in NGX, the expiration date is December 31, 2008. Enter a **Comment** and select a **Color** for the display of its icon. These fields will be prepopulated with information from the Template.

**Groups**
You can select which User Groups the user belongs to. You can **Add** and **Remove** from the **Available Groups** and the **Belongs to Groups** boxes. The tab will be prepopulated according to the Template.

**Authentication**
Here you can select which **Authentication Scheme** the user will have. The options are Undefined, SecurID, Check Point Password, OS Password, RADIUS, and
Chapter 9 • Authentication

TACACS. If you select Undefined, then the user will not be able to authenticate using a password, only a digital certificate. If you select RADIUS or TACACS, you can select which server to use for verification.

If you select Check Point Password, you can click Enter Password to assign and verify it. The passwords should be four to eight characters in length. Check Point stores a hash for these passwords in the internal database. A hash function is an irreversible, one-way, highly-sensitive-to-change function, specifically the UNIX crypt function, which can use only eight characters with DES-based encryption. There is virtually no probability that two strings will have the same hash value, even if they differ by only a letter. When a user inputs a password, the gateway compares the hash of the password with the hash stored in the user database to authenticate the user. For those technically inclined, when a user’s password is modified, Check Point creates a random salt, which is then returned in the first two characters of the hashed result. You could programmatically create a file that uses the crypt function to create a large list of users and password that can then be imported with the fwm dbimport command.

Location

Here you can select specific Source and Destination locations for the users. The fields will be prepopulated from the template. Remember that when you configure an authentication rule, you can decide whether the rule has to intersect with the location of the users, or it can ignore it.

Time

You can select what day of the week and range of time User may connect at. Although you can select several Day in week options, you are limited to a single range for Time of Day. The fields will be prepopulated from the template.

Certificates

In this tab, you will see the Certificate State (the message can be There is no certificate for this object, The certificate is pending for the user, or Object has a certificate), and the Distinguished Name if a certificate exists. Digital Certificates for users apply only for Remote Access VPNs, and will be covered in the SecuRemote/SecureClient chapter.

Encryption

Encryption is used for VPN Remote Access, and will be covered in Chapter 12. If you select IKE, the user will be able to participate in the Remote Access community.
External User Profiles

If you’re working with external directory servers (RADIUS, TACACS, OS Password, SecurID), you would still need to define each user that exists in the external directory server, and select the appropriate authentication method for the user. This can be a tedious and error-prone process. By creating External User Profiles, you can deal with users who are not defined in the Check Point user database. If users are recognized by an external directory server, they will be granted permissions based on the appropriate External User Profile.

Match by Domain

This profile allows you to selectively query an external user database based on the Domain that the user enters. The important properties are in the General Tab.

If you use Distinguished Name (DN) format, you can select a specific organizational unit, organization, or country to authorize. Or, you can use Free Format, where the domain will be separated from the username by a character like @ or \ (for Microsoft), either before or after the username. In Free Format you can choose Any Domain as Acceptable, or a specific Domain Name you select. Finally, select whether to omit the domain name when requesting authorization at the external directory server. See Figure 9.11 for details.

Figure 9.11 External User Profiles | Match by Domain
The other tabs in the External User Profile (Personal, Groups, Authentication, Location, Time, Encryption) function as they do in the normal user entity. Remember that in this scenario, you’re leaving authentication and authorization decisions to an external entity.

**Match All Users**
If you don’t need to be selective of a domain users have to log with, you can use the *Match All Users External Profile*. It is named *generic* and will match any user recognized by the external directory server. Remember that in this scenario, you’re leaving authentication and authorization decisions to an external entity.

The other tabs in the External User Profile (Personal, Groups, Authentication, Location, Time, Encryption) function as they do in a normal user entity.

**LDAP Group**
If you are using SmartDirectory LDAP integration, you can create LDAP groups. You will give a name to the group, a comment and a color, and the Account Unit that the LDAP Group belongs to. You should have created the LDAP Account Unit from the Manage Servers and OPSEC Applications dialog box.

In the Group’s Scope, you can select to recognize all the Account-Unit users, or only those in a certain subtree, branch, and prefix, or only a group in a branch, with a DN prefix. You can also apply a filter to create a dynamic group (for example, all users in ou=Access).

**Understanding Authentication Schemes**
Check Point NGX is flexible enough to work with several external directory servers, where a user entity can be defined in the Internal Check Point database, but the password is verified from different sources. Check Point refers to these as Authentication Schemes.

**Undefined**
The Undefined Authentication Scheme is used for disabling the user’s ability to enter a password. This will force users to employ strong authentication with a digital certificate.

**SecurID**
Selecting SecurID as the Authentication Scheme will enable Check Point NGX to become an ACE/Agent for RSA’s SecurID Tokens. This integration will require use
of a special *sdconf.rec* generated by the ACE server, and will allow you to enter new
PIN numbers and reauthenticate often to secure servers. However, it’s a lot more dif-
ficult to configure than through SecurID’s RADIUS interface.

**Check Point Password**

If you select Check Point Password (called VPN-1 & FireWall-1 Password previ-
ously), you will enter the user’s password directly into NGX’s internal database.
Passwords are four to eight characters in length. Be aware that the only way to assign
or change passwords is through the SmartDashboard interface.

**RADIUS**

The *Remote Authentication Dial-In User Service* (RADIUS) is a standard protocol that
can authenticate users with a RADIUS server that holds a database. The RADIUS
protocol is very flexible and relatively secure. It uses a specific secret key for securing
the authentication and only authorized clients can request authentication from the
server. You can also set up backup servers in case one of them is out of service.

To configure RADIUS Authentication, first create a RADIUS Server from the
Manage | Servers and OPSEC Applications… dialog as in Figure 9.12. Select
New… | RADIUS… to create the server. Input the appropriate data in the
Name, Comment, and Color. For Host, select the physical server that is running
the RADIUS server. If you don’t have the server created, use the New… button to
create a new node. Select the Service to use, either NEW-RADIUS (the official
port number) or RADIUS (the most common port number used). The Shared
Secret is a password used to secure the information sent between the Check Point
Gateways and the RADIUS server(s). You can select the Version to use, either ver-
sion 1.0 or 2.0. Also select a Priority, to know which server to use first if there is
more than one available. You can also create RADIUS groups for high availability
and load sharing. Figure 9.13 shows the dialog box for creating a RADIUS server.
Figure 9.12 The Manage Servers and OPSEC Applications Dialog

Figure 9.13 Creating a RADIUS Server
TIP

New for RADIUS support in NGX is being able to select a RADIUS Protocol, either PAP or MS-CHAP v2, for improved Microsoft compatibility. Also new is support for RADIUS’ accounting features. If you enable IP Pool Management (used in VPN environments), the RADIUS server can assign IPs to users.

TACACS

The Terminal Access Controller Access Control System is a standard protocol that can authenticate users with an external user database. To configure TACACS Authentication, first create a TACACS Server from the Manage | Servers and OPSEC Applications… dialog. Select New… | TACACS… and input the appropriate data in the Name, Comment and Color, as in Figure 9.14. For Host, select the physical server that is running the TACACS server. If you don’t have the server created, use the New… button to create a new Node. Select the Type of server used, either TACACS or TACACS+, which is more secure. If selecting TACACS+ you can also enter a Secret Key for encryption. Finally select the Service to use (UDP TACACS or TCP TACACSPLUS).

Figure 9.14 Creating a TACACS Server
Using Microsoft Internet Authentication Service (IAS)

Microsoft Windows Servers (NT, 2000, and 2003) have a built-in component called the Internet Authentication Service (IAS). It is basically a RADIUS service integrated into the NT Domain or the Active Directory. RADIUS authentication does not require a Check Point SmartDirectory license. To make the connection between IAS and NGX:

1. Create the appropriate RADIUS Server in the SmartDashboard. In the Start | Programs | Administrative Tools | Internet Authentication Service management console, add a new RADIUS client. The IP address should be the firewall gateway’s IP facing the RADIUS server. If you have a cluster, add a client for each cluster member. Input the same Shared Secret you defined when creating the RADIUS server in the SmartDashboard.

2. In the Remote Access Policies, edit one of the policies listed (in NT and 2000 you’ll see only one, in 2003 select the one for Connections to other access servers). Edit the Profile and go to the Authentication tab. Select either Unencrypted Authentication or Microsoft Encrypted Authentication version 2, which is more secure. In the Advanced tab of the profile, eliminate all attributes.

3. Modify the Active Directory users to allow connections. Go to Start | Settings | Control Panel | Administrative tools | Active Directory Users and Computers, edit the properties of the user you want to authenticate using RADIUS. In the Dial-In tab, select Allow Access and Apply.

4. In the SmartDashboard, create a generic* user (an external user profile that matches all users) with RADIUS authentication. Include the generic* user in User Groups you insert in the rulebase.

5. You can differentiate among RADIUS users by setting the add_radius_groups property with the GUIdbEdit application, and then configuring the RADIUS server to send a Class attribute for different user groups. In SmartDashboard create groups named RAD_<ClassAttribute> and integrate them in the rulebase.
User Authentication

User Authentication allows privileged use of some common Internet protocols, with little change to the user experience. It works by intercepting connections that are passing through the firewall (Check Point calls this “folding” the connection), and modifying the traffic in such a way that the firewall asks users to identify themselves before allowing the connection to pass through. Since the users request a connection to their final destinations, User Authentication is a type of *Transparent* authentication; in other words, the user doesn’t need to go through an intermediate process. Also be aware that this type of authentication is very demanding on the firewall, if it needs to fold a large number of connections.

Because NGX needs to modify the traffic itself, it can do so only with four specific services that it can understand well: HTTP, FTP, Telnet, and RLOGIN. These services belong to the *Authenticated* group in the predefined Check Point services. When one of these services is used to access a restricted resource, and there’s a rule configured to allow User Auth for that connection, the traffic is modified so the user can enter a password to enable the traffic. For Telnet, FTP, and RLOGIN, the user sees an intermediary prompt from the firewall, and once the user authenticates, a new connection to the final destination is made. In the case of HTTP traffic, the firewall instructs the user’s browser to display an authentication dialog.

User Authentication is performed on each connection so that if a machine is being shared by different users (for example, in a client/server or thin client environment) each user will authenticate his or her connection only, which is safe. Because the firewall needs to examine the traffic of these authenticated services, it requires more processing power from the firewall than Session and Client authentication.

Configuring User Authentication in the Rulebase

To allow user authentication, create a new rule. In the source field, select *Add User Access*, and add the user groups that will be able to authenticate using that rule. You can also select a restriction for the origin of the user connections. Add the appropriate *Destination* to that rule (if you want to authenticate all traffic to the Internet, leave it as *Any*), select which *Services* you’ll authenticate (remember, only HTTP, FTP, Telnet, or RLOGIN—if you enter any additional services, the policy will have compilation errors), select *User Auth* as the Action, and add appropriate *Track*, *Time*, and *Comment* configurations. See Figure 9.15 for a User Authentication Rule.
You should edit or verify the properties of the Action field in the User Authentication rule, as in Figure 9.16.

**Figure 9.16 Properties of the User Authentication Action**

**UserAuth | Edit Properties | General | Source**

**Source** is used to control whether the Restrict To location in the source of the rule has to intersect the configured location of the user in the database, or if it can ignore the location. Select **Ignore** to override the user database.
UserAuth | Edit Properties | General | Destination

**Destination** is similar to Source, in which you can choose to ignore the user database so that if a user has a configured destination location, and that location does not intersect the destination of the rule, the authentication will still take place.

UserAuth | Edit Properties | General | HTTP

Here you can select whether you want to allow authentication to a restricted number of servers, or to any accessible machines. The default is **Predefined Server**. When Predefined Servers is selected, users will be able to access only the list of servers that can be defined in the **Policy | Global Properties | FireWall-1 | Security Servers** dialog. To be able to authenticate traffic to any destination on the Internet, you need to select **All Servers** in the properties of the User Auth action.

**Interacting with User Authentication**

Depending on the service you will authenticate (HTTP, FTP, Telnet, or RLOGIN), the way your users will authenticate is different. Remember that if users need to resolve Web addresses (e.g., www.checkpoint.com), they will need to have access for domain-udp requests through the firewall, or to an internal DNS server. Let’s see what the user experience is for these services.

**Telnet and RLOGIN**

User Authentication for Telnet and RLOGIN is easy for the end user to understand. When a user tries a command like `telnet 172.29.109.1`, the firewall will intercept this command and present its own Telnet prompts, as in Figure 9.17. Once the user correctly authenticates, he or she will see the prompts from the original destination and can proceed accordingly.
FTP

User Authentication for FTP is a bit more complex to understand. The user will still use the command `ftp 172.29.109.1`, which will be intercepted by the firewall’s FTP security server, but the username will have to reflect the user that is authenticating at the FTP server (i.e., administrator), the user that is authenticating at the firewall (i.e., user), and the final destination of the FTP connection (even though it was used in the original ftp command). For example, in this case the username will be `administrator@user@172.29.109.1`, and the password will be the passwords of the FTP user separated by an @ sign from the password of the firewall user—`ftppassword@fwpassword`. Then a connection to the FTP server will be established. Look at Figure 9.18 for details.
HTTP

User authentication for HTTP is simple for users to use. When activated, an HTTP connection that should be authenticated is modified by the firewall in such a way that the user's browser displays an authentication dialog box or prompt, using HTTP's authentication mechanism. The prompt says FW-1: No user. Once the user authenticates with this prompt, the requested site is displayed in the browser. Look at Figures 9.19 and 9.20 for examples using Microsoft Internet Explorer and Mozilla Firefox.

WARNING

Selecting User Authentication for HTTP traffic to the Internet will mean that a user might need to authenticate as much as 10 times before seeing a single web page, since current web pages usually reference images or code from other sites, and the browsers need to reauthenticate for each different site.
In Figure 9.21 you can look at the entry in the SmartView Tracker generated from the authenticated HTTP access.
Figure 9.21 SmartView Tracker Entry for HTTP User Authentication

If the requested site itself requests a password, users can enter the list of usernames and passwords in reverse order, separated by an @ sign (or even @@ if you’re crossing multiple authentication daemons). You might need to use this for Outlook Web Access.

Placing Authentication Rules
Check Point rules are sequential, which means that once a rule can be applied to traffic passing through the firewall, that rule is applied and the next packet is processed. However, there is one exception. If there is a rule that allows traffic to a destination, even if there is a rule that would require authentication before that rule, the traffic will pass through without the need for users to authenticate. The reason for this is that an authentication rule does not deny traffic that fails to authenticate.
The best scenario for User Authentication is when you need to grant authenticated access to a limited number of servers you control (i.e., your intranet or servers you host in a data center). The reason for this is that you can control the HTML code of those servers so that they do not refer sites different from their own. Your users will have an easy-to-understand experience and will authenticate only once. Remember that if your code refers to external servers, each connection will prompt the user to authenticate again.

User Authentication for Telnet is also easy to understand and implement, as your users receive simple prompts to authenticate first to the firewall and then to their final destinations.

For example, look at Figure 9.22. If a machine in Net_Internal tries to access Web_DMZ, you would think that NGX would require the user to authenticate. If the user successfully authenticates, the traffic will be allowed. However, if the user doesn’t successfully authenticate, NGX would continue processing rules and would find the Any to Web_DMZ rule, and would allow the traffic nevertheless. Therefore, NGX will not ask for authentication.

Figure 9.22 Placing User Authentication in the Rulebase

You could still require users in Net_Internal to authenticate before accessing Web_DMZ by creating a rule that will block the http traffic before it is allowed, as in Figure 9.23. This would force Check Point NGX to authenticate the users.
User Authentication is a useful and convenient way to add an extra layer of verification to your users’ connections. However, many times you will find that the default configuration breaks some connections or isn’t secure enough for your needs. There are a myriad of configurations that you can make to User Authentication, and here we cover some of the most frequently requested ones.

**Advanced Topics**

User Authentication is a useful and convenient way to add an extra layer of verification to your users’ connections. However, many times you will find that the default configuration breaks some connections or isn’t secure enough for your needs. There are a myriad of configurations that you can make to User Authentication, and here we cover some of the most frequently requested ones.

**Eliminating the Default Authentication Banner**

You should always try to avoid disclosing unnecessary information. Check Point’s default banners (the initial identification) for the FTP, Telnet, and RLOGIN security servers identify your firewall as a Check Point Firewall. You can avoid this by setting the `undo_msg` property to `true` in the `dbedit firewall_properties`, using the `dbedit` utility. From a command prompt (or in SecurePlatform’s expert mode), run `dbedit`, log in with a username and password, and type `modify properties firewall_properties undo_msg true` and then `update properties firewall_properties`. After installing a policy, you will no longer see the default prompt. Be aware that some FTP clients need a banner for them to connect. In the following topic you can set your own banner.

**Changing the Banner**

Traffic that is intercepted by the firewall, be it FTP, Telnet, RLOGIN, or HTTP, displays a message from Check Point NGX requesting authentication. It is advisable to change this default message to a generic one that doesn’t broadcast the firewall’s
identity, and that can include additional information for users. You can select a file to be presented instead of the regular banner for FTP, Telnet, and RLOGIN (not for HTTP), in the SmartDashboard’s Global Properties | Firewall | Security Servers dialog box, as in Figure 9.24.

**Figure 9.24 Changing the Security Server Banners**

Use Host Header as Destination

If you are making HTTP connections, once the connection is authenticated, the firewall needs to redirect the original query to the intended destination. It does so by looking at the original URL’s IP address and redirecting the user’s browser to that IP. However, if the firewall resolves the destination URL to a nonroutable IP (i.e., the non-NAT’ed IP), or if the web server is configured to need the Host Header for access (i.e., a web hosting service that shares one IP with multiple web pages), then the connection will fail. To solve this, enable the setting `http_use_host_h_as_dst` using Policy | Global Properties | SmartDashboard Customization | Configure | FireWall-1 | Web Security | HTTP Protocol, as in Figure 9.25.
When you activate user authentication, the firewall needs to process and modify the traffic going through it, using the Security Servers (which are better explained in the Content Security chapter). In HTTP's case in particular, having the traffic pass through the Security Server also means that the verifications that the firewall makes on HTTP traffic will be made on all authenticated traffic. These verifications might break some connections that your users regularly make. In particular, accessing sites like Hotmail and downloading attachments can be affected and should be thought of when activating this.

Two changes you will need to make are enabling `http_allow_content_disposition` and `enable_propfind_method` settings in the Policy | Global Properties | SmartDashboard Customization | Configure | FireWall-1 | Web Security | Security section, as in Figure 9.26.
Session Authentication

Another method available for authentication in Check Point NGX is Session Authentication. This method uses a client program, called the Session Authentication Agent, which usually is installed on each machine that will be used for authentication. The Session Authentication Agent can be used for any service, and authenticates a particular session by the user. When the firewall encounters a rule with Session Authentication, it tries to query the appropriate machine using the FW1_snauth service on port 261. The Agent will then automatically present the user with an authentication dialog. Session Authentication combines User and Client Authentication, since it can authenticate per session for any service.

However, you need to consider that the Session Authentication Agent is a separate program that has to be installed in each user's machines, and that it is a Windows-Only program. Furthermore, the last available version is from NG Feature Pack 1 and there's no NGX version (yet).
WARNING

The firewall will try to authenticate all sessions that fall under a Session Authentication rule. If you configure a rule for authenticating any service to any destination, you might end up authenticating every DNS query and NBT broadcast sent, overloading the firewall authentication mechanism and network. Limit the number of services that use session authentication.

Configuring Session Authentication in the Rulebase

To allow session authentication, create a new rule. In the source field, select Add User Access, and then add the user groups that will be able to authenticate, and optionally restrict the location from where those groups can connect. Then, add the appropriate destination to that rule (if you want to authenticate all traffic to the Internet, leave it as Any), select which services you’ll authenticate (try not to use Any), select SessionAuth as the Action, and add the appropriate Track, Time, and Comment configurations, as in Figure 9.27.

Figure 9.27 Configuring Session Authentication in the Rulebase

Let’s take a look at the Action column of the Session Authentication Rule. Right-click on the field and select Edit Properties, as in Figure 9.28.
Figure 9.28 Properties of the Session Authentication Action

SessionAuth | Edit Properties | General | Source
Source is used to control whether the Restrict To location in the source of the rule has to intersect the configured location of the user in the database, or if it can ignore the location. Select Ignore to override the user database.

SessionAuth | Edit Properties | General | Destination
Destination is similar to Source, in which you can choose to ignore the user database so that if a user has a configured destination location, and that location does not intersect the destination of the rule, the authentication will still take place.

SessionAuth | Edit Properties | General | Contact Agent At
The Session Authentication Agent doesn’t have to be installed in the machine that wants to access restricted resource (the source of the connection). You can define the location that Check Point NGX will query to authenticate a connection. The options are Src (the source of the connection), Dst (the destination of the connection), or you can select a specific host or gateway configured in the object tree. You would be able to configure this so that a supervisor grants access to specific resources, or for authenticating X Windows connections, or when a user wants to authenticate incoming connections.
SessionAuth | Edit Properties
| General | Accept Only SecuRemote/SecureClient Encrypted Connections
If you select this option, only users connecting via a remote access VPN will be able to authenticate.

SessionAuth
| Edit Properties | General | Single Sign-On
Selecting this option will restrict authentication to UserAuthority and UserAuthority SecureAgent.

Configuring Session Authentication Encryption
One of the advantages of Session Authentication is that you can configure the Session Authentication Agent to communicate using SSL, to prevent password sniffing. You may change the `snauth_protocol` setting from the Policy | Global Properties | SmartDashboard Customization | Configure | FireWall-1 | Authentication | Session Authentication dialog box, as shown in Figure 9.29:

- **None** (or blank) No encryption of the authentication will be performed.
- **SSL** SSL will be active on all Session Authentication Agents. If you have an old agent you will not be able to authenticate.
- **SSL + None** SSL will be active on all Session Authentication Agents, but if you have an old agent you will be able to authenticate without encryption.

Figure 9.29 Changing the Session Authentication Protocol
The Session Authentication Agent

The Agent is needed to use Session Authentication. It’s a small 2MB program that installs without requiring a reboot. Since it is not included in the Check Point NGX CDs, you will have to download it from the Check Point User Center (usercenter.checkpoint.com) NG FP1 downloads section, as in Figure 9.30, or get it from a Check Point NG or Check Point NG with Application Intelligence CD.

Figure 9.30 Downloading the Session Authentication Agent

Once installed, the Session Authentication Agent will show up in the Windows Taskbar Notification Area as a blue circle with yellow and green arrows (see Figure 9.31), and open port 261 for listening to authentication requests from firewalls. If a request is received, the user automatically will see a Check Point prompt for username and password, making it easy for a novice user to understand what’s going on.
If you double-click on the agent icon, you can modify its configuration via three sections.

**Configuration | Passwords | Ask for Password**

You can configure three behaviors for the Agent, as shown in Figure 9.32:

- **Every Request**  Passwords will not be cached, and every request will need to be authenticated. Very secure, but also very cumbersome.

- **Once per session**  Passwords will be cached the first time the user enters it. No reauthentication will be needed until the user logs out of the Windows session.

- **After X minutes of inactivity**  Passwords will be cached the first time the user enters it. Reauthentication will be needed after the agent doesn’t authenticate connections for the amount of minutes entered.

**Figure 9.32 Configuring the Session Authentication Agent Passwords**
Configuration | Allowed FireWall-1
| Allow Authentication Request From

You can configure the Agent to accept requests from Any IP Address, or specify up to three IPs that the agent will respond to requests, as in Figure 9.33.

**Tip**

By limiting the requests, you can prevent a rogue firewall or program from contacting agents and receiving usernames and passwords.

**Figure 9.33** Configuring the Session Authentication Agent Allowed FireWall-1

Configuration | Allowed FireWall-1 | Options

Here you can configure whether to Allow Clear Passwords or not. Uncheck Allow Clear Passwords to ensure SSL encryption is used, if you’ve configured the protocol for it. You can also configure whether the agent should resolve addresses with DNS (see Figure 9.34).
Figure 9.34 Configuring the Session Authentication Agent Options

Interacting with Session Authentication

When you authenticate with Session Authentication, the user will be shown up to three prompts to authenticate a connection.

First, if it is the first time a particular firewall is requesting Session Authentication, the Session Authentication Agent will ask the user for permission to send authentication to that firewall, as in Figure 9.35. If accepted, it will also add the firewall’s IP to the list of gateways from which it will accept requests.

Second, the Agent will ask for the username. It will display the name of the firewall making the request, the destination of the connection, and the service requested, as in Figure 9.36.
Finally, the user will enter the password and can then be granted access to the desired resource, as in Figure 9.37.

Figure 9.37 Entering the Password for Session Authentication

In the SmartView Tracker, you can see an authentication entry with the FW1_smauth protocol, and following that, the actual session. For every session, you will see an authentication entry above it, as in Figure 9.38.
Client Authentication

Client Authentication is a versatile authentication method that can be used for most of your needs. Unlike User authentication, in which a connection is being authenticated, here you authenticate a machine or an IP (which the firewall considers the client). Client authentication is not transparent, which means that the connection has to be directed to the firewall so that it can ask for the specific authentication. Some of the benefits of client authentication are that any service can be authenticated, and that the authentication can last for a specific period of time or number of sessions (by default, five sessions in 30 minutes). Once a user achieves client authentication, traffic can flow freely with little intervention. Since the firewall doesn’t have to interpret or modify the passing connections, is it faster than user or session authentication and doesn’t intervene in the HTTP traffic passing through the gateway.
Warning

Client Authentication has some security disadvantages. In a multiuser environment (i.e., Citrix or Terminal Services), all requests originate from the same IP and will be given the same access. Also, if the user doesn’t sign off, other users that log at that machine will have permissions as the previously authenticated user until the authorization expires.

Configuring Client Authentication in the Rulebase

To allow client authentication, create a new rule above any rule that would block ports 900 and 259 to the firewall (usually the Stealth Rule). In the source field, select Add User Access, and then add the user group that will be able to authenticate, and optionally restrict to a location, where that group can connect from. Then, add the appropriate Destination to that rule (if you want to authenticate all traffic to the Internet, leave it as Any), select which Services you’ll authenticate (here you can use any service at all), select Client Auth as the Action, and add appropriate Track, Time, and Comment configurations, as in Figure 9.39.

Let’s take a look at the Action column of the Client Authentication Rule. There are many different behaviors that you have to select according to your desired policy. Once you select Client Auth as the Action, right-click on the field and select Edit Properties, and you will see the Client Authentication Action Properties window, as in Figure 9.40.
**Figure 9.40** Configuring Client Authentication Action Properties

![Client Authentication Action Properties](image)

**ClientAuth | Edit Properties | General | Source**

As in other authentication actions, **Source** is used to control whether the source of the rule has to intersect the configured **Location** of the user, or if it takes precedence. Select **Ignore** if you want to override the location configured for the user.

**ClientAuth | Edit Properties | General | Destination**

Client Authentication cannot determine the final destination of a connection, since users are authenticating directly to the firewall. Therefore, the destination field is grayed out and cannot be selected.

**ClientAuth | Edit Properties | General | Apply Rule Only If Desktop Configuration Options Are Verified**

Checking this box will allow the client to access resources granted in the rule, once the user authenticates, only if they are using Check Point SecureClient and the Secure Configuration Verification (SCV) has succeeded.
Chapter 9 • Authentication

ClientAuth | Edit Properties
| General | Required Sign-On

If you select **Standard Sign-On**, users will be able to access all resources permitted in the rule at which they authenticated. If you select **Specific Sign-On**, users will have to explicitly specify, through a form or a sequence of prompts, the services and destinations allowed for the client. Specific Sign-On is useful for a kiosk machine, where an administrator can authorize access to certain sites or services only, without interacting with the firewall administrator.

ClientAuth | Edit Properties
| General | Sign-On Method

The **Sign-On** method is one of the most important settings when using Client Authentication. Be sure to know how each method works to be able to select the most appropriate to your environment.

**Manual Sign-On**

Manual Sign-On method activates two ports on the Firewall Gateway for receipt of the authentication. They are port 900 using HTTP, and port 259 using Telnet (as in Figures 9.46, 9.47, and 9.48). Since users need to access these ports on the Firewall, the Client Authentication rule must be placed above the Stealth Rule (the one that drops all connections to the Firewall module).

This method is nontransparent, meaning that users will know they are first authenticating to a firewall, and then able to access the appropriate resources.

If you want to HTTP to port 900, you can look at Figure 9.41 where you enter the username, then in Figure 9.42 you enter the password, and in Figure 9.43, you select the Sign-On method. If you select Specific Sign-On, you will see Figure 9.44, where you can enter a list of services and destinations to authorize, and in Figure 9.45 the successful authentication screen (both for Specific and Standard Sign-On methods).
Figure 9.41 HTTP Manual Client Authentication—Entering the Username

![Image of entering the username]

Figure 9.42 HTTP Manual Client Authentication—Entering the Password

![Image of entering the password]
Chapter 9 • Authentication

**Figure 9.43** HTTP Manual Client Authentication—Selecting the Sign-On

![Authentication Form - Microsoft Internet Explorer](image)

**Figure 9.44** HTTP Manual Client Authentication—Specific Sign-On

![Authentication Form](image)
Figure 9.45 HTTP Manual Client Authentication—Successful Sign-On

If you want to Telnet to port 259, Figure 9.46 shows the Standard Sign-On method, and Figure 9.47 shows the Specific Sign-On method.

Figure 9.46 Telnet Manual Client Authentication—Standard Sign-On
Figure 9.47 Telnet Manual Client Authentication—Specific Sign-On

If you want to sign off, Figure 9.48 shows the Telnet client authentication Sign-Off method.

Figure 9.48 Telnet Manual Client Authentication—Sign-Off

Partially Automatic Sign-On

With Partially Automatic Authentication, if a user tries to access a resource that he or she could authenticate for, using any of the Authenticated services (remember,
HTTP, FTP, Telnet, or RLOGIN), then the firewall will intercept the connection and request authentication from the user, like it would do with User Authentication. Manual Authentication may still be used.

Once the user enters the username and password, the firewall interprets the authentication as it had been manually entered to the firewall as in client authentication. This is extremely useful, since now users will be required to authenticate only once and can use any resource that the rulebase allows them to. Partially Automatic Authentication is one of the most used methods of authentication. One thing to keep in mind is that as with User Authentication, it can be easy for an intruder sniffing the network to decipher usernames and passwords.

**Fully Automatic Sign-On**

With Fully Automatic Authentication, you further extend the ways the firewall can request authentication from the user. If an Authenticated service is used, the firewall intercepts the traffic and requests authentication as in User Authentication. For other services, it will try to invoke Session Authentication to authenticate the user at the connecting machine. Manual Authentication may still be used.

**Agent Automatic Sign-On**

With Agent Automatic Authentication, the firewall will try to authenticate connections using only the Session Authentication Agent at the connecting machine. Manual Authentication may still be used.

**Single Sign-On**

If you select Single-Sign On, the firewall will try to contact a User Authority server to query the identity of the user logged in at the station. You need to have a User Authority license, and the User Agent installed at the connecting machine.

**General | Successful Authentication Tracking**

Here you can select whether you want information or alerts sent to the log when a user successfully authenticates. If you select Alert it will also write the information to the log.

Once you configure the General properties for Client Authentication, you can configure the Limits for Client Authentication sessions. Figure 9.49 shows the tab for configuring these properties.
Limits | Authorization Timeout
Here you can select how long the user authorization lasts. Select indefinite to require an explicit sign-off from the user (via HTTP to port 900 or Telnet to port 259) to cancel the authorization (as in Figures 9.41 through 9.48). Select a specific time limit, in hours and minutes, if you want to require reauthentication after a time has lapsed. Select refreshable timeout if you want that as long as the connection is being used, the user will not be required to reauthenticate. This is similar to setting a time for a screen saver to come in—as long as there’s activity in the machine, the screen saver doesn’t come in. This is useful so that if a user leaves his machine, someone else can’t sit down and use it.

Limits | Number of Sessions Allowed
This limits the amount of open connections an authenticated user can make through the firewall. If you’re using FTP, Telnet, or RLOGIN connections, you could limit the number of sessions through this property. However, if you’re using HTTP connections, you will need to select Indefinite sessions because a browser will normally open many sessions when browsing a single page.
Advanced Topics

Client Authentication opens up the functions of authentication to more services and more situations. These additional functions also have additional features to configure, and we look at some of them here.

Check Point Gateway | Authentication

There are some properties that you need to configure and verify per gateway, to fine-tune the authentication experience for your users, as in Figure 9.50.

Figure 9.50 Check Point Gateway | Authentication

Enabled Authentication Schemes

It is very important to select which authentication schemes the gateway will allow for its users. If a user selects an authentication with a scheme that the gateway does not accept, the user will not be granted access to resources on the Gateway. You can select Check Point Password (previously VPN-1/FireWall-1 Password), SecurID, OS Password, RADIUS, and TACACS.
Authentication Settings

The user can select the following authentication settings:

- **User Authentication session timeout**  This setting (by default, 15 minutes) has two behaviors. For FTP and Telnet connections, connections with no activity are terminated after the timeout expires. For HTTP, it applies to the use of one-time passwords (i.e., SecurID tokens). Although the timeout hasn’t expired, the server will not request another one-time password for access to a previously authenticated server.

- **Enable Wait mode for Client Authentication**  This setting applies to Telnet Client Authentication, where the Telnet session remains open, and when the user closes the session (CTRL-C or some other manner), the authentication expires. If this option is not selected, the user will have to manually sign off or wait for the session timeout.

- **Authentication Failure Track**  Here, you can define if a failed authentication will generate an alert, a log, or no activity. We recommend at least logging all failed authentications.

**HTTP Security Server**

If you use an HTTP Security Server, you can configure an HTTP Proxy Server behind the Security Server. Enter the Host and Port for the proxy server to activate it.

**Global Properties | Authentication**

There are certain properties that are configured globally for all authentication performed by the Check Point gateways. Appropriately enough, you access them from the **Global Properties | Authentication** tab, as in Figure 9.51.
Failed Authentication Attempts
To prevent an intruder from brute-force guessing your passwords, the Failed Authentication Attempts setting can be used to terminate connections after a set number of failed attempts (i.e., wrong passwords). You can set a different amount of allowed tries for RLOGIN, Telnet, Client Authentication, and Session Authentication connections.

Authentication of Users with Certificates
For Remote Access VPNs, if you are using VPNs, you can restrict the gateway to accept only users who have a specific suffix in their certificates. Also, if an administrator initializes a certificate in the Certificate Authority, you can define how many days users have to pull that certificate before it expires.

Brute-Force Password Guessing Protection
To prevent an intruder from brute-force guessing your passwords, you can enable this protection, which is new for NGX. By setting a specific number of milliseconds to delay each authentication, you can dramatically affect any automatic password guessing system, while a user will barely notice a difference.
Chapter 9 • Authentication

Early Version Compatibility

If you are managing gateways prior to NG, the following setting can be applied globally. For NG or later gateways, these settings are set per gateway. By the way, since there isn’t the option to administer pre-NG gateways in NGX, you can probably ignore this section.

Registry Settings

Most of Check Point configuration is done from the SmartDashboard. However, some settings are not available through the SmartDashboard and have to be accessed directly in the Check Point internal registry. There are three ways you can access the registry:

- Some settings are available in the SmartDashboard through the Policy Menu | Global Properties | SmartDashboard Customization, which opens a tree of different settings you can change.
- You can use the GuiDBedit program, which is included in the installation folder of the Check Point SmartConsole clients. Its default location is C:\Program Files\Check Point\SmartConsole\R60\PROGRAM\GuiDBedit.exe. The program gives you complete access to the registry.
- You can use from the command line (or SecurePlatform’s expert mode) the dbedit program, which is a text interface to the registry. You will need to know the exact setting you want to access, since it isn’t very user-friendly.

Let’s look at some of the settings you might need when dealing with Authentication.

New Interface

The default Client Authentication HTTP interface requires four pages for a successful login: the username page, password page, method page, and the optional Specific Sign-On page. If you enable the hclient_enable_new_interface setting using the Policy Menu | Global Properties | SmartDashboard Customization | FireWall-1 | Authentication | Client Authentication | HTTP, the HTTP interface will combine the username and password pages into one, thus streamlining the user experience, as in Figure 9.52.
Use Host Header as Destination

If you are using Partially or Fully Automatic Sign-On with HTTP connections, once the connection is authenticated the firewall needs to redirect the original query to the intended destination. It does so by looking at the original URL’s IP address, and redirecting the user’s browser to that IP. However, if the firewall resolves the destination URL to a nonroutable IP (i.e., the non-NAT’ed IP), or if the Web server is configured to need the Host Header for access (i.e., a Web hosting service that shares one IP with multiple web pages), then the connection will fail. To avoid this, enable the `http_use_host_h_as_dst` setting using the Policy Menu | Global Properties | SmartDashboard Customization | FireWall-1 | Web Security | HTTP Protocol.

Opening All Client Authentication Rules

When you begin to create a complex policy with different rules for granting user access to different resources, you need to take into consideration that the default behavior for Client Authentication is to grant access for the rule only where you authenticated. If you need to authenticate once and be granted access by all rules that would permit the user, you have to enable the `automatically_open_ca_rules` setting using the Policy Menu | Global Properties | SmartDashboard Customization | FireWall-1 | Authentication | Client Authentication section.
Configuration Files

Besides the configuration of gateway properties, global properties, rules, and registry settings, some configuration files can change the authentication behavior, to include encryption or to present your own look and feel to the user.

Enabling Encrypted Authentication

Since Telnet and HTTP are not encrypted, Client Authentication is inherently less secure than Session Authentication. However, you can configure NGX to enable HTTPS Manual Authentication, which will give you the encryption you want when using the built-in HTTP server at port 900 for authentication. Look in the “Are You Owned?” sidebar for details.

Are You Owned?

Securing Client Authentication

Although Client Authentication is a great tool, flexible and easy to understand, you have to be aware of its security implications. When you enable Client Authentication and place the rule above the stealth rule, you’re opening ports 259 and 900 to the firewall. Remember that neither of these services is encrypted, so that you’re vulnerable to snifing attacks. Never allow Client Authentication from Any sources, as it’s easy for a scanner to detect these ports running on your firewall and try to brute force a username and password combination (use the new NGX Brute Force Protection to minimize this).

It’s a good idea to change the HTTP server on port 900 to an encrypted HTTPS server, and to disable Telnet authentication to port 259. Edit the $FWDIR/conf/fwauthd.conf file and change the line

```
900 fwssd in.ahclientd wait 900
```

to

```
900 fwssd in.ahclientd wait 900 ssl:defaultCert
```

and eliminate the line

```
259 fwssd in.aclientd wait 259
```

You could also change the default 900 and 259 ports to other port numbers, and edit the fw1_clnauth_http and fw1_clnauth_telnet services to reflect the new ports.
Custom Pages

If you will use Client Authentication’s HTTP interface, you will probably want to change its appearance and include your company’s logo, an unauthorized use warning, and some nice graphics. You can do this by editing the HTML files in the directory $FWDIR/conf/ahclientd. Remember to leave the % commands intact, as they are used by NGX to insert the information it needs.

Installing the User Database

The Check Point User Database is independent of the SmartDashboard objects and rulebases. When you install a policy (from SmartDashboard, Policy | Install…), in fact the SmartDashboard saves the current policy, and installs the policy (containing the objects and rules) and the user database. If you have made changes only to the user database (password changes, created users, changed group membership), you might want to install only the user database, which is a lot faster.

To install the user database, from the SmartDashboard you can select Policy | Install Database…, or from the Manage | Users and Administrators dialog, click on Actions… and then select Install… as in Figure 9.53.

Figure 9.53 Using Actions in the Manage Users and Administrator Dialog
You will see the *Install Database* window, like in Figure 9.54. You can then select **OK** to install the database. If you have more than one gateway or SmartCenter, you will see different objects and select among them.

**Figure 9.54** Installing the User Database
Summary

Many security rulebases do not have the need for individual user rights, and work with Hosts, Gateways, Networks, Groups, Ranges, and Servers. However, both for security and for tracking purposes you might need to integrate authentication with your security policy. You’ll be able to identify users’ navigation, and grant privileged users access to restricted resources or connections with specific services.

You have a choice of how to recognize a user, accessing external directory servers with RADIUS or TACACS, or using the internal user database. You can integrate with Microsoft Active Directory through the Microsoft Internet Authentication Service, or get the SmartDirectory license for LDAP integration.

You can choose between User, Client, and Session Authentication, depending on your needs and a balance of security, ease-of-use, and flexibility. User authentication is easy to use and transparent, but is not flexible, has no security, and can be cumbersome for accessing external web sites. Client Authentication is flexible and can be secure, but it is not transparent to the user and less secure than other methods. Session Authentication is flexible, secure, and easy to use, but installing the agent on each machine will be something you have to consider.

Check Point NGX gives you many options; you just have to choose which to implement.

Solutions Fast Track

Authentication Overview

- With authentication you can grant specific permissions to groups of users who might have different IPs and be moving around different computers.
- Users in DHCP environments are suitable for implementing authentication, as well as roaming administrators who need to access special files.
- Authentication rules had to be placed carefully, so that a nonauthentication rule does not override the need for a user to authenticate.
Users and Administrators

- Several schemes are available to authenticate users, both from the internal database and from external directories like RADIUS, TACACS, LDAP, SecurID.
- Administrators are created in the SmartDashboard and assign Permissions Profiles to limit their actions within the configuration applications.
- Users are created based on templates, and should be placed in groups to be integrated with the rulebase.

User Authentication

- User Authentication is transparent to the user and doesn’t require configuration of client machines.
- Only the four Authenticated services—HTTP, FTP, Telnet, and RLOGIN—can work with User Authentication.
- User-authenticated HTTP access to the Internet will require users to authenticate multiple times for a single Web page.

Session Authentication

- Session Authentication can authenticate each session from the client, and it can have encryption enabled for security.
- Session Authentication requires a Session Authentication Agent installed in the authorizing machine.
- The Session Authentication Agent can be configured to respond only to certain firewalls and with encryption.

Client Authentication

- Client Authentication works with any defined service available.
- Manual Authentication is performed by an HTTP connection to the port 900 or Telnet to port 259.
- Other sign-on methods integrate User, Session, and SSO Authentication into Client Authentication.
Frequently Asked Questions

The following Frequently Asked Questions, answered by the authors of this book, are designed to both measure your understanding of the concepts presented in this chapter and to assist you with real-life implementation of these concepts. To have your questions about this chapter answered by the author, browse to www.syngress.com/solutions and click on the “Ask the Author” form.

Q: How can I enter a single user in a rule instead of a group?
A: You can enter only groups in a rule. However, you can create a group that contains only one user.

Q: Can I use Check Point NGX as a Web proxy?
A: Yes, you can. However, NGX does not store in memory frequently accessed pages, so the connections will not be accelerated. These connections will use the HTTP Security Server for additional protection, and you need to enable the http_connection_method_proxy property using dbedit or GUIDBedit.

Q: Which authentication should I use?
A: For HTTP intranet access, try User Authentication. For accessing the Internet, try encrypted Manual Client Authentication. Try redirecting nonallowed traffic to the Client Authentication page using a resource.

Q: We came back from our New Year’s celebration and no one could authenticate to the firewall. What can I do?
A: Check the expiration date of the users.

Q: I would like to give external access to an internal Web site. What authentication method should I use?
A: I would recommend user authentication, since you have a controlled environment (your Web site) and you should authenticate every session, for increased security.
Chapter 9 • Authentication

Q: When trying User Authentication, users received the error “FW-1 (password) Reason FW-1 Rule.” The rulebase is configured, the user is configured, and the traffic should be accepted.

A: Edit the properties of the User Auth action, and make sure the HTTP property is set to All Servers.

Q: Is there a limit to the numbers of users available in the internal Check Point database?

A: There is no set limit for the Check Point user database. However, if you have a large number of users, integrating with an external user directory will be easier to manage.

Q: A user cannot authenticate with his password, after he recently changed it. In the SmartView Tracker is, there an error regarding a URL worm?

A: HTTP Passwords are obscured when transmitted, and the new obscured password could have a pattern similar to a known URL worm, and could be blocked by Web Intelligence’s General HTTP Worm Catcher feature. You can either remove the pattern being enforced, or (new for NGX) configure the General HTTP Worm Catcher to apply only to selected Web servers.

Q: In a thin client environment (Citrix, Terminal Services), which authentication method should I use?

A: In a thin client environment you should configure User Authentication. Both client and session authentication will not be able to authenticate a specific user’s connection.

Q: Is there a Session Authentication Agent for non-Windows systems?

A: There is no Check Point-provided nor supported Session Authentication Agent for non-Windows systems. However, you can find an agent written in Perl on the user-supported sites like www.cpug.org.

Q: Is there a way to show users a message when they need to first use Client Authentication, or to redirect them to the authentication page?

A: You can create a URI resource that will match all pages, and has a Replacement URI pointing to the authentication page or the message page.
Create a rule among the last in the rulebase, with the HTTP service using the created resource and a drop action.

Q: A user cannot authenticate when connected from Network X, only when connected from Network Y. Other users cannot authenticate at all. What should I check?

A: Verify each users' location properties. Remember that there needs to be an overlap of the location defined in the user properties and the location (source or destination) defined in the rulebase. Try to edit the Action properties to ignore the user database, to see if that is the source of the problem.