Check Point UserAuthority Guide

Version NGX R61



700358 January 2006

© 2003-2006 Check Point Software Technologies Ltd.

All rights reserved. This product and related documentation are protected by copyright and distributed under licensing restricting their use, copying, distribution, and decompilation. No part of this product or related documentation may be reproduced in any form or by any means without prior written authorization of Check Point. While every precaution has been taken in the preparation of this book, Check Point assumes no responsibility for errors or omissions. This publication and features described herein are subject to change without notice.

RESTRICTED RIGHTS LEGEND:

Use, duplication, or disclosure by the government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013 and FAR 52.227-19.

TRADEMARKS:

©2003-2006 Check Point Software Technologies Ltd. All rights reserved.

Check Point, Application Intelligence, Check Point Express, the Check Point logo, AlertAdvisor, ClusterXL, Cooperative Enforcement, ConnectControl, Connectra, CoSa, Cooperative Security Alliance, Eventia, Eventia Analyzer, FireWall-1, FireWall-1 GX, FireWall-1 SecureServer, FloodGate-1, Hacker ID, IMsecure, INSPECT, INSPECT XL, Integrity, InterSpect, IQ Engine, Open Security Extension, OPSEC, Policy Lifecycle Management, Provider-1, Safe@Home, Safe@Office, SecureClient, SecureKnowledge, SecurePlatform, SecuRemote, SecureXL Turbocard, SecureServer, SecureUpdate, SecureXL, SiteManager-1, SmartCenter, SmartCenter Pro, Smarter Security, SmartDashboard, SmartDefense, SmartLSM, SmartMap, SmartUpdate, SmartView, SmartView Monitor, SmartView Reporter, SmartView Status, SmartViewTracker, SofaWare, SSL Network Extender, Stateful Clustering, TrueVector, Turbocard, UAM, User-to-Address Mapping, UserAuthority, VPN-1, VPN-1 Accelerator Card, VPN-1 Edge, VPN-1 Pro, VPN-1 SecureClient, VPN-1 SecuRemote, VPN-1 SecureServer, VPN-1 VSX, VPN-1 XL, Web Intelligence, ZoneAlarm, ZoneAlarm Pro, Zone Labs, and the Zone Labs logo, are trademarks or registered trademarks of Check Point Software Technologies Ltd. or its affiliates. All other product names mentioned herein are trademarks or registered trademarks of their respective owners. The products described in this document are protected by U.S. Patent No. 5,606,668, 5,835,726, 6,496,935 and 6,850,943 and may be protected by other U.S. Patents, foreign patents, or pending applications.

For third party notices, see "THIRD PARTY TRADEMARKS AND COPYRIGHTS" on page 173.

Contents

Preface

Who Should Use This Guide 10
Summary of Contents 11
Appendices 12
Related Documentation 13
More Information 16

Chapter 1 **Introduction**

The Need for UserAuthority 18
Identity-based Access Control for Outbound Connections via VPN-1 Pro Gateway ... 19
Underlying Concept and Advantage 20
Typical Deployment 21
UserAuthority SSO for VPN-1 Pro Deployment 21
OPSEC Protocols 23
How to Use this Guide 24

Chapter 2 **UserAuthority Deployments and Installation**

Overview 26
Deployments 27
Outbound Access Control 27
Citrix MetaFrame or Windows Terminal Services 32
Installation and Configuration 35
Installing and Configuring UAS on VPN-1 Pro 35
Installing and Configuring the UAS on the Windows DC 47

Chapter 3 **Outbound Access Control**

The Challenge 58
The UserAuthority Solution 59
Identification using SecureAgent 61
Identity Sharing 61
Retrieving Windows Groups with UserAuthority 66
Outbound Access Control using Citrix Terminals as TIP 67
Scenario - An Organization using Multiple Windows DCs 68
Scenario - An Organization Using Multiple Domains 70
Configurations 72
Adding Additional Windows DCs 72
Outbound Access Control on Citrix or Windows Terminals 73
Configuring UserAuthority Domain Equality 73

Chapter 4 **User Management in UserAuthority**

Overview 78

Managing Users and Groups 79
Users in UserAuthority 79
User Groups in UserAuthority 79
Using a Local Check Point Database 81
Using an External Database 82
Using the Windows User Identity 83
Users in the Windows Domain 83
Configuring UserAuthority to Recognize Windows User Groups 83

Chapter 5 **Auditing in UserAuthority**
Overview 86
Using Logs for Auditing 87
Auditing Outbound Traffic Using UserAuthority Outbound Access Control 88
Configuring UserAuthority for Auditing 92
Configuring Auditing of Requests for External Resources 92

Chapter 6 **High Availability and Load Balancing**
Overview 94
High Availability 94
Load Balancing 94
High Availability and Load Balancing in UserAuthority 95
Using Multiple Windows DCs 96
Using a VPN-1 Pro Cluster 97
Using VPN-1 Pro Clusters 97
Synchronizing the Credentials Manager 97

Chapter 7 **UserAuthority CLIs**

Chapter 8 **UserAuthority OPSEC APIs**
Overview 110
Programming Model 111
Defining a UAA Client 114
Client Server Configuration 114
OPSEC UserAuthority API Overview 114
Function Calls 125
Session Management 125
Assertions Management 126
Managing Queries 129
Managing Updates 130
Managing Authentication Requests 131
Assertions Iteration 132
Managing UAA Errors 134
Debugging 135
Event Handlers 136
UAA_QUERY_REPLY Event Handler 136
UAA_UPDATE_REPLY Event Handler 137
UAA_AUTHENTICATE_REPLY Event Handler 138

Chapter 9 **Monitoring the UserAuthority Environment**
Overview 142
System Monitoring 143
Monitoring the System Status 143
User Monitoring 148
Monitoring User Activities 148
Monitoring Example: SecureAgent Cannot Provide User Identity 149

Chapter 10 **Troubleshooting UserAuthority**
Overview 152
General Problems 153
Why is there no established SIC? 153
Why are Domain Controller Queries not Sent Properly? 156
User-Related Problems 157
Why does SecureAgent not identify the user? 157
Why are Terminal Server Clients not Identified by UAS? 160
Why does the Firewall Report Identify Users as Unknown? 161

Appendix A **Integrating UserAuthority with Meta IP**
Overview 164
Required Components 165
Preliminary Steps 166
Windows DC Configuration 167
VPN-1 Pro Policy Configuration 168
DHCP Server Configuration 170

Appendix B **Glossary**
Acronyms and Abbreviations 176

Index 181

Preface

In This Chapter

Who Should Use This Guide page 6
Summary of Contents page 7
Related Documentation page 9
More Information page 12

Who Should Use This Guide

This guide is intended for administrators responsible for maintaining network security within an enterprise, including policy management and user support.

This guide assumes a basic understanding of

- System administration.
- The underlying operating system.
- Internet protocols (IP, TCP, UDP etc.).

Summary of Contents

This guide provides step-by-step instructions for configuring UserAuthority.

In order to assist you in the deployment of UserAuthority, this guide contains various scenarios that suit the deployments of most enterprises. These scenarios are followed by detailed workflows that can be used to help with your deployment. You can also combine the deployments and workflows described in this guide to best suit the deployment in your enterprise.

Table A-1

Chapter	Description
Chapter 1, “Introduction”	describes the User Authority concept, deployment and management solution.
Chapter 2, “UserAuthority Deployments and Installation”	provides the foundation for the deployment of UserAuthority in its most basic form
Chapter 3, “Outbound Access Control”	describes UserAuthority’s part in access to external resources.
Chapter 4, “User Management in UserAuthority”	provides information about managing users and groups with a Check Point database and external databases.
Chapter 5, “Auditing in UserAuthority”	explains how UserAuthority uses the SmartView Tracker, Check Point's advanced tracking tool, to enable auditing of both UserAuthority Server (UAS).
Chapter 6, “High Availability and Load Balancing”	describes how the UserAuthority Server (UAS) can be configured to provide both high availability and load balancing.
Chapter 7, “UserAuthority CLIs”	explains the UserAuthority command line interfaces.
Chapter 8, “UserAuthority OPSEC APIs”	describes OPSEC APIs
Chapter 9, “Monitoring the UserAuthority Environment”	describes how system and user monitoring allows the system administrator to view the system status for debugging and problem solving in the system.
Chapter 10, “Troubleshooting UserAuthority”	provides help for common problems that might arise when using UserAuthority.

Appendices

This guide contains the following appendices**:**

Table A-2

Appendix	Description
Appendix A, “Integrating UserAuthority with Meta IP”	explains how UserAuthority can easily be integrated with the Meta IP product to provide authenticated IP addresses from an authenticated IP pool to authenticated users.
Appendix B, “Glossary”	describes the acronyms and abbreviations used in this guide.

Related Documentation

The NGX R61 release includes the following documentation

TABLE P-1 VPN-1 Pro documentation suite documentation

Title	Description
Getting Started Guide	Contains an overview of NGX R61 and step by step product installation and upgrade procedures. This document also provides information about What's New, Licenses, Minimum hardware and software requirements, etc.
Upgrade Guide	Explains all available upgrade paths for Check Point products from VPN-1/FireWall-1 NG forward. This guide is specifically geared towards upgrading to NGX R61.
SmartCenter Guide	Explains SmartCenter Management solutions. This guide provides solutions for control over configuring, managing, and monitoring security deployments at the perimeter, inside the network, at all user endpoints.
Firewall and SmartDefense Guide	Describes how to control and secure network access; establish network connectivity; use SmartDefense to protect against network and application level attacks; use Web Intelligence to protect web servers and applications; the integrated web security capabilities; use Content Vectoring Protocol (CVP) applications for anti-virus protection, and URL Filtering (UFP) applications for limiting access to web sites; secure VoIP traffic
Eventia Reporter	Explains how to monitor and audit traffic, and generate detailed or summarized reports in the format of your choice (list, vertical bar, pie chart etc.) for all events logged by Check Point VPN-1 Pro, SecureClient and SmartDefense.

TABLE P-1 VPN-1 Pro documentation suite documentation (continued)

Title	Description
SmartView Tracker Guide.	Provides information about how to collect comprehensive information on your network activity in the form of logs. Learn how to use SmartView Tracker to audit these logs at any given time, analyze traffic patterns and troubleshoot networking and security issues.
SecurePlatform Guide	Explains how to install and configure SecurePlatform. This guide will also teach you how to manage your SecurePlatform and explains Dynamic Routing (Unicast and Multicast) protocols.
Provider-1 Guide	Explains the Provider-1/SiteManager-1 security management solution. This guide provides details about a three-tier, multi-policy management architecture and a host of Network Operating Center oriented features that automate time-consuming repetitive tasks common in Network Operating Center environments.

TABLE P-2 Integrity Server documentation

Title	Description
Integrity Advanced Server Installation Guide	Explains how to install, configure, and maintain the Integrity Advanced Server.
Integrity Advanced Server Administrator Console Reference	Provides screen-by-screen descriptions of user interface elements, with cross-references to relevant chapters of the Administrator Guide. This document contains an overview of Administrator Console navigation, including use of the help system.
Integrity Advanced Server Administrator Guide	Explains how to managing administrators and endpoint security with Integrity Advanced Server.
Integrity Advanced Server Gateway Integration Guide	Provides information about how to integrating your Virtual Private Network gateway device with Integrity Advanced Server. This guide also contains information regarding deploying the unified SecureClient/Integrity client package.

TABLE P-2 Integrity Server documentation (continued)

Title	Description
Integrity Advanced Server System Requirements	Provides information about client and server requirements.
Integrity Agent for Linux Installation and Configuration Guide	Explains how to install and configure Integrity Agent for Linux.
Integrity XML Policy Reference Guide	Provides the contents of Integrity client XML policy files.
Integrity Client Management Guide	Explains how to use of command line parameters to control Integrity client installer behavior and post-installation behavior.

More Information

- For additional technical information about Check Point products, consult Check Point's SecureKnowledge at https://secureknowledge.checkpoint.com/.



- See the latest version of this document in the User Center at http://www.checkpoint.com/support/technical/documents

Chapter 1
Introduction

In This Chapter

The Need for UserAuthority page 14
Underlying Concept and Advantage page 16
Typical Deployment page 17
OPSEC Protocols page 19
How to Use this Guide page 20

The Need for UserAuthority

In today's business environment, enterprises need to provide employees, partners and customers with the ability to access and work with many different applications and services. It is important that access to these applications be simple and convenient, and, at the same time, secure, reliable, and easy to manage. UserAuthority is able to leverage the security needs of your existing or new environment to higher levels.

UserAuthority can improve access control management in your enterprise with identity-based access control for outbound connections via the VPN-1 Pro gateway.

Identity-based Access Control for Outbound Connections via VPN-1 Pro Gateway

UserAuthority can provide access control to external resources at the network level (Internet or other services outside the perimeter gateway). Through VPN-1 Pro gateways, firewall authentication can be configured in the security policy to supply such demand (Client, Session authentications). The major difference with UserAuthority is the benefit of SSO to those authentications, eliminating the need for the user to re-authenticate. UserAuthority enables the user to be identified transparently via the gateway without human intervention. This functionality is also known as UserAuthority SSO for VPN-1 Pro or Outbound SSO.

Underlying Concept and Advantage

One of the greatest advantages of UserAuthority is its ability to extract the user identity from a Trusted Identification Point (TIP). UserAuthority establishes a trust relationship with TIPs on the network to ensure that it is receiving trusted information.

UserAuthority TIPs include:

- Windows' logons to Domain Controllers
- VPN-1 Pro authentication (SecureRemote/SecureClient) or any other authentications to the gateways)
- MS Terminal Services/Citrix MetaFrame servers

Extracting the user identity from the TIP enables the following benefits:

- Once a user is logged on to the system and identified by UserAuthority, there is no need to authenticate again, even when accessing a Web application.
- Pure SSO, requiring only the initial network log on to a TIP. No other authentication is required.
- Utilization of existing authentication in the network environment to retrieve user identification, without requiring the end user to identify to an additional identification mechanism.
- Integration of network level authentication with Web applications.
- Deployment does not require any changes to Web applications.

Typical Deployment

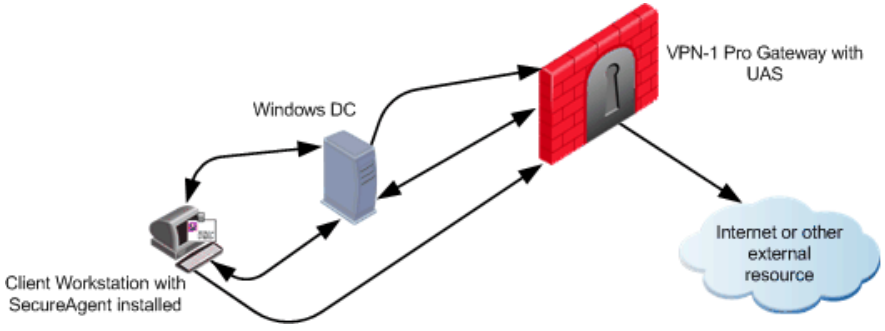
This section describes three common types of deployments, and the particular benefits of integrating UserAuthority into each of the deployment types. A detailed description of the various UserAuthority deployment types, and how they are set up and implemented, is presented in Chapter 2, “UserAuthority Deployments and Installation””.

The following example illustrates identity-based access control for outbound connections via a VPN-1 Pro gateway.

UserAuthority SSO for VPN-1 Pro Deployment

UserAuthority can provide authorization to external resources at the network level. Most enterprises already use VPN-1 Pro authentication rules that require client or session authentication to external resources. UserAuthority expands on this by providing SSO to the VPN-1 Pro as well as auditing capabilities.

Figure 1-1 SSO for VPN-1 Pro Deployment



UserAuthority eliminates the need for a user to authenticate each time an external resource is accessed. This is done by using the information on the Windows DC to identify the user. When the user requests an external resource, the UserAuthority Server on the VPN-1 Pro gateway queries the UserAuthority Server installed in a Windows DC. The UserAuthority Server on the Windows DC sends a query to a desktop application called SmartAgent, which identifies the user according to the Windows DC identification that was used at sign-on.

This information is sent back to the UserAuthority Server on the VPN-1 Pro gateway to provide authentication on behalf of the user. In this way, the user is automatically authenticated each time without the need to re-authenticate each time a request for external resources is made. This scenario is illustrated in Figure 1-1.

UserAuthority can be also configured to create logs each time a user requests an external resource. This provides information on how users are accessing external resources. Logs can provide various types of information, such as whether users are violating enterprise policy or whether there are communications problems when trying to access external resources.

UserAuthority extends the capabilities of VPN-1 Pro authentication by providing SSO, which eliminates the need for users to authenticate to VPN-1 Pro and provides auditing capabilities for requests to external resources. For more information, see Chapter 3, “Outbound Access Control””.

OPSEC Protocols

UserAuthority supports all Check Point Open Platform for Security (OPSEC) standards. OPSEC provides a single integration framework by using the OPSEC Software Development Kit (SDK) for integration with Check Point VPN-1 Pro. OPSEC APIs provide solutions for third-party and in-house integration.

The UAA (UserAuthority) API set can be used to create a single authorization solution for any application. For example, an enterprise might want to use a single user identification for applications that are not Web-based (such as a client installation) in addition to their Web applications. The UAA OPSEC API enables the integration of any application that requires authentication and authorization, and provides all UserAuthority benefits to the application.

Integration can be easily programmed by in-house programmers using the OPSEC APIs. In addition, it is possible to turn to an OPSEC partner to develop a solution for the enterprise. OPSEC partners are a group of professional programmers who use the OPSEC standard.

For information on the OPSEC UAA API set, see Chapter 8, “UserAuthority OPSEC APIs””.

How to Use this Guide

This guide provides step-by-step instructions for configuring UserAuthority.

In order to assist you in the deployment of UserAuthority, this guide contains various scenarios that suit the deployments of most enterprises. These scenarios are followed by detailed workflows that can be used to help with your deployment. You can also combine the deployments and workflows described in this guide to best suit the deployment in your enterprise.

Please note that Chapter 2 provides the foundation for the deployment of UserAuthority in its most basic form. Subsequent chapters elaborate on these deployments. In addition some configurations have been excluded from these deployments. These configurations can easily be added once your network has been deployed with User Authority.

Chapter 2

UserAuthority Deployments and Installation

In This Chapter

Overview page 22
Deployments page 23
Installation and Configuration page 31

Overview

This chapter describes typical UserAuthority deployments and how to install and configure the UserAuthority Server (UAS) used in the deployments.

The following deployments are described in this chapter:

- **Outbound Access Control**. This deployment is used to provide authorization of users when they access external resources and for monitoring users' requests to access external resources. In this deployment, an administrator defines rules that allow users on an internal network to access external systems (for example, Internet or external subnets) without having to repeatedly authenticate to the VPN-1 Pro gateway. In other words, UserAuthority is configured to eliminate the need to authenticate to VPN-1 Pro each time a request for an external resource is made. In addition, each time a request to access an external resource is made, a log entry is created. The administrator can configure UserAuthority to make these logs available, so the administrator can view a list of user activities. For more information, see Chapter 3, "Outbound Access Control".
- **UserAuthority installed on Citrix MetaFrame or Windows Terminal Services**. This deployment also provides user authorization, auditing and Web SSO. The main difference between this deployment and the Enterprise with Web Applications deployment is that the client computers are connected to a Citrix MetaFrame or Windows Terminal Services. In this case, all users access applications from the same source (the terminal), which has only one IP address. UserAuthority uses port information to get the user identity in order to authorize and/or authenticate the user.

Although each of these deployments can adequately serve an enterprise, it is possible to combine them to create the deployment that best fits the enterprise's network. The deployments described in this chapter are presented as follows:

- a general workflow for each process is described;
- the necessary components for the deployment are given;
- detailed step-by-step procedures are then described.

This chapter also explains how to carry out the basic installations and configurations for the UAS, and other components that are necessary to carry out the deployments described in this chapter. The configurations described are the simplest configurations necessary to deploy UserAuthority. In most cases, additional configuration is not required, however, in complex networks, more advanced configurations are possible. These configurations are described in later chapters of this book.

Deployments

In This Section

Outbound Access Control *page 23*
Citrix MetaFrame or Windows Terminal Services *page 28*

This section presents some typical deployments to assist a network administrator in determining the most suitable type of deployment for the enterprise's network. This section also describes how the elements in each deployment complement one another and how they can be combined.

Outbound Access Control

Outbound Access Control deployment is used to provide authorization and auditing for users accessing external resources. When clients access the Internet from inside a local network, UserAuthority captures authentication information from a TIP (for example, VPN-1 Pro, Windows DC), which eliminates the need to authenticate to VPN-1 Pro in order to achieve identity-level authorization and auditing.

Outbound Access Control deployment provides:

- "Single Sign-On to VPN-1 Pro" for local clients by eliminating the need to authenticate each time the user goes through VPN-1 Pro
- Auditing capabilities by providing a log of each user request to an external resource
- Authorization capabilities

The following components are required for the deployment:

- UAS installed on the VPN-1 Pro module.
- UAS installed on at least one Windows DC.
- VPN-1 Pro management installed on a gateway or other server.
- SecureAgent installed on each client. This installation is performed automatically when a client signs on to the Windows Domain.

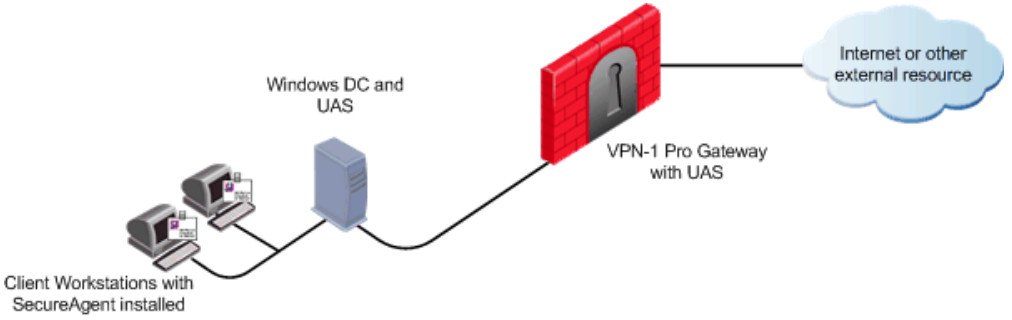
For information on installing the various components, see "Workflow" on page 24.

For more information on Outbound Access Control, see Chapter 3, "Outbound Access Control".

For information on installing VPN-1 Pro, the management applications, or SmartDashboard, see the *Check Point SmartCenter Guide.*

Figure 2-1 shows a deployment that provides Outbound Access Control.

Figure 2-1 Outbound Access Control Deployment



In this deployment, the following takes place:

1. The user signs on to the Windows DC, and logs into the client host.
2. When the user accesses an external resource for the first time, the VPN-1 Pro module queries the user identity through the UAS on the module.
3. The query is then forwarded to the UAS on the Windows DC.
4. The UAS on the Windows DC checks the client credentials through the SecureAgent module on the client desktop.

For more information about Single Sign-On for VPN-1 Pro, see Chapter 3, "Outbound Access Control".

Workflow

To carry out the deployment:

1. Install the UAS on the machine with the VPN-1 Pro gateway (see "Installing and Configuring UAS on VPN-1 Pro" on page 31).
2. Install the UAS on the Windows DC (see "Installing and Configuring the UAS on the Windows DC" on page 43).
3. Configure the system to automatically install SecureAgent (see "Configuring SecureAgent Automatic Installation" on page 50).
4. From the SmartDashboard **Security** tab, configure an SSO rule (see "Adding an SSO Rule" on page 25).

Test Your Deployment

- Try to access an external resource. Make sure that you can enter the resource without getting an authentication request from the VPN-1 Pro.

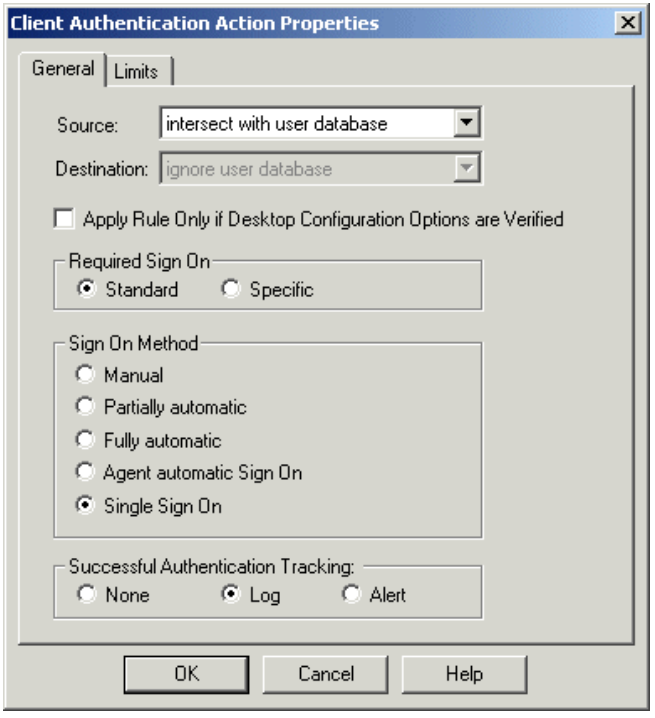
Adding an SSO Rule

In this deployment, you must establish SSO for VPN-1 Pro users accessing external resources. This section describes how to configure an SSO rule. This configuration is carried out in the SmartDashboard. For more information on using SmartDashboard, see the *Check Point SmartCenter Guide*.

To create an SSO rule:

1. From SmartDashboard, click the **Security** tab.
2. Click the **Add Rule** button in the tool bar to add a blank rule line.
3. In the new rule, right click the **Source** field to add a source. Click **Add Users Access** and select the User's Group that you want to use for this rule. For a basic SSO rule, you can keep the **Any** default.
4. Right click the **Destination** field, and add a destination. This is the destination to which the rule will apply. For a basic SSO rule, you can keep the **Any** default.
5. Right click the **VPN** field to enter the VPN match conditions. For a basic SSO rule, you can keep the **Any Traffic** default.
6. Right click the **Service** field to determine the types of services that apply to this rule. For a basic SSO rule, you can keep the **Any** default.
7. Right click the **Action** field and then click **Client Auth** from the menu to create SSO for this deployment.
8. Double click the **Action** field to display the **Client Authentication Action Properties** window.

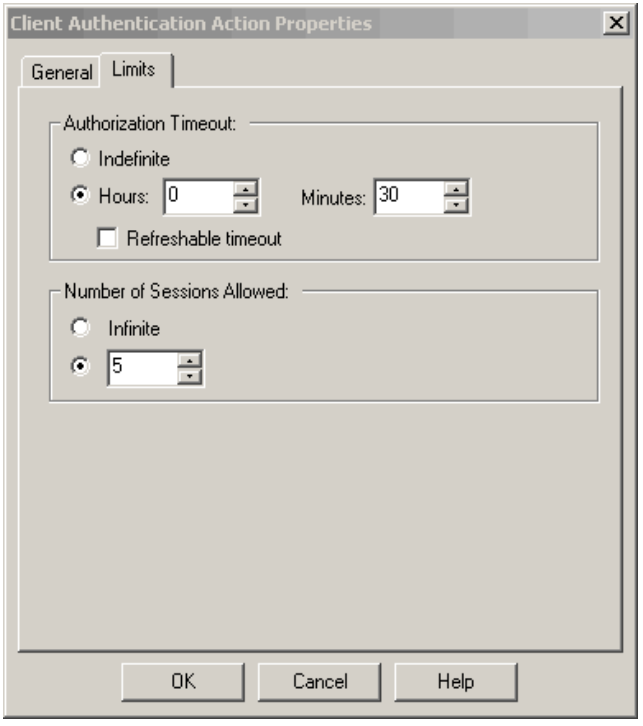
Figure 2-2 Client Authentication Action Properties Window - General Tab



9. In the **Sign On Method** area, click **Single Sign On**.
10. Click the **Limits** tab and set the timeout to determine how long a session lasts.

    It is recommended to keep the default timeout limit of 30 minutes. If you do not want UserAuthority to count the time that a user is working, select the **Refreshable timeout** checkbox.

Figure 2-3 Client Authentication Action Properties Window - Limits Tab



11. In the **Number of Sessions Allowed** area, set the number of connections that can be made before querying for user identity.

    It is recommended to enter `1` for security reasons, however some Web sites that use HTTP 1.0 protocol count sessions for each link that is clicked, therefore it may be best to use a higher number to save system resources.

12. Click **OK** to close the window and return to the SmartDashboard **Security** tab.

13. In the **Security** tab, right click the **Track** field to select how you want to keep track of user requests in the system. It is recommended to select **Log** to provide auditing capabilities.

14. In the **Security** tab, right click the **Install on** field and select **Add** from the drop-down menu, and select the location where the policy is installed. For a basic SSO rule, you can keep the **Policy Targets** default.

15. Click **Install** on the toolbar to install the policy.

The following is an example of an SSO policy in the SmartDashboard:

Figure 2-4 Basic SSO Rule

NO.	SOURCE	DESTINATION	VPN	SERVICE	ACTION	TRACK	INSTALL ON
1	* Any	* Any	* Any Traffic	* Any	Client Auth	Log	* Policy Targets

Citrix MetaFrame or Windows Terminal Services

This deployment is intended for networks where the local host clients are, or include, Citrix MetaFrame Server or Windows Terminal Services. This deployment provides authorization and auditing capabilities for the users signing on to a Citrix or Windows terminal. In this deployment, the UAS is installed on the MetaFrame Server or Terminal Services. UAS on the Terminal Services identifies the user for each outbound request from the server. This can be used for auditing and authorization. This deployment can be used by any of the enterprises listed in the deployments described in this chapter.

The following components are required for this deployment:

- UAS installed on the VPN-1 Pro module
- UAS installed on the Citrix MetaFrame Server or Terminal Services
- VPN-1 Pro management

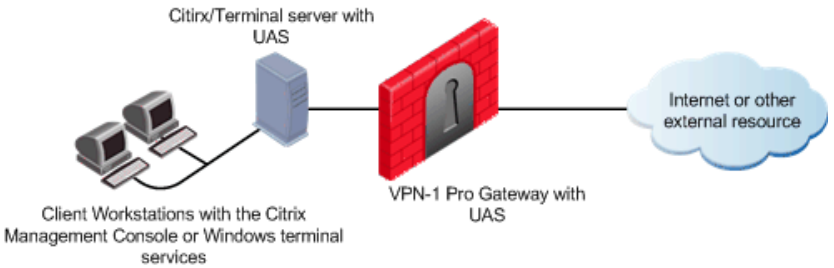
For information on installing the various components see "Workflow" on page 29.

For more information on Outbound Access Control, see Chapter 3, "Outbound Access Control".

For information on installing VPN-1 Pro, the management applications, or SmartDashboard, see the *Check Point SmartCenter Guide.*

Figure 2-5 shows UserAuthority deployed in a Citrix or Windows Terminal Services system.

Figure 2-5 Citrix MetaFrame or Windows Terminal Services Deployment



In this deployment:

1. The user signs on to the Citrix MetaFrame Server or the Terminal Services, and logs into the client host.
2. When the user accesses an external resource for the first time, the VPN-1 Pro module queries for the user identity through the UAS on the module.
3. The query is then forwarded to UAS on the Citrix MetaFrame Server or the Terminal Services. The user is identified and the identification information is forwarded to VPN-1 Pro to authorize and audit the request.

Workflow

To carry out the deployment:

1. Install the UAS on the machine with the VPN-1 Pro gateway (see "Installing and Configuring UAS on VPN-1 Pro" on page 31).
2. Install the UAS on the Citrix MetaFrame Server or Terminal Services (see "Installing and Configuring the UAS on the Windows DC" on page 43).
3. From the SmartDashboard **Security** tab, configure an SSO rule (see "Adding an SSO Rule for Citrix MetaFrame or Windows Terminal Services" on page 29).
4. Save the policy in SmartDashboard and install the firewall policy on the VPN-1 Pro gateway where UserAuthority installed.

Test Your Deployment

Try to get an external resource. Attempt to enter the resource without getting an authentication request from the VPN-1 Pro.

Adding an SSO Rule for Citrix MetaFrame or Windows Terminal Services

An SSO rule for Citrix MetaFrame or Windows Terminal Service is created in the same way as for Outbound Access Control, except that the SSO rule must be applied through session authentication instead of client authentication. This is because the browser and other applications are on the server and many different clients may be using them.

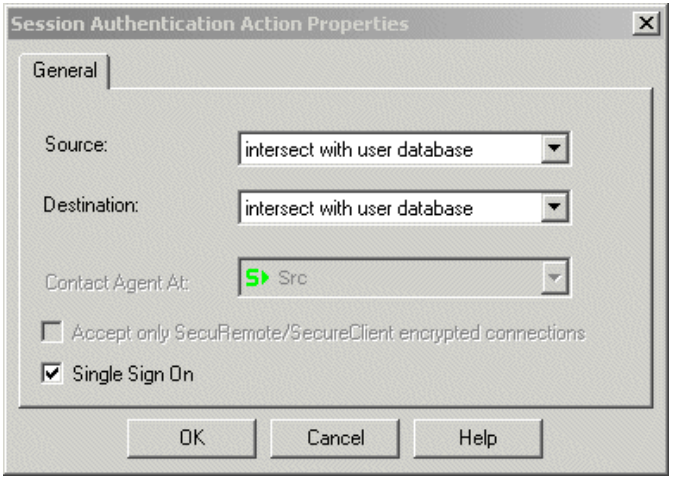
This section describes how to configure an SSO rule. This configuration is carried out in the SmartDashboard. For more information on using SmartDashboard see the *Check Point SmartCenter Guide*.

To create an SSO rule:

1. From SmartDashboard, click the **Security** tab.

2. Click the **Add Rule** button in the tool bar to add a blank rule line.
3. In the new rule, right click the **Source** field to add a source. For a basic SSO rule, you can keep the **Any** default.
4. Right click the **Destination** field, and add a destination. This is the destination to which the rule will apply. For a basic SSO rule, you can keep the **Any** default.
5. Right click the **VPN** field to enter the VPN match conditions. For a basic SSO rule, you can keep the **Any Traffic** default.
6. Right click the **Service** field to determine the types of services that apply to this rule. For a basic SSO rule, you can keep the **Any** default.
7. Right click the **Action** field and then click **Session Auth** from the menu *to create SSO for this deployment*.
8. Double click the **Action** field to display the **Session Authentication Action Properties** window.

Figure 2-6 Session Authentication Action Properties Window



9. Select the **Single Sign On** checkbox.
10. Click **OK** to close the window and return to the SmartDashboard **Security** tab.
11. Right click the **Track** field in the rule line to select how you want to keep track of user requests in the system. It is recommended to select **Log** to provide auditing capabilities.
12. Right click the **Install on** field in the rule line and from the **Add** the drop-down menu, select where the policy is installed. For a basic SSO rule, you can keep the **Policy Targets** default.
13. Click **Install** on the toolbar to install the policy.

Installation and Configuration

In This Section

Installing and Configuring UAS on VPN-1 Pro *page 31*
Installing and Configuring the UAS on the Windows DC *page 43*

This section provides step-by-step directions for the installations and configurations necessary to deploy UserAuthority.

Installing and Configuring UAS on VPN-1 Pro

The following components are required to install the UAS on the firewall gateway:

- VPN-1 Pro module installed on a gateway or other server
- VPN-1 Pro management installed on a gateway or other server
- SmartDashboard

For information on how to use and install these products, see the appropriate Check Point user guide.

The installation process comprises the following steps:

- Install the UserAuthority License
- Install the UAS software on the VPN-1 Pro gateway
- Configure the UAS
- Configure UAS domain equality

Installing the UserAuthority License

UserAuthority requires a license per client (user), not per server. You can retrieve a license from the Check Point User Center at www.checkpoint.com/usercenter after the software is purchased. Licences can be stored and maintained in the SmartUpdate repository. For more information on SmartUpdate, see the *Check Point SmartCenter Guide*.

Licenses created in the Check Point User Center include:

- **IP address**: IP address of the computer for which the license is intended.
- **Certificate Key:** A string of twelve alphanumeric characters.
- **Expiration date**

- **SKU/Features**: The character string that defines an individual license. The string for UserAuthority is:

  CPUA-UAU-*-NG, where * is the number of licenses (i.e., the number of users).

The license can be installed using the Check Point Configuration tool. The validation code supplied by the Check Point User Center should be compared with the validation code calculated in the Check Point Configuration Tool. These strings should be identical.

For information on using the Check Point Configuration tool to install a license, see the *Check Point SmartCenter Guide*.

Installing UAS on the VPN-1 Pro Gateway

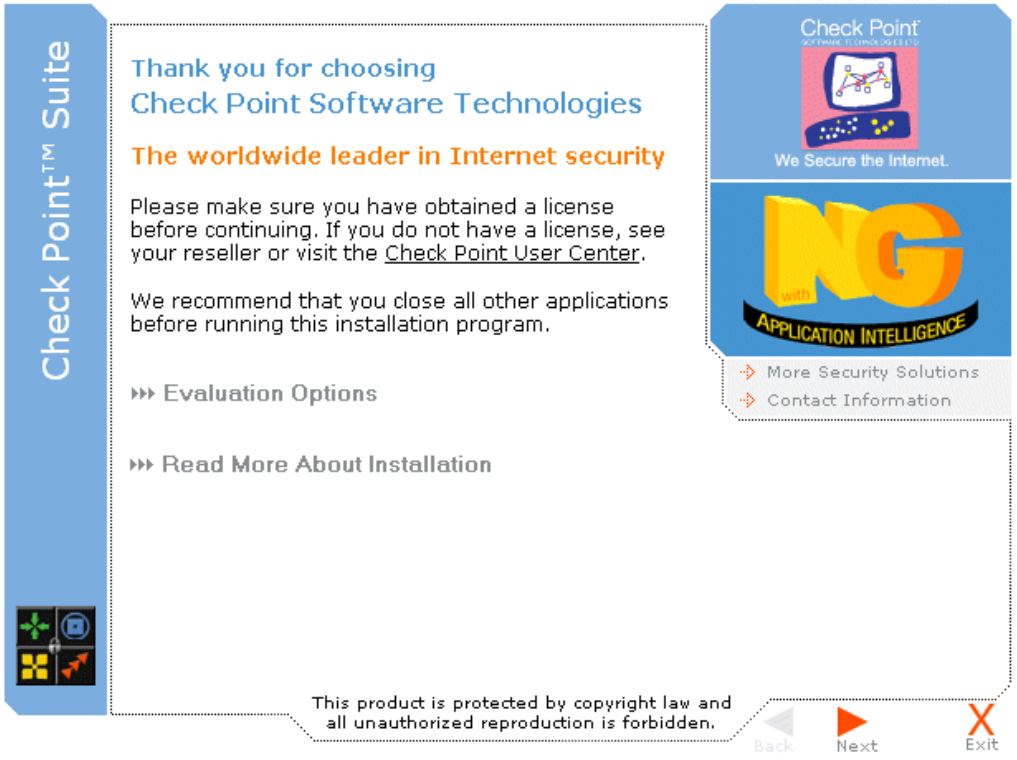
Windows

Before installing the UAS, be sure that SVN Foundation and VPN-1 Pro are installed. If they are not installed, see the instructions in the *Check Point SmartCenter Guide*.

To install UAS on a Windows gateway:

1. Insert the Wrapper CD and then run the Wrapper. The Installation **Welcome** window is displayed.

Figure 2-7 Installation Welcome Window



2. Click **Next** to display the End-Users License Agreement (EULA).

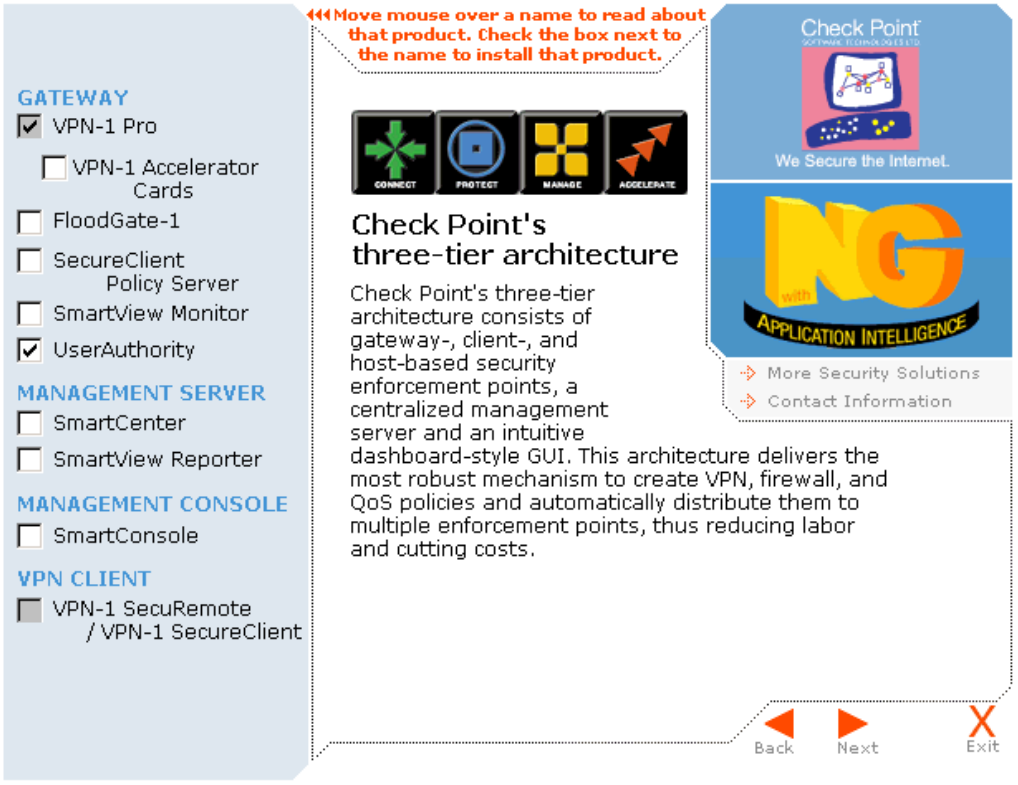
Figure 2-8 End Users License Agreement



3. Read the End-Users License Agreement (EULA) and then click **Yes** to accept it. The next installation window is displayed.
4. Select **Check Point Enterprise** for the type of installation, and then click **Next**. The next installation window is displayed.
5. Select **UserAuthority** from the list of CheckPoint products.

Note - If the VPN-1 Pro module and other gateway components are not installed, you can install them at the same time by selecting them in the Product Selection list. If already installed, the checkbox is selected and grayed as shown in FIGURE 1-16.

Figure 2-9 Product Selection



6. Click **Next** to start the **Install Shield** and follow the on-screen instructions.
7. Browse to a folder where you want to install **UserAuthority**, or click **Next** to install in the default folder.
8. At the end of the installation, click **OK.**
9. If VPN-1 Pro is already installed on the machine, then this is the end of the installation. Restart your computer to finish the installation. After the restart, you must add the UserAuthority license (see "Installing the UserAuthority License" on page 31).

   OR,

   If VPN-1 Pro is not installed, the **License** window is displayed.

   If your license is not listed in the window, you must install a license to continue (see "Installing the UserAuthority License" on page 31).

10. Click **Next**. If there are no other Check Point installations on the computer, you must enter information in the **Key Hit Session** and the **Secure Internal Communication** (SIC) windows. If other applications are already installed, skip to step 11 on page 36.

    a. Click **Next**, if there are no other Check Point installations on the computer, the **Key Hit Session** window is displayed. Follow the directions in the window and then click **Next**.

    b. The **Secure Internal Communication** window is displayed. Enter a password key in the **Activation Key** field and then enter it again in the **Confirm Activation** Key field to confirm it. *Be sure to remember your key, you need to enter it in the SmartDashboard configuration.*



Note - If you have already installed VPN-1 Pro, you do not need to configure the Key Hit session or SIC. If these windows are displayed on the computer, skip these steps.

11. Click **Finish**. The **Thank you for using** message is displayed.
12. Click **OK**.
13. Remove the CD and then click **Finish** to restart the computer.

UNIX/Linux-based Platforms

The following software should be installed before installing UAS:

- Check Point SVN Foundation (most current version)
- Check Point VPN-1 Pro (most current version). For information on installing VPN-1 Pro, see the *Check Point SmartCenter Guide*.

To install UserAuthority on a UNIX/Linux-based machine:

1. Insert the Wrapper (package) in the machine's CD drive.
2. Turn on the machine (the machine should be configured to boot from the CD drive).

   Follow the on-screen instructions. For information on the configurations necessary for the installation, including establishing SIC, see the section on "Windows" on page 332. Although the GUI interface is different, the procedure is the same. Note that if you have already installed the VPN-1 Pro, establishing SIC is not necessary.

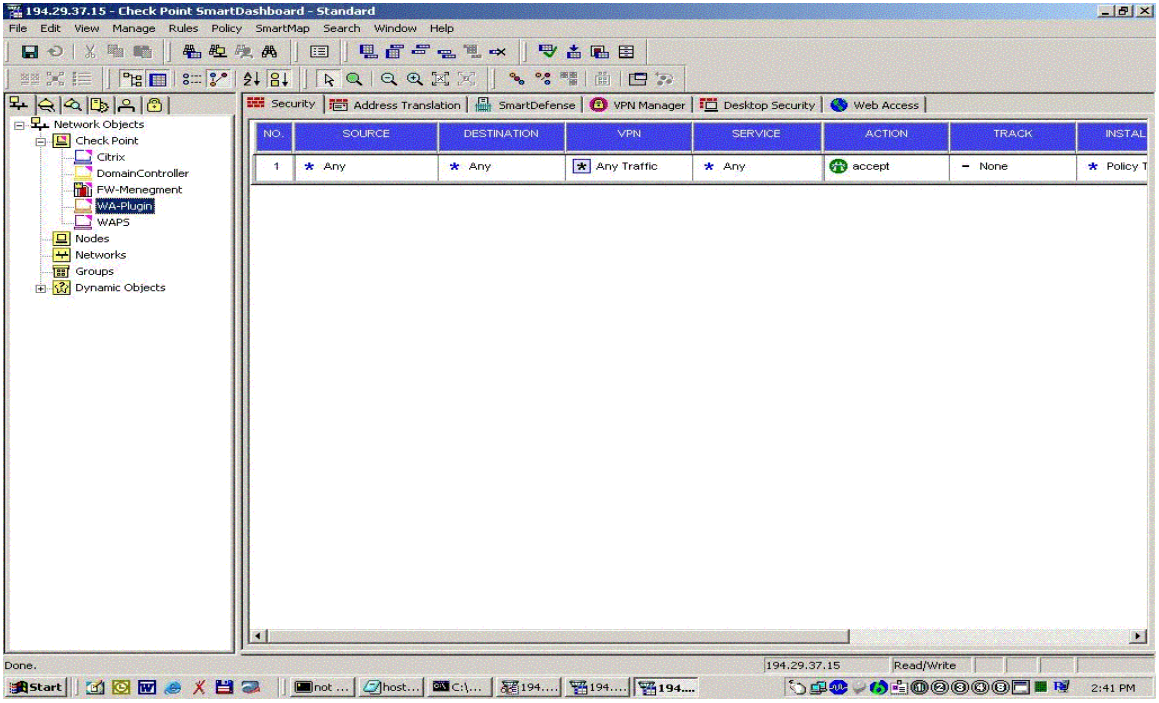
3. Use the Check Point Configuration Tool to install a license on the SmartCenter machine (see "Installing the UserAuthority License" on page 31). For information on the Check Point Configuration Tool, see the *Check Point SmartCenter Guide.*

Configuring the UAS

You now need to configure UAS using SmartDashboard. For more information on SmartDashboard, see the *Check Point SmartCenter Guide*.

Figure 2-10 shows the SmartDashboard **Main** window with the **Network Objects** tree in the Tree pane.

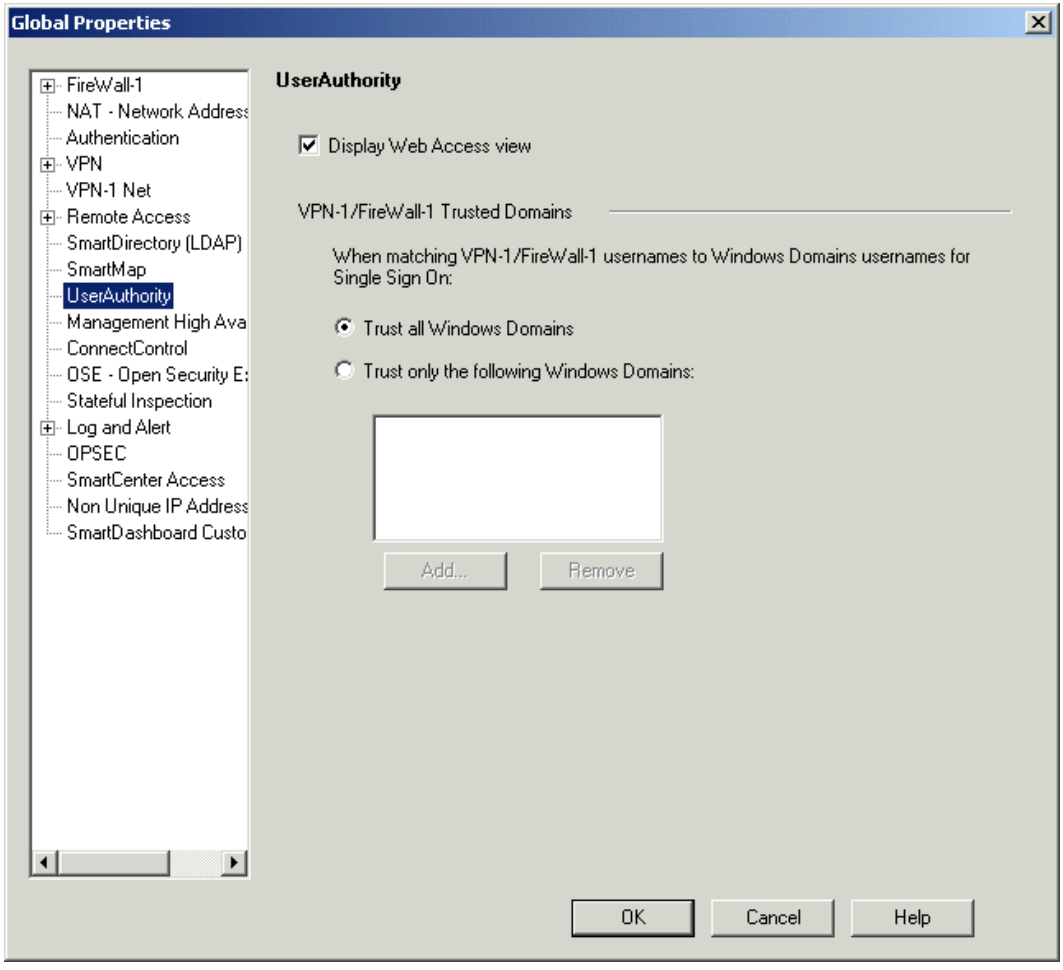
Figure 2-10 SmartDashboard Network Objects



To configure the UAS:

1. From the SmartDashboard **Policy** menu, select **Global Properties**. The **Global Properties** window is displayed.
2. In the Tree pane, click **UserAuthority** to display the **UserAuthority Properties** window.

Figure 2-11 Global Properties Window (UserAuthority Properties)



3. Select the **Display Web Access view** checkbox. This displays the **Web Access** tab in SmartDashboard. If your deployment does not include the WAPS, this step is optional. Click **OK**.

4. Create a new network object. (Carry out this step only if a network object for the VPN-1 Pro gateway has not already been created. If a network object has already been created, skip to step 6 on page 40):

   a. In the SmartDashboard **Network Objects** tree, right click **Network Objects**. From the shortcut menu, select **New > Check Point > Gateway**. The **Check Point Gateway** window is displayed.

   b. In the **Name** field, enter the name of the firewall gateway where the UAS is installed.

c. Enter the IP address for the firewall gateway in the **IP Address** field.

d. From the **Version** drop-down list, select **NGX R61**.

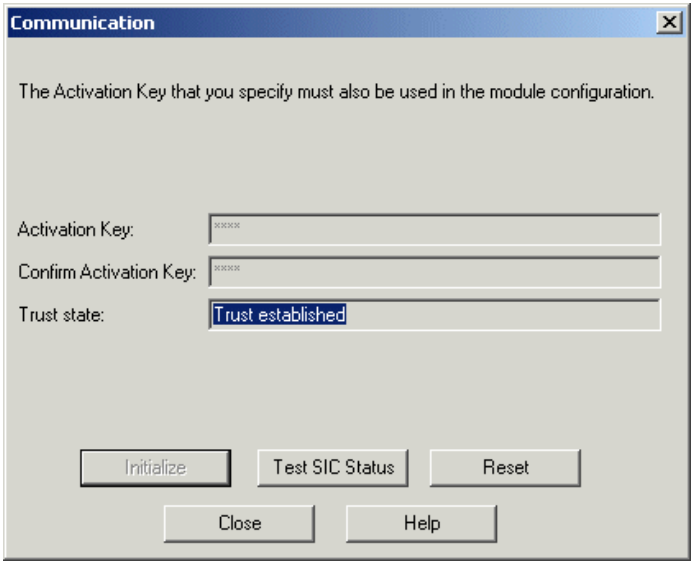
e. From the list of Check Point products, select **UserAuthority Server**. (You may have to scroll down the list to find **UserAuthority Server**.)

Note - If you did not select **Display Web Access view** in step 3 and you are not using UserAuthority WebAccess in your deployment, ignore the error message displayed. If you are using UserAuthority WebAccess in your deployment and a UserAuthority WebAccess error message is displayed, go to step 3 to and select **Display Web Access view** in the **User Authority** tab of the **Global Properties** window.

5. Establish SIC:

a. In the **Secure Internal Communication** area of the **Check Point Gateway** window, click **Communication** to display the **Communication** window.

Figure 2-12 Communication window



b. In the **Activation Key** field, enter the Activation Key that you created when you configured the SIC Policy (see "Installing UAS on the VPN-1 Pro Gateway" on page 32, step b on page 36).

c. Enter the Activation Key again in the **Confirmation** field.

d. Click **Initialize**.

If the operation is successful, the words **Trust established** are displayed in the **Trust state** field.



Note - If the SIC operation is not successful, click **Reset** and reset the SIC on the UAS. Try again. *Verify that you are entering the correct SIC Activation Key.*

e. Click **Close** to return to the **Check Point Gateway** window.

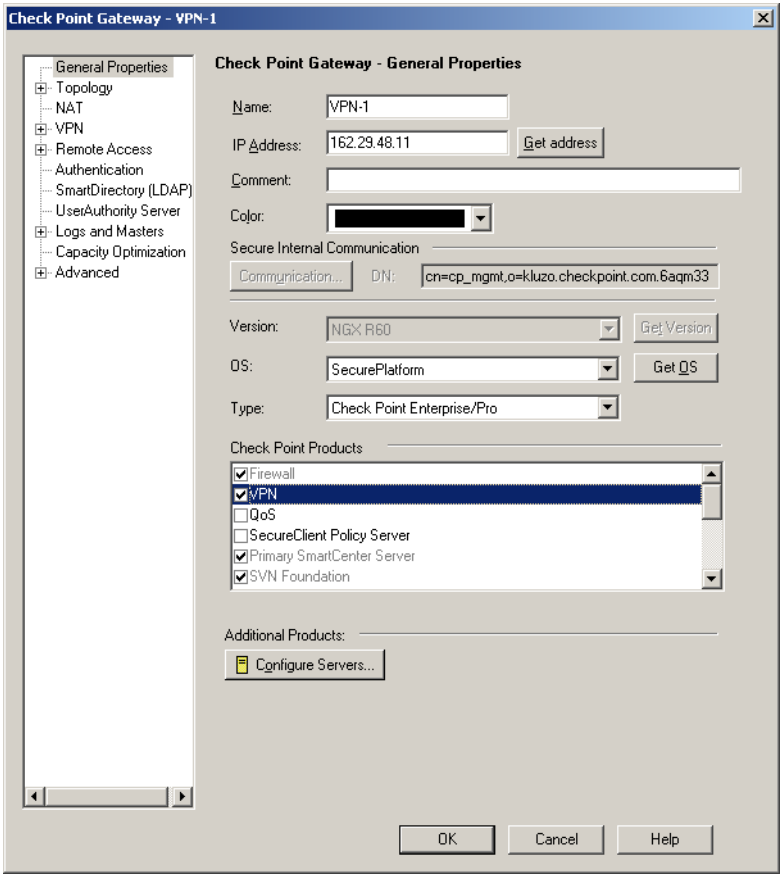
6. Add UAS to an existing VPN-1 Pro network object. If you added a network object and initiated SIC in step 4 and step 5, then skip to step 7 on page 41.

   a. Double click the VPN-1 Pro network object in the **Network Objects** tree in the Tree pane.

   b. From the list of Check Point products, select **UserAuthority Server**. (You may have to scroll down the list to find **UserAuthority Server**.) UserAuthority is displayed in the Tree pane of the **Check Point Gateway** window.

      The **Check Point Gateway** window should resemble Figure 2-13.

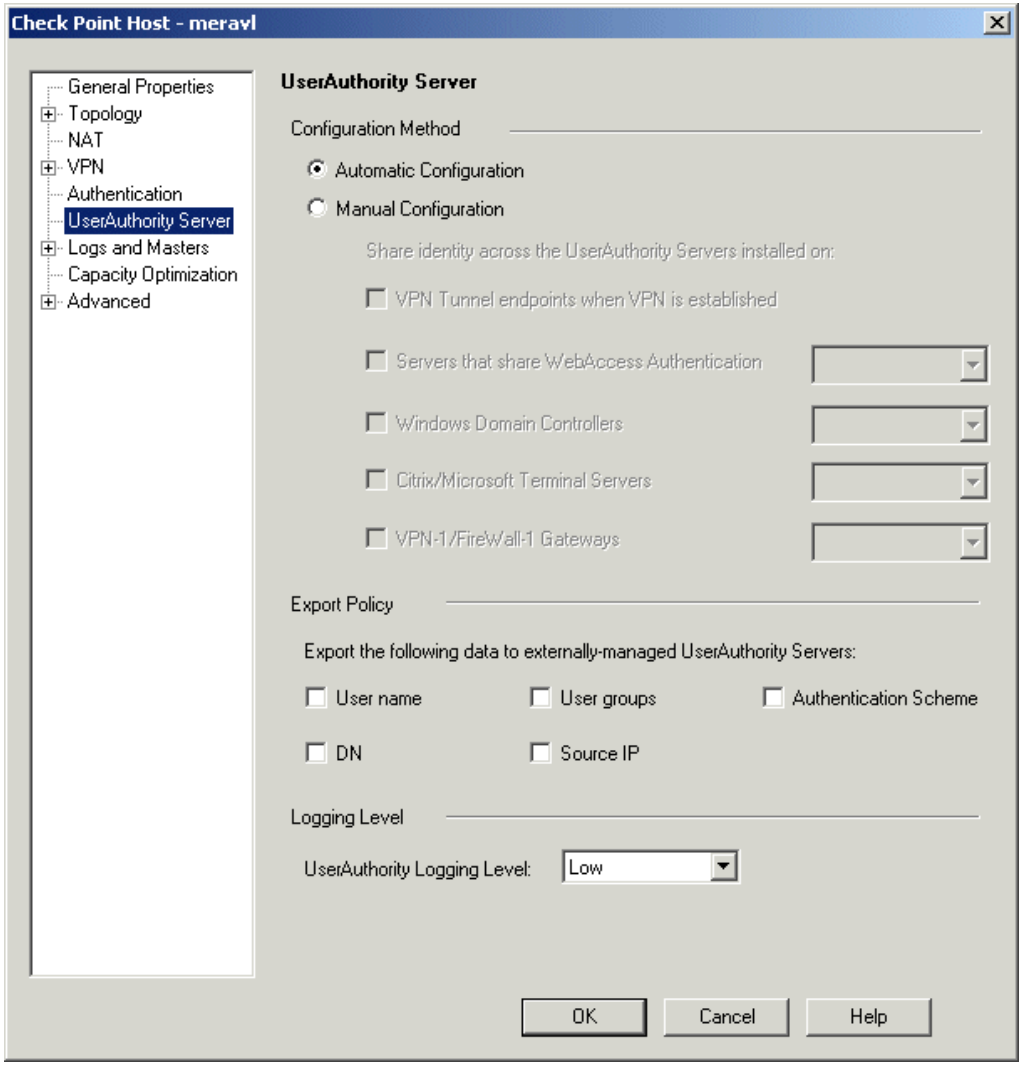
Figure 2-13 Check Point Gateway Window



7. Click **UserAuthority Server** in the Tree pane of the **Check Point Gateway** window to open the **UserAuthority** host window. Leave the default **Automatic Configuration** chaining option selected. This automatically sets up your deployment for chaining. For information on advanced chaining options, see "Configuring Manual Identity Sharing Options" on page 58.

   The **UserAuthority Server** window should resemble Figure 2-14.

Figure 2-14 Shared Identity Options



8. Click **OK** to close the window.

Installing and Configuring the UAS on the Windows DC

For deployments where the Windows DC is used to identify clients on the network, you need to install the UAS as a stand alone module on the Windows DC. The UAS is used for administration and enforcement of user authentication for the enterprise's network.



Note - The UAS can be installed on any computer in the domain.

The following components are required for this installation:

- VPN-1 Pro module installed on a gateway or other server
- VPN-1 Pro management installed on a gateway or other server
- SmartDashboard
- UAS installed on a VPN-1 Pro gateway

The following steps are required to install and configure the UAS on the Windows DC:

- Install UAS
- Configure SIC policy
- Configure SecureAgent automatic installation
- Configure the UAS properties
- Add an SSO rule

Installing the UAS

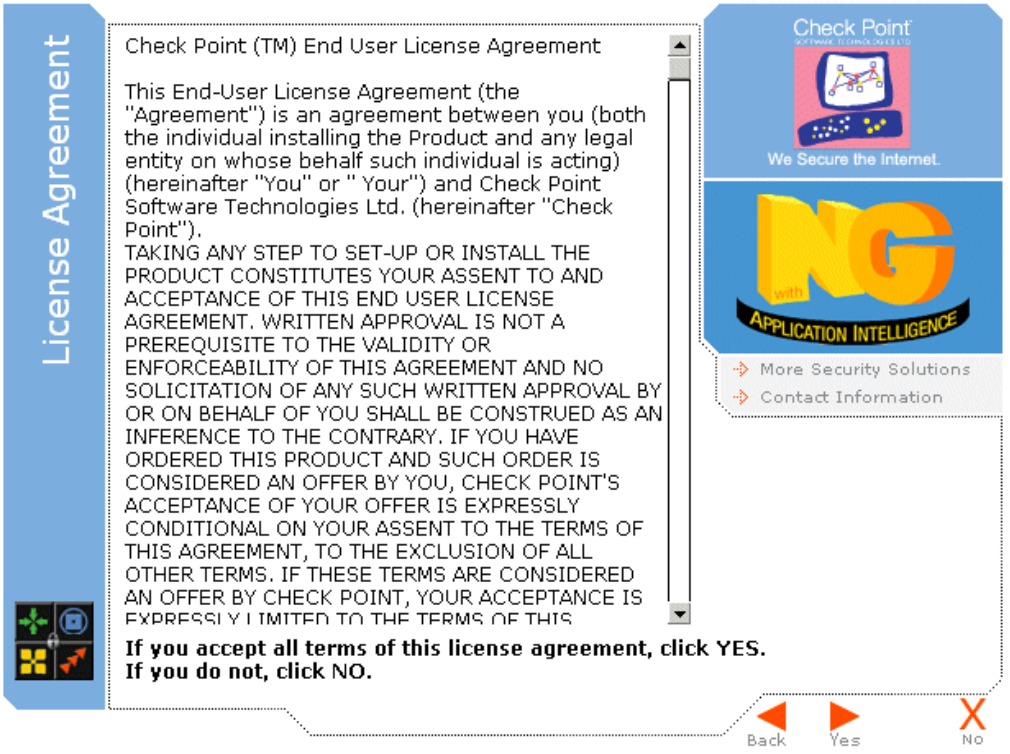


Note - This installation automatically includes the Secure Virtual Network (SVN) Foundation.

To install the UAS:

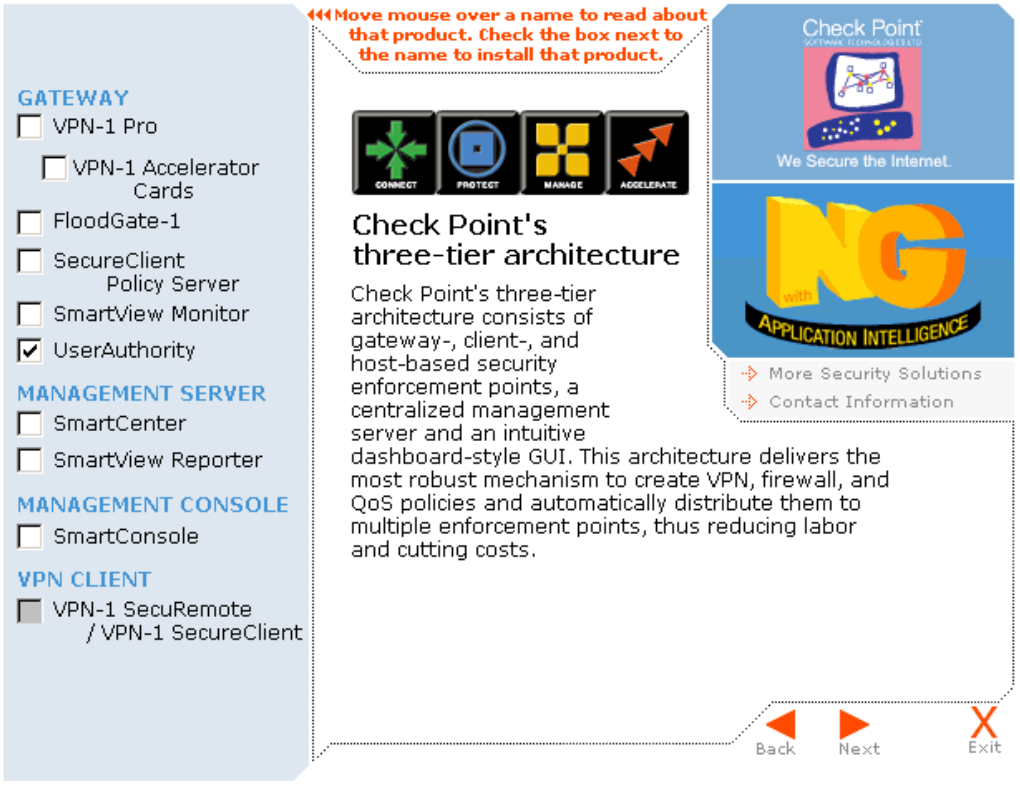
1. Insert the Wrapper CD and then run the Wrapper. The Installation Welcome window is displayed.
2. Click Next. The End-Users License Agreement (EULA) is displayed.

Figure 2-15 Licence Agreement



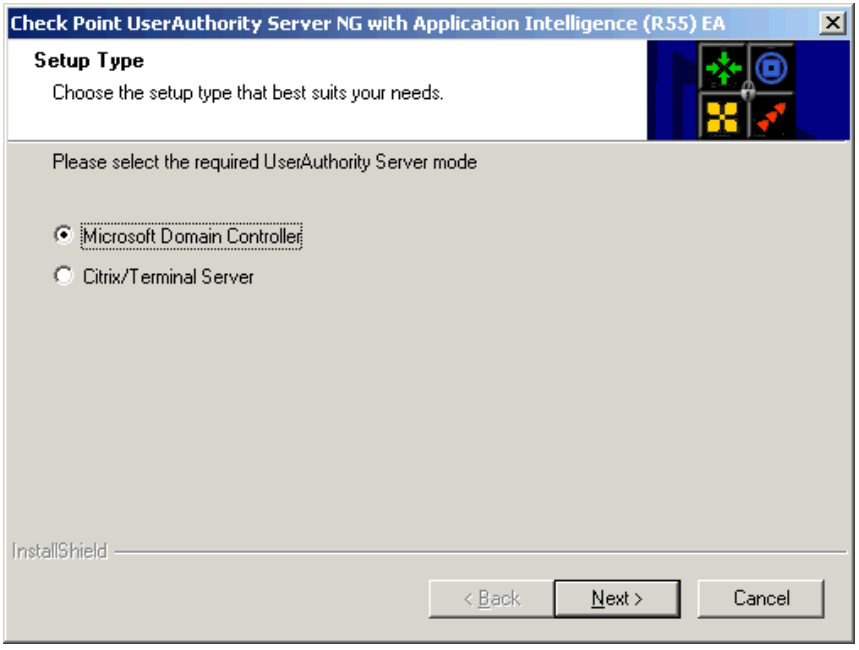
3. Read the End-Users License Agreement (EULA) and then click Yes to accept it. The next installation window is displayed.
4. Select Check Point Enterprise/Pro as the type of installation, and then click Next. The next installation window is displayed.
5. Select New Installation and click Next. The next installation window is displayed.
6. Select UserAuthority from the list of Check Point products. Clear all other checkboxes.

Figure 2-16 Product Selection for UserAuthority on the Windows DC



7. Click Next to start the Install Shield. A list of the products you selected to install is displayed. UserAuthority should be the only product listed.
8. Follow the on-screen instructions. You should be aware of the following:
   - The SVN Foundation is installed automatically.
   - If you are installing UAS on a Citrix or Terminal Services (not on a Windows DC), select Citrix/Terminal Services in the Setup Type window.

Figure 2-17 Setup Type



9. Click Next, the next window is displayed.
10. Browse to the folder in which you want to install UserAuthority, or click Next to install in the default folder.
11. At the end of the installation, click OK. The License window is displayed.
12. You do not need a license for UAS on the Windows DC. Click Next and then click Yes when the warning “You have no licenses” is displayed.
13. The Key HIt Session window is displayed. Follow the on-screen instructions and click Next.
14. The Secure Internal Communication (SIC) window is displayed. Enter a password key in the Activation Key field and then enter it again in the Confirm Activation Key field. Be sure to remember your key, you will need to enter it in the SmartDashboard configuration.
15. The Thank you for using... message is displayed. Click OK.
16. Remove the CD and then click Finish to restart the computer.
17. If you installed the UAS on another machine in the Windows Domain instead of on the Windows DC, you need to configure the uatcs-acl.txt file.
    a. Open the uatcs-acl.txt file in Windows WordPad.

b. Edit the following file parameters:

[hostname]: The host name of the UAS

[ipaddress]: The IP address of the UAS

[port]: The UAS UDP source port (this should always be 19195)

The following is an example of a uatcs-acl.txt file configured to accept queries from a Windows DC with the name DC, IP address 10.0.0.2, and port number 19195.

```
#
#hostname              ipaddress              port
#
DC                     10.0.0.2               19195
```

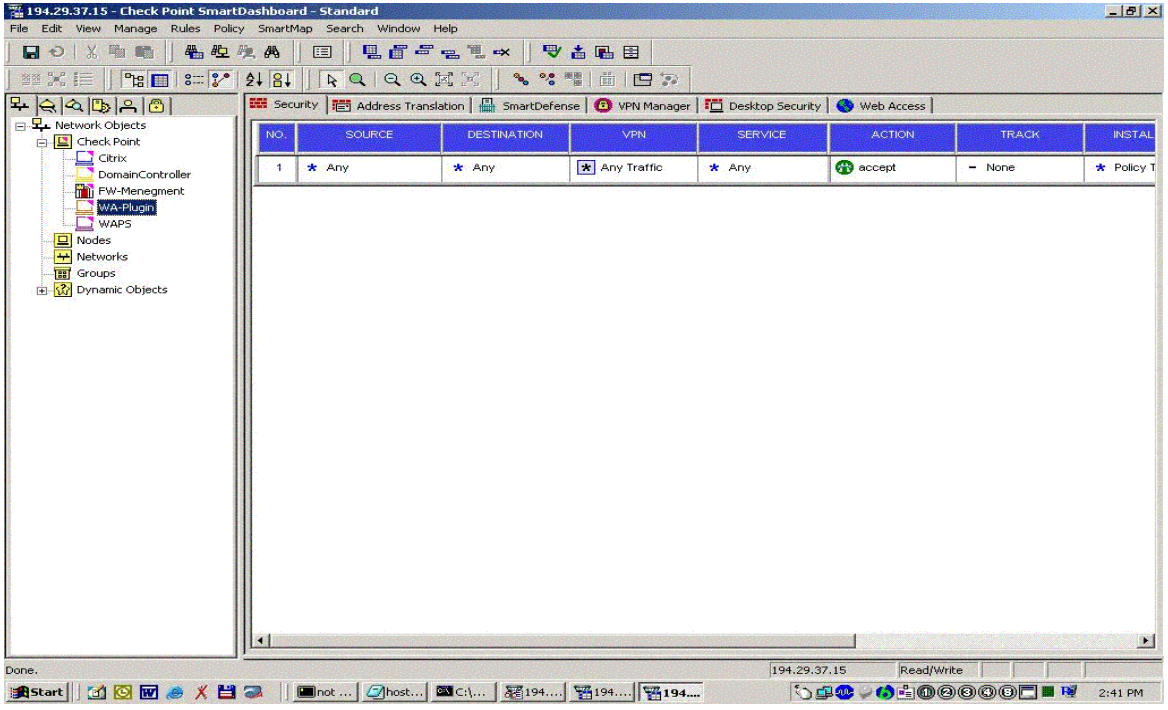
c. Save and close the file.

Configuring UAS Properties

You need to configure the UAS using SmartDashboard. For more information on how to use SmartDashboard or if it is not installed on the management server, see the Check Point SmartCenter Guide.

Figure 2-18 shows the SmartDashboard Main window with the Network Objects tree in the Tree pane.

Figure 2-18 SmartDashboard Network Objects



To configure the UAS:

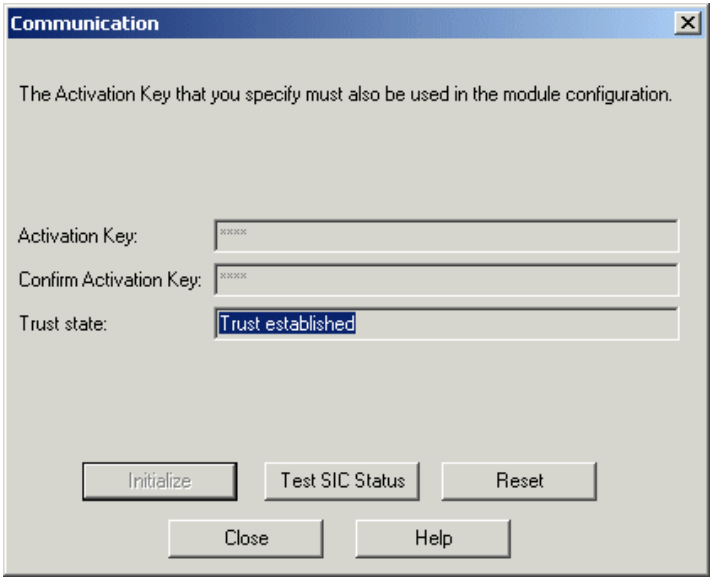
1. Create a new network object:
   a. In the SmartDashboard Network Objects tree, right click Network Objects. From the shortcut menu, select New > Check Point > Host. The Check Point Host window is displayed.
   b. In the Name field, enter the name of the Windows DC (or other computer in the domain) where UAS is installed.
   c. Enter the IP address for the Windows DC in the IP Address field.
   d. From the Version drop-down list, select **NGX R61**.
   e. From the list of Check Point products, select UserAuthority Server. (You may have to scroll down the list to find UserAuthority Server.)

Note - In the event that an alert about the UserAuthority WebAccess rule base is displayed, ignore it and continue.

2. Establish SIC:

a. In the Secure Internal Communication area of the Check Point Host window, click Communication to display the Communication window.

Figure 2-19 Communication Window



b. In the Activation Key field, enter the Activation Key that you created when you configured the SIC Policy (see "Installing the UAS" on page 43, step 14 on page 46).

c. Enter the Activation Key again in the Confirmation field.

d. Click Initialize.

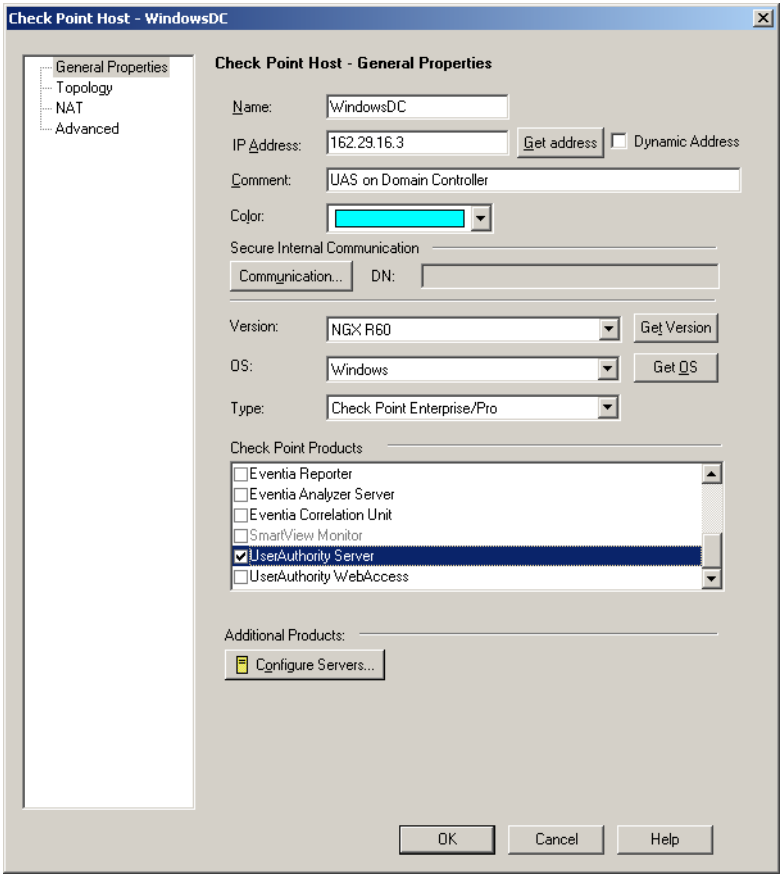
If the operation is successful, the words Trust established are displayed in the Trust state field.

Note - If the SIC operation is not successful, then click **Reset** and rest the SIC on the UAS and on the Windows DC. Try again. Verify that you are entering the correct SIC Activation Key.

e. Click Close to return to the Check Point Host window.

The Windows DC Host window should resemble Figure 2-20.

Figure 2-20 New Windows DC Window



3. Click OK to close the Check Point Host window.
4. Save and install the policy on the VPN-1 Pro where the UAS is installed.

Configuring SecureAgent Automatic Installation

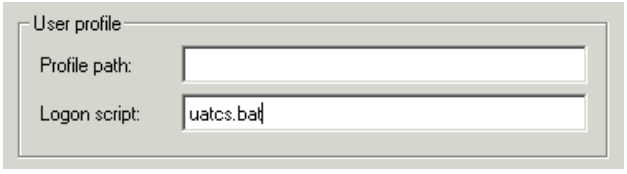
UserAuthority can be configured to automatically install SecureAgent on the client at startup using a Windows logon script. The logon scripts must be in a Windows DC folder called NETLOGON Share. If you installed the UAS on another machine in the Domain instead of on the Windows DC, copy the files listed in Table 2-1 on page 51 to the NETLOGON directory on the Windows DC.

If a logon script exists, modify it so that it also runs instuac.bat. If there is no logon script, perform one of the following procedures.

On Windows 2000 with Active Directory:

1. From the Control Panel, double click Administrative Tools.
2. Double click Active Directory Users and Computers.
3. In the Tree pane, right click a user name and then click Properties from the menu. The Properties window is displayed.
4. Click the Profile tab.
5. In the Logon script field, enter uatcs.bat.
6. Click OK to close the window.

Figure 2-21 User Profile Login Script



On Windows NT:

1. From the Control Panel, double click Administrative Tools.
2. Double click User Manager for Domains.
3. Select the name of a user.
4. From the User menu, select Properties to display the User Properties window.
5. In the User Properties window, click the Profile tab.
6. In the Logon script field, enter uatcs.bat.
7. Click OK to close the window.

The following files are installed in the NETLOGON share folder:

Table 2-1 NETLOGON Share Files

Instuac.exe	The SecureAgent installation and uninstall program.
uatc.exe	The SecureAgent executable.
uatcs.bat	A batch file that runs instuac.exe with some parameters to install SecureAgent.
uatcs_uninstall.bat	A batch file that runs instuac.exe to uninstall SecureAgent.
uatcs-acl.txt	An access list that determines to which UASes the SecureAgent responds.

You can also adjust the SecureAgent installation mode. By default, uatcs.bat installs SecureAgent with a GUI, a log file and a shortcut to the Start menu. You can make changes to the file using the following parameters.

Table 2-2 uatcs.bat Parameters

/help or/?	Displays the usage.
/norun	Do not run after installation.
/shortcut	Installs a shortcut in the **Start** menu.
/uninstall	Uninstalls SecureAgent.
/uatcfile <filename>	Installs <filename>.
<args>	Passes specific arguments to the SecureAgent executable file (see following parameters).
/icon	Runs SecureAgent with the icon displayed in the task bar system tray.
/debug	Prints system information into a SecureAgent log file (`uatc.log`). The file is located in the same directory as SecureAgent.
/kill	Stops SecureAgent.
/nodiscover	Does not perform Windows DC auto-discovery. (This option should not be selected because it allows SecureAgent to accept queries from any source.)

Chapter 3

Outbound Access Control

In This Chapter

The Challenge	page 54
The UserAuthority Solution	page 55
Retrieving Windows Groups with UserAuthority	page 62
Outbound Access Control using Citrix Terminals as TIP	page 63
Scenario - An Organization using Multiple Windows DCs	page 64
Scenario - An Organization Using Multiple Domains	page 66
Configurations	page 68

The Challenge

Many enterprises grant their users access to external resources (such as the Internet) from the local network. The network administrator often needs to control the traffic that leaves the internal network. This can be achieved by:

- Restricting access to specific external resources for some or all users
- Auditing user requests for external resources

For a variety of reasons, an enterprise may want to restrict users' access to external resources. Internal policy may determine that users cannot access competitors' Web sites to ensure that privacy is maintained, or that users can only access the Internet if their position in the enterprise requires it. In other cases, an enterprise may decide to limit Internet access to specific users, or allow differing levels of access based on the user's position.

In addition, an enterprise may want to keep track of users' access of external resources, for example, the amount of time spent using external resources and which resources are being used.

Many available security applications intercept and limit traffic entering and exiting various external networks and the Internet. A firewall, such as Check Point's VPN-1 Pro, is one such solution that can also be used to monitor a local network's inbound and outbound traffic, providing the enterprise with valuable information regarding how each user is utilizing external resources. Users must authenticate to the security application each time they access an external resource.

The added challenge here is to create Single Sign-On (SSO) for LAN users who are accessing external resources. UserAuthority provides Single Sign-On (SSO), eliminating the need to repeatedly submit credentials. SSO provides one-time authentication for all applications, which remains valid for subsequent access attempts. In this case however, UserAuthority requires no additional authentication if the user has already been authenticated by Windows.

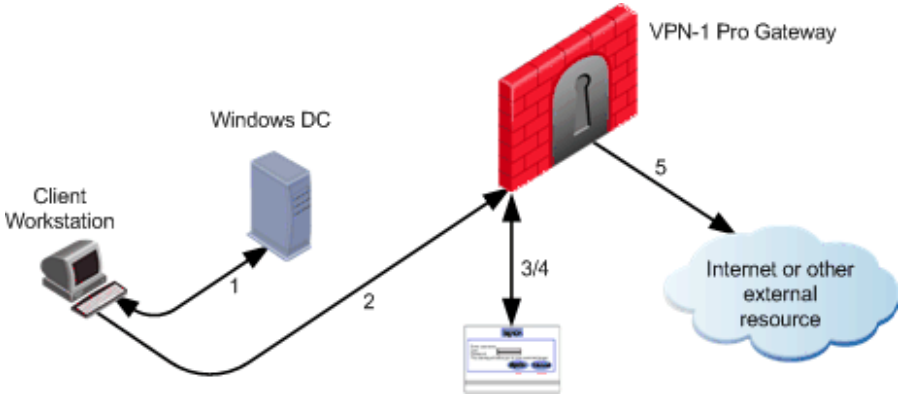
The UserAuthority Solution

In This Section

Identification using SecureAgent page 57
Identity Sharing page 57

UserAuthority eliminates the need for authentication by retrieving the user's identity from the Windows Domain Controller (DC) and providing it to VPN-1 Pro. In a system without UserAuthority, VPN-1 Pro requires authentication each time an external resource is requested, in order to identify the user and allow the user's request to go through the VPN-1 Pro. In addition, without the ability to identify the user, there is no way to keep track of the outbound traffic. Figure 3-1 shows how outbound traffic is handled by the firewall in a system without UserAuthority.

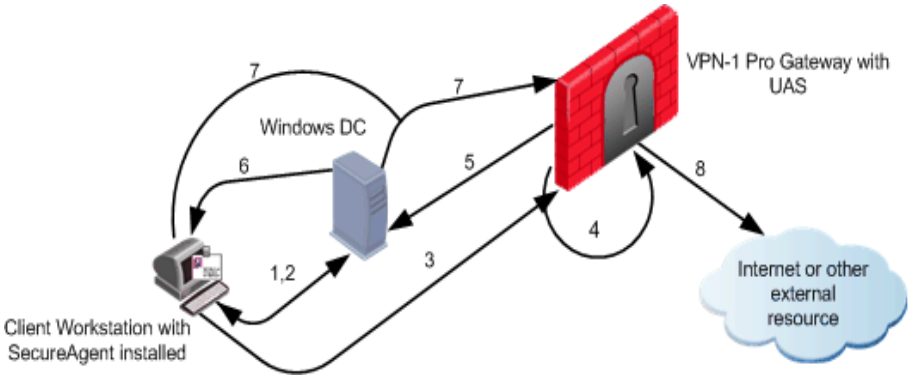
Figure 3-1 Outbound Requests without UserAuthority



1. A user signs on to the domain and authenticates to the Windows DC.
2. The user accesses an external resource.
3. The VPN-1 Pro gateway intercepts the request and, based on the VPN-1 Pro policy (authorization or auditing), tries to authenticate the user.
4. The user enters credentials for VPN-1 Pro and sends them back.
5. VPN-1 Pro receives the credentials and grants the user access to the external resource.

UserAuthority provides the means to easily identify the user and keep track of user activities. If a UserAuthority Server (UAS) is installed on the VPN-1 Pro gateway and the Windows DC, identification is performed by UserAuthority, without the user having to authenticate to VPN-1 Pro. Figure 3-2 illustrates this process.

Figure 3-2 Outbound Request with Outbound Access Control



1. A user signs on to the Domain and authenticates to the Windows DC.
2. UserAuthority SecureAgent is copied to the user's desktop.
3. The user accesses an external resource.
4. The VPN-1 Pro gateway intercepts the request and, based on the VPN-1 Pro policy (authorization or auditing), queries the UAS installed on the gateway for the user's identity.
5. The UAS on VPN-1 Pro sends the request to the UAS on the Windows DC.
6. The UAS on the Windows DC retrieves the user identity from SecureAgent on the user's desktop.
7. The identity is sent back through the Windows DC to the VPN-1 Pro gateway.
8. The user is granted access to the external resource.

The examples described in this section show how UserAuthority solves the authentication problem by using the UserAuthority SecureAgent to identify the user.

Identification using SecureAgent

Outbound Access Control uses UserAuthority SecureAgent to identify the user. SecureAgent is automatically installed on all clients in the network, so there is no need for individual installation and configuration. UserAuthority SecureAgent is an executable that is installed and run on desktop computers in a Windows domain. SecureAgent identifies the user (who is signed on to the Windows domain) by responding to queries from the UAS installed on the domain. UserAuthority provides SSO, eliminating the need for the user to repeatedly submit his/her credentials.

The Trusted Identification Point (TIP) for this scenario is the Windows DC and the UAS installed on the Windows DC provides the identification.

Identity Sharing

Identity sharing is used by the UAS to get the user's identity from other UASes in the enterprise's intranet. In the Outbound Access Control deployment, identity sharing is used by the UAS on the gateway to retrieve the user's identity from the UAS on the Windows DC.

By default, identity sharing is automatically configured in your deployment and sharing is implemented when the UAS does not have any information about the user's identity.

The default identity-sharing configuration is:

- If the request arrives over a VPN tunnel from another gateway, the UAS queries the UAS on the originating gateway.
- UAS queries all UASes on Windows DCs or Terminal Services.

Identity sharing can also be configured manually if it is necessary for your deployment. For information on configuring identity sharing, see "Configuring Manual Identity Sharing Options" on page 58.

UserAuthority uses two protocols for identity sharing. The UAA protocol is used for communication between UASes, and the SSPI protocol is used for communication between the UAS on the Windows DC and UserAuthority SecureAgent.

Configuring Manual Identity Sharing Options

One of the greatest advantages of UserAuthority is its ability to extract the user identity from a Trusted Identification Point (TIP). UserAuthority establishes a trust relationship with TIPs on the network to ensure that it is receiving trusted information.

UserAuthority searches the local hosts and servers to find the information necessary to carry out a request. If the information is not available locally, identity sharing is invoked to search other components in the deployment, for the information.

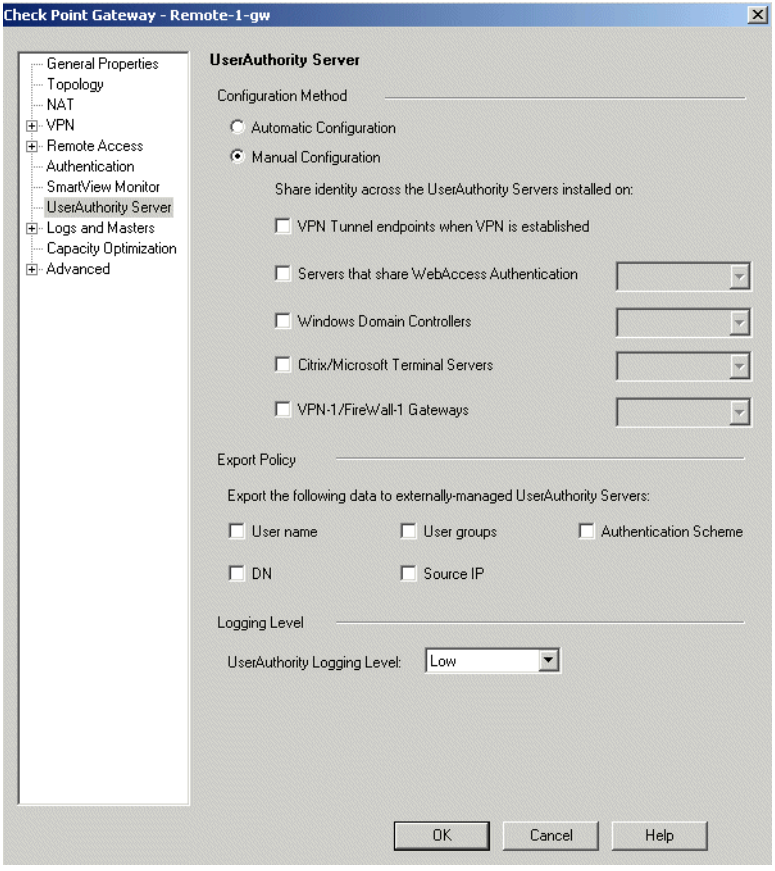
Most deployments of UserAuthority use *automatic* identity sharing (default configuration). Automatic identity sharing searches each UserAuthority module on the same internally managed domain, for example Domain Controllers, Citrix machines and VPN peers, “chaining” them together to retrieve the user identity.

This section describes how to configure manual identity sharing in UserAuthority.

To set manual identity sharing options:

1. Using SmartDashboard, select the desired **VPN-1 Pro** (with UAS) Network Object on the **Network Object** tree, in the left-hand pane of the window.
2. Double click the **VPN-1 Pro** Network Object. The **VPN-1 Pro Host** window is displayed.
3. Click **UserAuthority Server** on the tree, in the left pane of the **VPN-1 Pro Host** window. The **UserAuthority Server** window is displayed.

Figure 3-3 UserAuthority Identity Sharing Options



In this window you can configure the settings for UserAuthority Servers (UAS) "chaining", enabling you to retrieve user identity information from other UserAuthority Servers.

4. In the **Configuration Method** area, select **Manual Configuration**. The identity sharing options can now be configured.

5. Select one, or more options. UserAuthority determines the identity sharing priority according to the user's point of entry.

   The last four options require you to select a group of UA Servers to be queried. To be able to do this, you must create a UAS group as explained in "Creating UAS Groups" on page 61.

   - **UserAuthority Servers on VPN tunnels endpoints**: When a user enters the network via a VPN connection, the opposite end of the VPN tunnel is queried for the user's identity.

- **UserAuthority Servers that share WebAccess Authentication**: When a user authenticates to UserAuthority WebAccess, the UAS associated with the WAPS is queried for the user's identity. When this option is selected, you must select the Server Group to be searched from the drop-down list.
- **UserAuthority Server on Windows Domain Controllers:** For users in a Windows domain, the Windows DC(s) are queried for the user's identity. When this option is selected, you must select the Server Group to be searched from the drop-down list.
- **UserAuthority Server on Citrix/MicroSoft Terminal Services**: If a user uses resources via a Citrix/Windows terminal, UAS on a Terminal Server is queried for the user's identity. When this option is selected, you must select the Server Group to be searched from the drop-down list.
- **UserAuthority Server on Remote Access VPN Gateways**: Searches for the user's identity by querying all valid remote VPN gateways, not only VPN endpoints, for information. When this option is selected, you must select the Server Group to be searched from the drop-down list.

6. In the **Export Policy** area, specify the information that will be exported to *externally managed* UserAuthority Servers (that is, VPN peers that are not managed by this SmartCenter server). There is no restriction on information made available to *internally managed* UserAuthority Servers.



Note - Exporting of UserAuthority information can be done only between two VPN-1 gateways. The UserAuthority Server performing the query should be configured by enabling UserAuthority Servers over VPN tunnels. The UserAuthority Server supplying the information should be configured with an appropriate Export Policy. Both sides should have Security Policy rules that will allow the UserAuthority protocol (FW1_uaa) using IKE, and encrypt it.

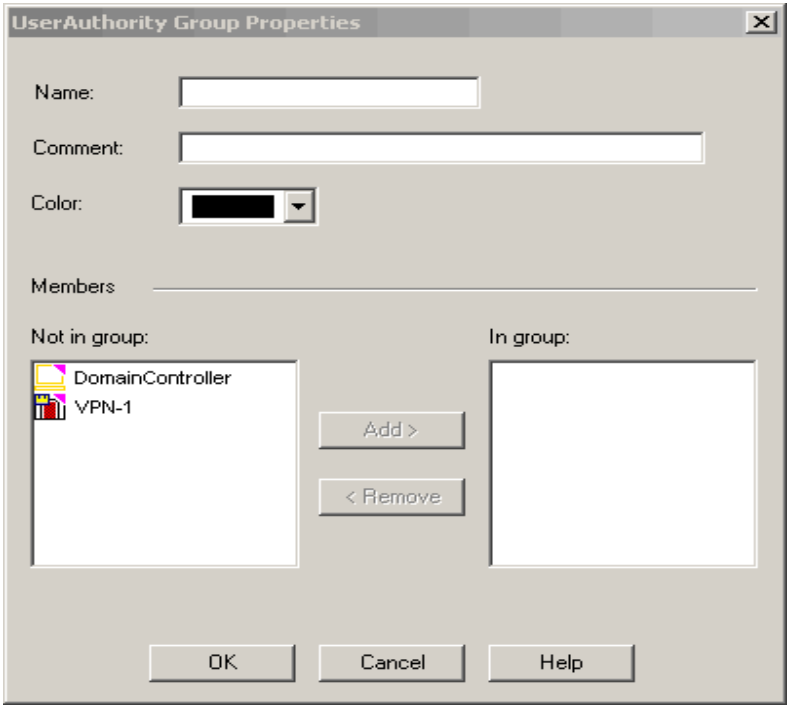
7. In the **Logging Level** area, select the logging level from the drop-down list. The following levels are available:
   - **Low:** Logs all non-query related events, for example, loading policy, UAS up, UAS down.
   - **Medium:** Logs all non-query related events *and* all query replies dealing with authentication failures.
   - **High:** Also logs all queries and replies.
8. Click **OK**.

Creating UAS Groups

1. Create a UAS Group:
   a. In SmartDashboard, right click **Groups** in the **Network Objects** tree. From the shortcut menu, select **New Groups > UserAuthority Server Group**. The **UserAuthority Groups Properties** window is displayed.
   b. Enter a name for the group in the **Name** field.
   c. You can enter a comment in the **Comment** field (optional).
   d. From the **Not in group** field, select the name you gave to the object containing the UAS.
   e. Click **Add**. The selected Network Object is moved to the **In group** field.
   f. Click **OK** to close the window.

The window should resemble Figure 3-4.

Figure 3-4 UserAuthority Group Properties Window



Retrieving Windows Groups with UserAuthority

UserAuthority can retrieve Windows-defined groups for users identified by UserAuthority. This allows authorization to be handled on the VPN-1 Pro using pre-defined Windows groups.

Groups are defined in SmartDashboard to make it easier to administer large numbers of people. UserAuthority uses the Group data on users from the Windows environment to authorize users, eliminating the need to transfer data from Windows to the VPN-1 Pro database or to an LDAP database.

This is done by creating Groups in SmartDashboard that correspond to the Groups in the Windows Domain. To do this, create a Group and give it the exact same name as its related Windows Group with the prefix `WIN_<domain name>_`. For additional information see "Configuring UserAuthority to Recognize Windows User Groups" on page 79.



Note - Using Windows groups with UserAuthority requires that UAS be installed on the Windows DC.

Outbound Access Control using Citrix Terminals as TIP

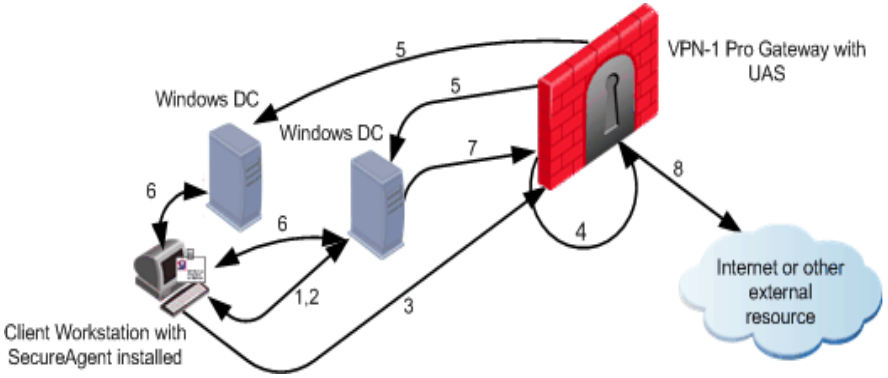
Users working on a Citrix or Terminal Services machine have no IP uniqueness because each machine connected to the terminal shares the same IP address. In this case, UserAuthority on the Terminal Services retrieves the user information and identifies the user through the connection information. Because the UAS on the Terminal Services can retrieve the connection information to identify the user, there is no need to use UserAuthority SecureAgent. When you configure VPN-1 Pro in this deployment, you must set S*ession Authentication in SmartDashboard*. (See "Citrix MetaFrame or Windows Terminal Services" on page 28.)

Scenario - An Organization using Multiple Windows DCs

Some enterprises work with more than one Windows DC on a domain. In this case, the UAS on the VPN-1 Pro gateway may need to query various Windows DCs for user identification. Because the automatic identity-sharing configuration checks all UASes for user identification, no special configuration is required. This scenario is recommended for high availability.

Figure 3-5 shows a deployment with more than one Windows DC.

Figure 3-5 Outbound Access Control deployed with Multiple Windows DCs



1. A user signs on to the domain and authenticates to the Windows DC.
2. UserAuthority SecureAgent is copied to the user's desktop.
3. The user accesses an external resource.
4. The VPN-1 Pro gateway queries the UAS installed on the gateway for the user's identity.
5. The UAS on VPN-1 Pro queries the UAS on each Windows DC.
6. UAS on the Windows DC retrieves the user identity from UserAuthority SecureAgent on the user's desktop.
7. The user's identity is sent back to the VPN-1 Pro gateway from the first UAS to identify the user.
8. The user is granted access to the external resource.

Workflow

To deploy Outbound Access Control with multiple Windows DCs:

1. Install the UAS on the VPN-1 Pro gateway. See "Installing and Configuring UAS on VPN-1 Pro" on page 31.
2. Install and configure UAS on the Windows DCs in your network. See "Installing and Configuring the UAS on the Windows DC" on page 43.
3. Configure the system to automatically install SecureAgent on each of the Windows DCs. See "Configuring SecureAgent Automatic Installation" on page 50.
4. From the SmartDashboard **Security** tab, configure an SSO rule. See "Adding an SSO Rule" on page 25.

Test Your Deployment

The deployment should work the same as with a single DC.

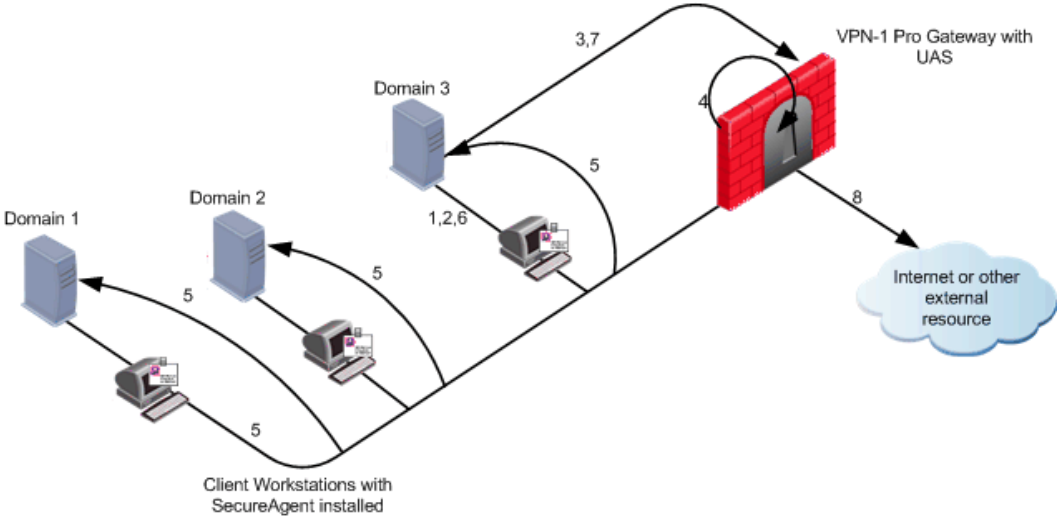
Scenario - An Organization Using Multiple Domains

Some enterprises work with more than one Domain. In this case, the UAS on the VPN-1 Pro gateway may need to query the UASes on the Windows DCs in the different domains for user identification. Because the automatic identity-sharing configuration checks all UASes for user identification, no special configuration is required. This scenario requires installing a UAS on each domain.

Each user identity in a Windows domain is defined with two parts, the domain and the user identity (usually the user name). By default, UserAuthority ignores the domain name when identifying a user. In this way, the user is recognized no matter which domain they are using. For example, domain1/Bill and domain2/Bill are recognized as the same user by UserAuthority. This is called domain equality. If you do not want UserAuthority to recognize the user as the same on all domains, you must manually configure the domain equality options. For information on configuring domain equality, see "Configuring UserAuthority Domain Equality" on page 69.

Figure 3-6 shows a deployment with more than one domain.

Figure 3-6 SSO for VPN-1 Pro deployed with Multiple Domains



1. A user signs on to the domain and authenticates to the Windows DC.
2. UserAuthority SecureAgent is copied to the user's desktop.
3. The user accesses an external resource.

4. The VPN-1 Pro gateway queries the UAS installed on the gateway for the user's identity.
5. The UAS on VPN-1 Pro queries UAS on each Windows DC.
6. The UAS on the Windows DC retrieves the user identity from SecureAgent on the user's desktop. Only the UAS on the Windows DC where the user was authenticated can identify the user using SecureAgent.



Note - If a Windows Trust is established between the domains, then any domain can identify the user.

7. The identity is sent back to the VPN-1 Pro gateway.
8. The user is granted access to the external resource.

Workflow

To deploy Outbound Access Control with multiple Windows DCs:

1. Install the UAS on the VPN-1 Pro gateway. See "Installing and Configuring UAS on VPN-1 Pro" on page 31.
2. Install and configure UAS on the Windows DCs in your network. See "Installing and Configuring the UAS on the Windows DC" on page 43.
3. Configure the system to automatically install SecureAgent on each of the Windows DCs. See "Configuring SecureAgent Automatic Installation" on page 50.
4. From the SmartDashboard **Security** tab, configure an SSO rule. See "Adding an SSO Rule" on page 25.

Test Your Deployment

The deployment should work the same as with a single DC.

Configurations

In This Section

Adding Additional Windows DCs	page 68
Outbound Access Control on Citrix or Windows Terminals	page 69
Configuring UserAuthority Domain Equality	page 69

The configurations for a basic Outbound Access Control deployment are in Chapter 2, "Installation and Configuration" on page 31. The sections below describe the configurations for the special scenarios described in this chapter.

Adding Additional Windows DCs

You can use a basic Outbound Access Control deployment and add additional Windows DCs.

Workflow

1. Add as many additional Windows DCs as you need to your deployment.
2. Install the UAS on each new Windows DC (see "Installing and Configuring the UAS on the Windows DC" on page 43). Don't forget to create a network Object in SmartDashboard for each of the new Windows DCs.

   Configure the system to automatically install SecureAgent on each of the Windows DCs. See "Configuring SecureAgent Automatic Installation" on page 50.
3. Configure the system to automatically install SecureAgent from each of the Windows DCs. See "Configuring SecureAgent Automatic Installation" on page 50
4. Verify that an SSO rule is defined (see "Adding an SSO Rule" on page 25).

Outbound Access Control on Citrix or Windows Terminals

To deploy Outbound Access Control on Citrix MetaFrame Servers or Windows Terminal Services you must install UAS on the Citrix or Windows Terminal Services. UserAuthority gets the identification information from the connection, therefore SecureAgent is not required for Citrix.

1. Install the UAS on the VPN-1 Pro gateway. See "Installing and Configuring UAS on VPN-1 Pro" on page 31
2. Install and configure a UAS on each of the Citrix MetaFrame Servers or the Windows Terminal Services. See "Installing and Configuring the UAS on the Windows DC" on page 43.
3. From the SmartDashboard **Security** tab, configure an SSO rule. See "Adding an SSO Rule for Citrix MetaFrame or Windows Terminal Services" on page 29.

Configuring UserAuthority Domain Equality

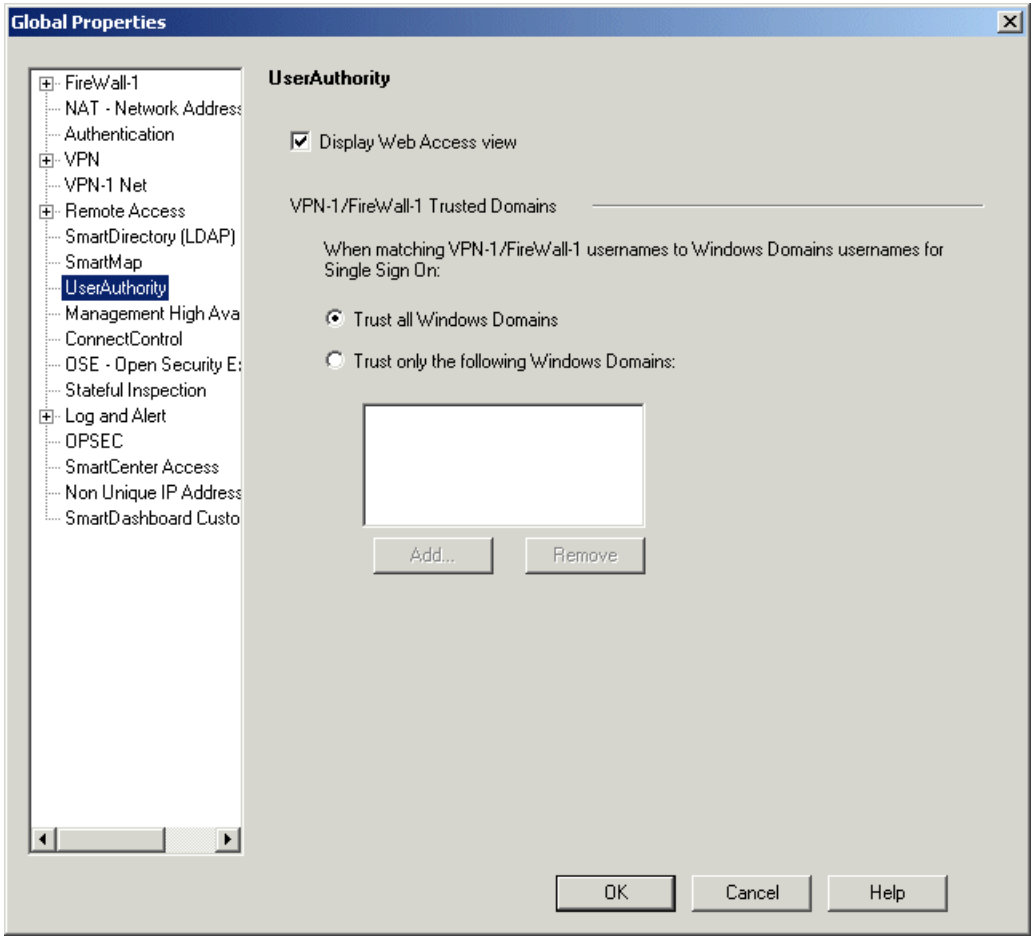
This section describes how to change the domain equality options. For more information on domain equality, see "Scenario - An Organization Using Multiple Domains" on page 66.

To configure domain equality:

1. In the SmartDashboard **Policy** menu, select **Global Properties**.

   The **Global Properties** window is displayed.
2. From the tree in the left pane of the window, click **UserAuthority**. The **UserAuthority** properties window is displayed.

Figure 3-7 UserAuthority Global Properties Window



3. Select one of the following options:

   - **Trust all Windows Domains**: This indicates that the firewall matches the user name no matter what comes before it. Therefore, a user is recognized from any domain.



Note - Trust all Windows Domains is selected by default.

   - **Trust only the following Windows Domains**: This options allows you to indicate specific window domains to authenticate.

     To enter a domain name(e.g., Finance_Gurus), click **Add** and enter the Windows Domain name.

     To remove a domain name, select the domain name and click **Remove**.

4. Click **OK**.
5. Click **Install Policy** from the tool bar to install the Policy.

Chapter 4

User Management in UserAuthority

In This Chapter

Overview	page 74
Managing Users and Groups	page 75
Using a Local Check Point Database	page 77
Using an External Database	page 78
Using the Windows User Identity	page 79

Overview

Managing users is a central part of UserAuthority because Single Sign-On (SSO), authorization and authentication rules are dependant on defining users and User Groups in the system.

UserAuthority provides SSO, user authorization and auditing to users in an enterprise by identifying the user. If the system does not have a database of users, UserAuthority cannot carry out these functions.

Users and User Groups can be managed in three ways:

- Using a local Check Point database in SmartDashboard (see "Using a Local Check Point Database" on page 77).
- Using an external database (for example, Radius, LDAP) (see "Using an External Database" on page 78).
- Using the user's identity in the Windows domain (see "Using the Windows User Identity" on page 79).

Managing Users and Groups

In This Section

Users in UserAuthority page 75
User Groups in UserAuthority page 75

Users in UserAuthority

In order for UserAuthority to perform SSO, identification and authorization, it must have a database of users defined in the system. One of the advantages of UserAuthority is that it uses the same databases that are used by VPN-1 Pro, including LDAP databases or the VPN-1 Pro database. There is no need to create and define separate user databases and groups for each security-related module in the network.

User Groups in UserAuthority

User Groups are important in UserAuthority because the SSO for VPN-1 Pro policy can be defined in terms of User Groups. User groups are created or used for the following reason:

- **Defining client and session authentication actions in the VPN-1 Pro security policy.** UserAuthority provides SSO for VPN-1 Pro authentication by creating an SSO rule in the VPN-1 Pro policy. User groups can be as part of a rule to indicate which users can access the requested resource. For more information, see Chapter 3, "Outbound Access Control"".

Access control is based on User Groups and not individual users. Groups can be defined on the VPN-1 Pro local database, on an LDAP server, or based on Windows User Identity. For information on how to use these options, see:

- "Using a Local Check Point Database" on page 77
- "Using an External Database" on page 78
- "Using the Windows User Identity" on page 79

Using a Local Check Point Database

Check Point's user management solution utilizing a local database is intended for small deployments, such as:

- Computer labs
- Enterprises with a small number of users and computers

Users and User Groups in a local Check Point database are managed using SmartDashboard. The SmartDashboard hierarchical tree structure allows you to define users, User Groups and Administrators by right clicking the correct object tree.

The Check Point local database is created on the Check Point SmartServer and is transferred to the VPN-1 Pro gateway when the policy is installed. In order to use this database with UserAuthority, you must be sure that the policy is installed on the VPN-1 Pro gateway where the UserAuthority Server (UAS) is installed. For information on creating and defining various users and groups in SmartDashboard, see the *Check Point SmartCenter Guide*.

Using an External Database

If you already have an external user management infrastructure in place, such as an LDAP, it is usually easier to use the existing user databases. UserAuthority and VPN-1 Pro support LDAP technology and use existing LDAP servers to obtain user information for authentication. If you have a large user account, it is recommended to use an LDAP system. Another advantage of using an LDAP database is that the information is external and can be shared by other systems.

VPN-1 Pro, UserAuthority, and all other components in the Check Point security system act as LDAP clients. The SmartCenter Server (for example, SmartDashboard, SmartView Tracker, SmartView Status) can manage the information on the LDAP server. UserAuthority can use all of the information on the LDAP server to create policy for the UAS.

To use the information on the LDAP database, you must define the LDAP server in SmartDashboard (see the section on defining an LDAP server in the *Check Point SmartCenter Guide)*. Once the LDAP server is defined in SmartDashboard and connected, all users in the database can be used by UserAuthority. To use groups with UserAuthority over an LDAP server, you must create LDAP groups in SmartDashboard. This is done by creating a new LDAP group from the Users tree.

LDAP groups are defined according to the LDAP tree (or subtree), a DN prefix, or by a defined filter.

For a detailed description on how to manage Users with an LDAP server, see the *Check Point SmartCenter Guide*.

Using the Windows User Identity

In This Section

Users in the Windows Domain page 79
Configuring UserAuthority to Recognize Windows User Groups page 79

Users in the Windows Domain

UserAuthority can take advantage of user databases in the Windows environment to identify users. This saves a costly migration from Windows to the VPN-1 Pro database or to LDAP. Windows groups in UserAuthority are used for Outbound Access Control and require the installation of UAS on the Windows Domain Controller (DC).

To use Windows groups with UserAuthority, you must define groups in SmartDashboard according to the Window's group name. UserAuthority imports the group information from Windows and matches it to the groups you defined in SmartDashboard. For information on how to define these groups, see "Configuring UserAuthority to Recognize Windows User Groups" on page 79.

You can maintain a database on the VPN-1 Pro Firewall and also use the Windows domain to identify users. In this case, you must ensure that user identity in the Windows Domains and in VPN-1 Pro are the same.



Note - By default, UserAuthority identifies the user from the Windows systems by the user name only, without the user's domain.

Configuring UserAuthority to Recognize Windows User Groups

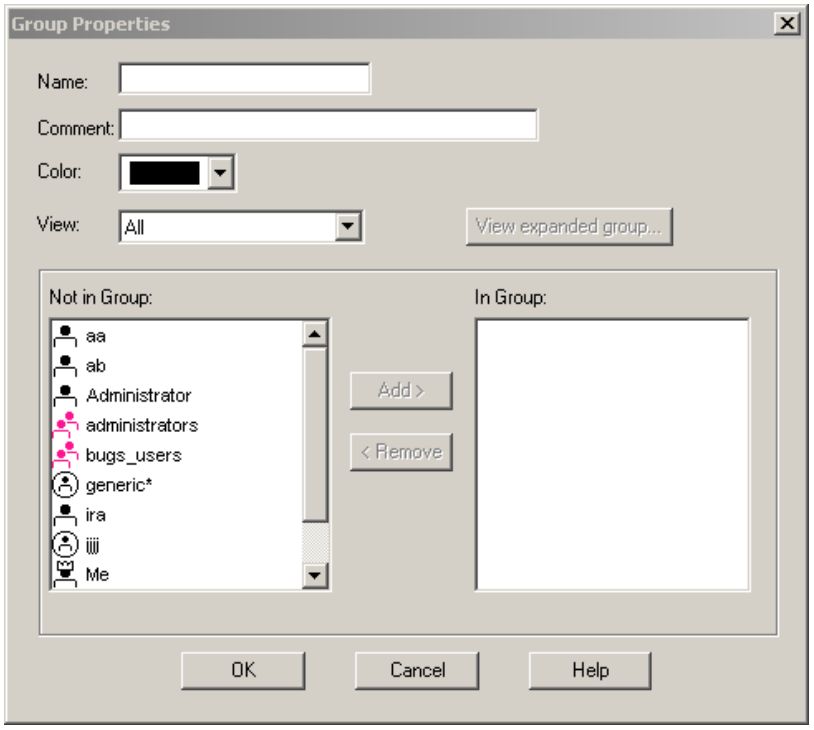
Windows groups are defined in a Windows Domain database, such as ActiveDirectory. In order to use Windows groups, a UAS must be installed on a Windows DC.

To enable UserAuthority to use Windows Domain groups for Outbound Access Control, you must define a users group with the name `WIN_<Domain Name>_<Windows Group Name>` by doing the following:

1. From the **Users and Administrators**, right click the **User Groups** object and select **New Group**.

   The **Group Properties** window is displayed.

Figure 4-1 Group Properties Window



2. In the **Name** field, write a name in the form of `WIN_<Domain Name>_<Windows Group Name>`, for example `WIN_INTUSERS_Managers`.
3. Click **OK** to close the window. The new group appears under the User Groups object.
4. Install the policy.

Chapter 5
Auditing in UserAuthority

In This Chapter

Overview page 82
Using Logs for Auditing page 83
Configuring UserAuthority for Auditing page 88

Overview

UserAuthority uses the SmartView Tracker, Check Point's advanced tracking tool, to enable auditing of both UserAuthority Server (UAS).

Auditing enables you to:

- Troubleshoot security issues.
- Gather information for legal purposes.
- Generate reports and analyze traffic patterns.
- Generate logs in specific instances, for example, if the system is being attacked.

Auditing in UserAuthority provides the following advantages:

- Auditing user requests (permitted and not permitted) for Outbound Access Control.
- Auditing successful and unsuccessful UserAuthority Identification and Authentication queries.
- Auditing authenticated outbound requests, enabling you to keep track of all outbound traffic from the local network.

Using Logs for Auditing

In This Section

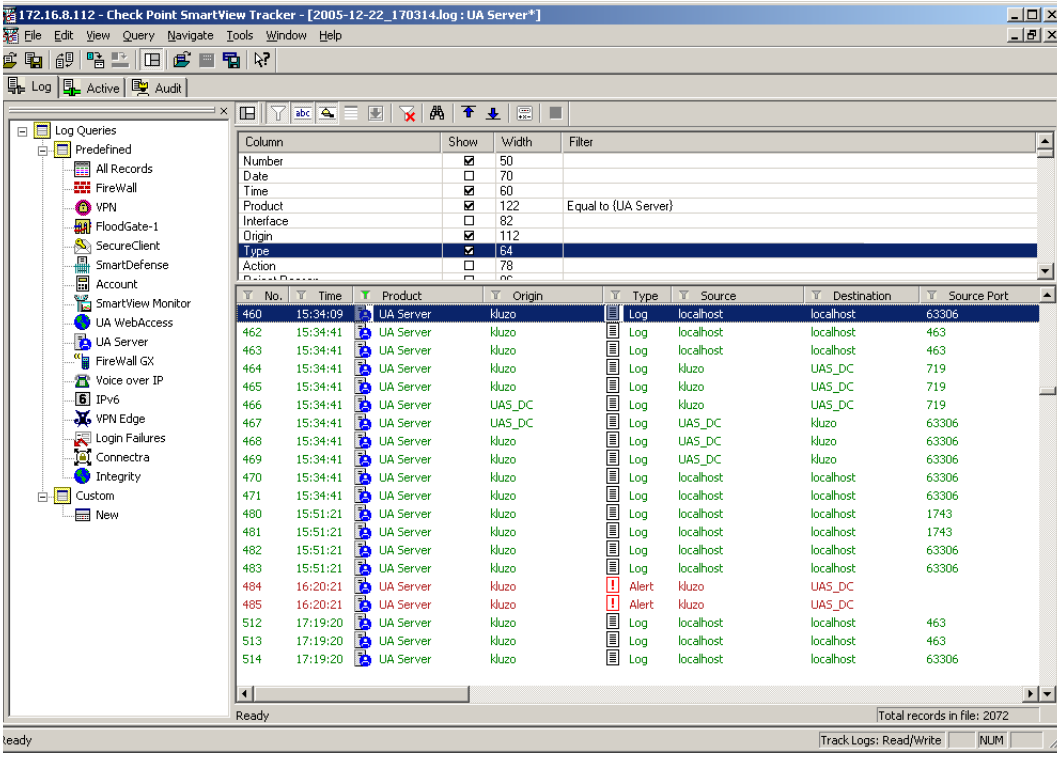
Auditing Outbound Traffic Using UserAuthority Outbound Access Control page 84

Auditing in UserAuthority is performed using logs and alerts. UserAuthority and can be configured to create logs for specific activities. These logs provide comprehensive information on network activity. You can analyze the information provided by the logs to get a complete picture of your UserAuthority system.

The SmartView Tracker provides an interface that displays all of the logs generated in the system. Each log contains specific information about network activity. F

Figure 5-1 shows the SmartView Tracker interface.

Figure 5-1 SmartView Tracker



The SmartView Tracker display is divided into the following panes:

1. **Query Tree pane**: This pane displays pre-defined and custom queries. The queries in this pane that are important to auditing in UserAuthority are **FireWall-1,** and **UA Server**. Double clicking these queries displays logs for the selected products only.
2. **Query Properties pane**: This pane allows you to select and customize the properties displayed for each log record. To display a field, find the field name and select the adjacent checkboxes.
3. **Records pane**: This pane displays all the log records and the log information for each one. Double click on a record to open a window that displays the log information.

Another important feature of the SmartView Tracker is its filtering ability. Each query acts as a filter. Double click **UA Server** in the **Query Tree** to filter the **Records** pane to display only logs from UserAuthority Server. You can also filter other parameters. For example, filtering according to the **UA Session ID** enables you to display only the records from a single session, making it easier to track the activity for that session.



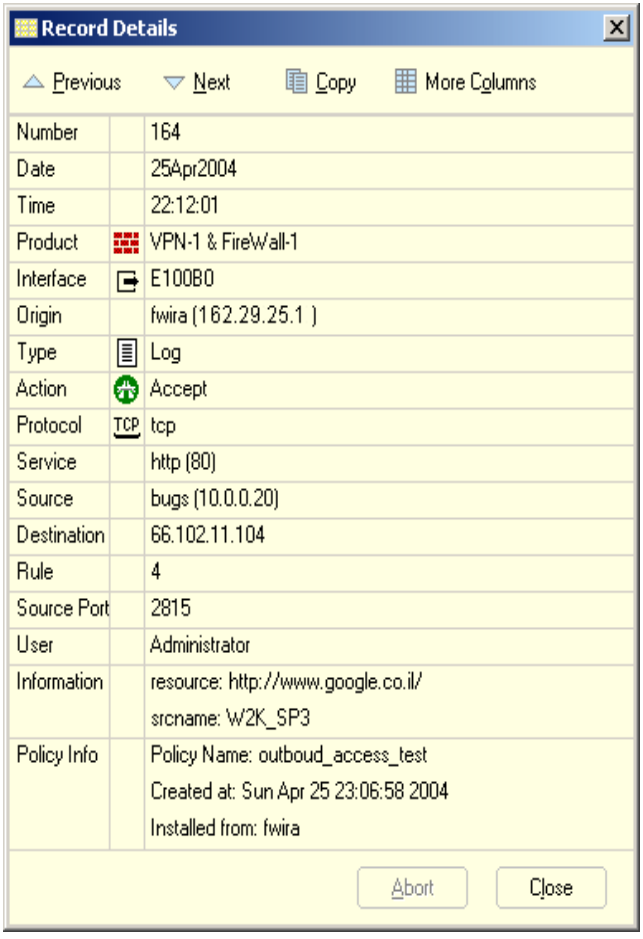
Note - For details on how to use the SmartView Tracker, see the *Check Point SmartCenter Guide.*

Auditing Outbound Traffic Using UserAuthority Outbound Access Control

When deploying a local network for Outbound Access Control, you can use the VPN-1 Pro logs to show which users are accessing which external resources. Although these logs contain many different fields, the fields that provide the information necessary to audit user activities are the **User**, **Destination**, and **Information** fields. Figure 5-2 shows a **FireWall-1 Record Details** window.

If you do not see one of those fields, you should customize your view in the Query Properties pane. See "SmartView Tracker" on page 83.

Figure 5-2 FireWall-1 Record Details Widow



The **User**, **Destination**, and **Information** fields in Figure 5-2 show the following:

- **User**: The user is identified as Administrator. This is the name of the user in the Windows domain. SecureAgent identifies the user at the Trusted Identification Point (TIP) according to the credentials entered when the user first authenticates to the Windows Domain Controller (DC).
- **Destination**: The **Destination** field indicates the IP address (66.102.11.104) for the requested external resource. This is used to identify the requested external resource. This field can also display DNS entries.
- **Information**: This field provides special information, including information on resources that are configured in the VPN-1 Pro SSO policy. You can configure an SSO rule to display the URL in the logs by creating a resource in

SmartDashboard that obtains the Fully Qualified Domain Name for the requested resource. See "Displaying the Resource Name in the Information Field" on page 86.

Displaying the Resource Name in the Information Field

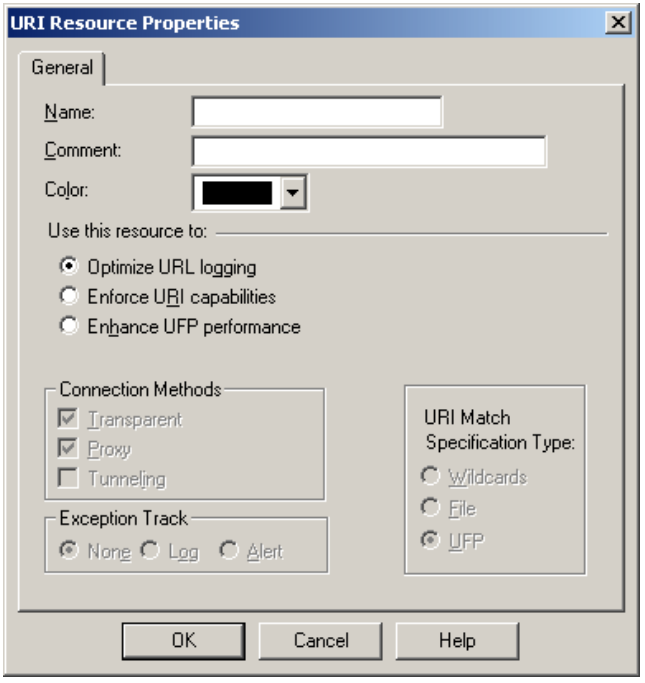
To display the name of the URL in the **Information** field, you must create a resource in SmartDashboard.

To create a resource in SmartDashboard:

1. In the Resource tree, right click the Resource and select **New -> URI...**.

   The **URI Resource Properties** window is displayed.

Figure 5-3 URI Resource Properties Window



2. In the **Name** field, enter a name for the URI resource.
3. In the **Use this resource to** section, select **Optimize URL logging**.
4. Click **OK** to close the window.

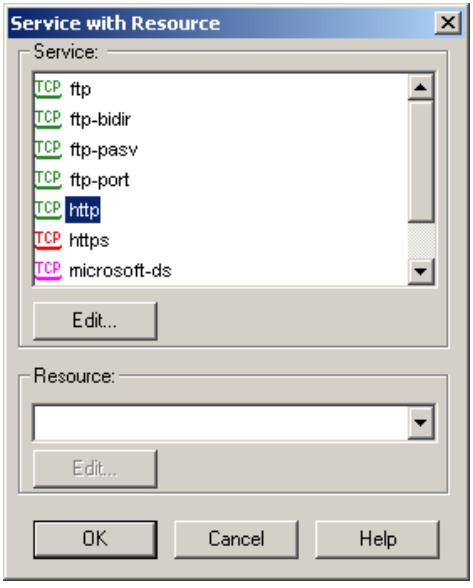
   The URI resource appears in the Resource tree under URI.

When you create your SSO policy, you need to configure a Service with the resource you created in the **Service** field of the **Security** tab.

To configure a service with the resource:

1. In the **Security** tab of the SmartDashboard, right click on the **Service** field and select **Add With Resource**. The **Service with Resource** window is displayed

Figure 5-4 Service with Resource Window



2. From the **Service** list, select the required service. The **Resource** drop-down list is enabled.
3. Select the URI Resource you created.
4. Click **OK** to close the window.

Figure 5-5 SSO Policy Configure to Display the URL

SOURCE	DESTINATION	VPN	SERVICE	ACTION	TRACK	INSTALL ON
Sales_Managers@A	Any	Any Traffic	http->URL	Client Auth	Log	meravl

In **Figure 5-5**, the **Service** field indicates that requests from the Sales Managers group are accepted with **Client Authentication** for the HTTP service with a URI resource named “URL”.

Configuring UserAuthority for Auditing

In This Section

Configuring Auditing of Requests for External Resources page 88

This section describes how to configure UserAuthority to create logs that can be used for auditing.

Configuring Auditing of Requests for External Resources

Auditing of requests to external resources is performed using VPN-1 Pro logs. For a description on how to use these logs to audit outbound traffic, see "Auditing Outbound Traffic Using UserAuthority Outbound Access Control" on page 84.

To configure the system to create VPN-1 Pro logs:

1. In SmartDashboard, click the **Security** tab.
2. In the **Security** tab, create a basic Outbound Access Control rule (see "Adding an SSO Rule" on page 25).
3. To configure logging, right click **Track** and then select **Log**.
4. Save the policy and install it on the firewall.

Chapter 6

High Availability and Load Balancing

In This Chapter

Overview page 90

Overview

In This Section

High Availability page 90
Load Balancing page 90
High Availability and Load Balancing in UserAuthority page 91

High Availability

High availability indicates that a product or system is available at almost all times. An accepted standard in high availability is called *five nines*, which indicates that a product system is available 99.999% of the time. Although this standard is rarely reached, a system or product should come close to this benchmark to be considered highly available.

One way to ensure high availability is to use a cluster of two or more computers or servers. Each computer in the cluster performs the same job, however only one of the computers is active at a given time. All system updates are made to all of the computers in the cluster. If the main computer goes offline for any reason, another computer containing identical information is available to take its place - with no adverse effect on system performance. This also allows system administrators to perform maintenance tasks on the main computer without impacting on system availability.

Load Balancing

Distributing requests in high-traffic Web sites is called load balancing. Load balancing plays an important role in high availability because it ensures that a server will not go offline due to excessive traffic. Load balancing uses clusters to distribute traffic between servers. Requests are received by a managing computer that balances the traffic load. All of the computers in the cluster are active computers and hold identical information. The “balancing” computer receives the request and sends it to one of the computers in the cluster based on pre-configured criteria. In most cases, the configuration aims to evenly distribute traffic between the available servers.

High Availability and Load Balancing in UserAuthority

The UserAuthority Server (UAS) can be configured to provide both high availability and load balancing. Clusters can be set up and configured to ensure that network traffic to the security system is handled efficiently and virtually without interruption.

The following areas can be configured or set up to provide high availability and load balancing:

- UAS on a Windows Domain Controller (DC)
- UAS on the VPN-1 Pro gateways

Using Multiple Windows DCs

UASes that are installed on the Windows DC do not contain any dynamic information that must be updated. Each time a query is made to the UAS, it sends the query to the user's desktop to receive the user identity (through SecureAgent).

In this case, high availability is easily achieved by installing the UAS in more than one location on the same Windows domain. No special configuration is required. The default Shared Identity option automatically queries each UAS until the user's identity is established. If one UAS in the Windows domain is offline, the others will still be queried so that the user identity can be obtained.

Using a VPN-1 Pro Cluster

In This Section

Using VPN-1 Pro Clusters page 93
Synchronizing the Credentials Manager page 93

Using VPN-1 Pro Clusters

High Availability for the UAS on the VPN-1 Pro gateway is provided when there is more than one gateway. In this situation, the network is configured with VPN-1 Pro clusters. In each cluster, one VPN-1 Pro gateway is the primary gateway to which all requests are automatically routed. When the primary gateway is offline, requests go to the gateway that is designated as the secondary gateway. For more information on VPN-1 Pro clustering, see the Check Point *Firewall Guide*.

To create high availability, install the UAS on each VPN-1 Pro machine in the cluster. One component of the UAS, the Credentials Manager, contains information that must be synchronized. For this purpose, UserAuthority provides a script called db_sync that ensures that the Credentials Manager on each UAS contains the same information at all times. Therefore, if one UAS goes offline and another takes over, users that are already signed on to the system can still be authenticated automatically.

Synchronizing the Credentials Manager

When UserAuthority identifies a user, it inserts the user's credentials into requested applications using information stored in the Credentials Manager (see "Mapping User Identity to Application Information by UserAuthority" on page 106 for information about the Credentials Manager). When multiple UASes are deployed in a VPN-1 Pro cluster for high-availability purposes, the Credentials Managers for each UAS must be synchronized. This ensures that the user information is available if there is a failover from the primary UAS to another UAS in the cluster in the course of a user's session.

Automatic Synchronization

The UASes on the same VPN-1 Pro cluster communicate through a special communications interface. When the main Credentials Manager is updated, updates are sent through this interface to the other Credentials Managers in the cluster. However, this interface can be offline, even if all the UASes are still operational. In this case, the updates will not be made. To ensure that all updates are made, you can configure the UASes to update the Credentials Managers simultaneously through all possible communications interfaces. To do this, you must change the default settings in the netso.ini file.

To set UserAuthority to update all Credentials Managers' multiple communications interfaces on each UAS:

1. Run `uagstop`.
2. Find the `netso.ini` in the `$uagdir\conf` directory (for Windows the file is in the `UAG\conf` folder).
3. Find the line `cluster_update_chaining _only_to_main_ips =`
4. Type `true` after the (=) sign.
5. Save the file.
6. Run `uagstart`.



Note - This solution works when all UASes on the cluster are online. If a UAS is offline for any reason when an update is made, the Credentials Manager for that UAS will not be updated. In this case, you must manually update each Credentials Manager by running the `db_sync` script. For information on how to run the db_sync script, see "Using the db_sync Script".

Using the db_sync Script

You can sychronize the Credentials Managers on the same cluster by running the db_sync script. The script synchronizes Credentials Managers that are deployed with same exact information. You must run the script on the machine with the UAS that contains the Credentials Manager that needs to be updated. If there are more than two machines in the cluster, you must update each Credentials Manager individually.

To synchronize Credentials Managers:

- From the machine with the Credentials Manager that must be updated, run the script:

  ```
  db_sync <Remote Gateway's IP Address>
  ```

The IP address must be the IP address for the UAS with the Credentials Manager that has the updated information.

The following message is returned:

```
Synchronization successfully finished!
```

If a problem occurs, the following error message is returned:

```
Synchronization error. Please try again or contact Check Point Support.
Bad status received. The status is <reason for error>.
```

Chapter 7
UserAuthority CLIs

In This Chapter

UAS	page 98
uas debug	page 98
uas drv	page 98
uas reconf	page 99
uas d	page 99
uas kill	page 99
uas ver	page 99
netsod debug	page 100
netsod drv	page 100
netsod d	page 100
netsod kill	page 100
netsod simple	page 101
netsod simple kill	page 101
netsod ver	page 101
uas	page 102
cpstop	page 102
cpstart	page 102
cprestart	page 103
uagstart	page 103

UAS

Description The UAS command activates the UserAuthority Server (UAS) in NG with Application Intelligence or later.

Usage `UAS.`

uas debug

Description This command is used to activate or deactivate the debug log directory.

Usage `uas debug on`

Usage `uas debug off`

Syntax

Argument	Description
on	Writes development logs in the `UA_log.elg` directory.
off	Stops writing logs in the `UA_log.elg` directory.

Return Value
```
UAS debug already off
UAS debug already on
```

uas drv

Description This command is used to activate or deactivate a UAS on the device driver.

Usage `uas drv on`

Usage `uas drv off`

Syntax

Argument	Description
on	Loads UAS device driver.
off	Stops UAS device driver.

Comments Note that all kernel information in the UAS is swapped when running `uas drv off`.

uas reconf

Description This command reconfigures the UAS using the `netso.ini` file.

Usage `uas reconf`

Return Value `UserAuthority: Reconfiguring using netso.ini file`

uas d

Description This command initializes the UAS daemon.

Usage `uas d`

Return Value `CheckPoint UserAuthority Server is already running.`

Comments If the UAS is not running, then a list of debugging outputs is returned.

uas kill

Description This command shuts down all parts of the UAS.

Usage `uas kill`

Return Value `UserAuthority Server is going down...`

uas ver

Description This command displays the UAS version installed.

Usage `uas ver`

Return Value `This is Check Point UserAuthority(TM) Server NGX (version information) - Build 011`

Comments The version information contains the name of the version and build.

Example This is an example of a return value: `This is Check Point UserAuthority (TM) Server NGX (R61) Build 011.`

netsod

Description The netsod command activates the UAS operation in modes prior to NG with Application Intelligence.

Usage `netsod`

netsod debug

Description This command is used to activate or deactivate logging in the log directory.

Usage `netsod debug on`

Usage `netsod debug off`

Syntax

Argument	Description
on	Writes logs in the `UA_log.elg` directory.
off	Stops writing logs in the `UA_log.elg` directory.

Return Value
```
Switching UAG to debug ON
Switching UAG to debug OFF
```

netsod drv

Description This command is used to activate or deactivate UAS on the device driver.

Usage `netsod drv on`

Usage `netsod drv off`

Syntax

Argument	Description
on	Loads the UAS device driver.
off	Stops the UAS device driver.

netsod d

Description This command initializes the UAS daemon.

Usage `netsod d`

Return Value `Check Point UserAuthority Server is already Running`

Comments If the UAS is not running, then a list of debugging outputs is returned.

netsod kill

Description This command shuts down all parts of the UAS.

Usage	`netsod kill`
Return Value	`UserAuthority Server is going down...`

netsod simple

Description Turns on the netsod simple mode.

Usage	`netsod simple`
Return Value	`<there is no return value>`
Comments	`netsod simple` is a mode of operation that allows you to manually send plain text messages (queries) to the netsod daemon using telnet port 19190. If the UAS is running in simple mode, it can translate the message and send a return. UAS is active in simple mode by default. You do not need to run this command unless simple mode was turned off.

netsod simple kill

Description Turns off the netsod simple mode.

Usage	`netsod simple kill`
Return Value	`<there is no return value>`
Comments	`netsod simple` is a mode of operation that allows you to manually send plain text messages (queries) to the netsod daemon using telnet port 19190. If the UAS is running in simple mode, it can translate the message and send a return. UAS is active in simple mode by default. You do not need to run this command unless simple mode was turned off.

netsod ver

Description This command displays the UAS version installed.

Usage	`netsod ver`
Return Value	`This is Check Point UserAuthority (TM) Server <version information>`
Comments	The version information contains the name of the version and build.
Example	This is an example of a return value: `This is Check Point UserAuthority (TM) Server NG Feature Pack 3 (R 55) Build 047.`

uas

Description This command displays the command lines and the descriptions of each command available for the UAS.

Usage
```
uas
```

Return Value
```
uas d # initialize uas daemon
uas renconf # Reconfigure UAS using netso.ini
```

Comments This return value is a list of commands and their definitions. The above return is an example of the first part of the return.

cpstop

Description This command stops all Check Point product services running on the computer.

Usage
```
cpstop
```

Return Value
```
The Check Point UserAuthority Service is stopping
The Checkpoint UserAuthority Service was stopped successfully
```

Comments This return value is followed with a similar return for all other Check Point modules installed on the machine. The second line indicates the success or failure of the request.

cpstart

Description This command starts all Check Point product services running on the computer.

Usage
```
cpstart
```

Return Value
```
The Check Point UserAuthority Service is starting
The Checkpoint UserAuthority Service was started successfully
```

Comments This return value is followed with a similar return for all other Check Point modules installed on the machine. The second line indicates the success or failure of the request.

cprestart

Description This command stops and then automatically restarts all Check Point product services running on the computer.

```
Usage          cprestart

Return Value   The Check Point UserAuthority Service is stopping
               The Checkpoint UserAuthority Service was stopped successfully

Return Value   The Check Point UserAuthority Service is starting
               The Checkpoint UserAuthority Service was started successfully
```

Comments These return values are followed with similar messages for all other Check Point modules installed on the machine. The second line indicates the success or failure of the request.

uagstop

Description This command stops the UAS installed on the computer.

```
Usage          uagstop

Return Value   The Check Point UserAuthority Service is stopping
               The Checkpoint UserAuthority Service was stopped successfully
```

Comments The second line indicates the success or failure of the request.

uagstart

Description This command starts the UAS installed on the computer.

```
Usage          uagstart

Return Value   The Check Point UserAuthority Service is starting
               The Checkpoint UserAuthority Service was started successfully
```

Comments The second line indicates the success or failure of the request.

Syntax

Chapter 8

UserAuthority OPSEC APIs

In This Chapter

Overview page 106
Programming Model page 107
Function Calls page 121
Event Handlers page 132

Overview

Check Point's OPSEC (Open Platform for Security) integrates and manages all aspects of network security through an open, extensible management framework. Third-party applications can plug into the OPSEC framework through published application programming interfaces (APIs). Once integrated into the OPSEC framework, the security aspects of these applications can be configured and managed from a central point, utilizing a single Security SmartDashboard.

For information about how to integrate third-party HTTP Proxies with Check Point UserAuthority, see "Web SSO with an Internal Proxy" on page 108.

Programming Model

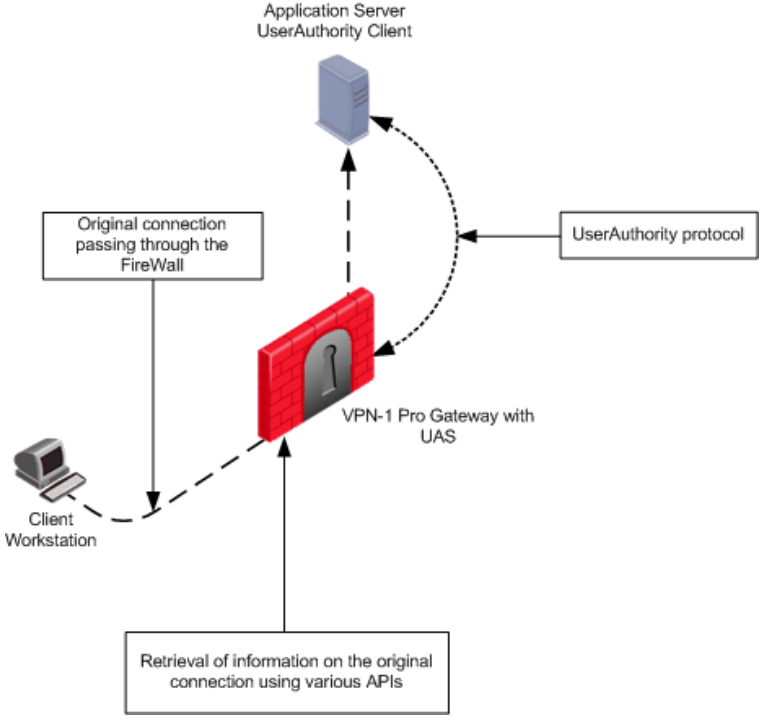
In This Section

Defining a UAA Client	page 110
Client Server Configuration	page 110
OPSEC UserAuthority API Overview	page 110

UserAuthority API (UAA) provides third-party application servers with network security information from various Check Point products, such as VPN-1 Pro, SecuRemote/SecureClient. This enables the application servers to use Check Point's security mechanisms rather than implementing their own.

Figure 8-1 illustrates the system architecture.

Figure 8-1 System Architectures





Note - If the original connection comes from a LAN, then it can be sent through a UAA server on the Domain Controller or a Citrix/Terminal services.

The desktop connecting to the application server can also use VPN-1 SecuRemote or VPN-1 SecureClient. VPN-1 SecuRemote enables PC users to securely communicate sensitive and private information over untrusted networks by encrypting and decrypting information leaving and entering their computers.

VPN-1 SecureClient enables administrators to enforce a security policy on desktops and prevents unauthorized users from taking control of authorized connections. When the SecureClient connects to the Policy Server from which it obtains its desktop policy, the Policy Server can verify the SecureClient machine's configuration and deny access to misconfigured machines.

The UAA server resides on a VPN-1 Pro Module and collects information about the connections made through that module. This information might include:

- **Connection Sign-On Information:** The network security information associated with a specific connection, including user information (user name, distinguished name (DN), and group membership), authentication scheme, and type of encryption.
- **Client Sign-On Information:** The network security information associated with a specific IP Address, including user information, authentication scheme, and whether the SecureClient's configuration is secure.
- **Credential Management Information:** The UserAuthority server can store and provide user credentials for several authentication domains (user name and password) to enable Single Sign-On and enhanced security.

The UserAuthority Server collects information about the logins made to the local network. This information might include NT domain controller logon, DHCP, and RADIUS authentications. The UserAuthority Server also keeps historical information for logging purposes, which can be accessed through the UserAuthority Administration Server.

The types of connections made through VPN-1 Pro for which information is collected are shown in Table 8-1.

Table 8-1 Network Security Information Collected by UAA

Type of Information	Collected When	Information Includes*
Connection Signon information	A connection is made through a Security Policy rule specifying User, Client, or Session Authentication.	UI, AS
	A SecuRemote connection is made.	UI, AS, ET
	A VPN connection is made.	ET
Client Signon information	A user logs onto a Client Authentication Server.	UI, AS
	SecuRemote executes a key exchange with VPN-1 Pro.	UI, AS
	A SecureClient user logs onto a Policy Server.	UI, AS, SCS

* Information includes:

- AS: Authentication Scheme
- ET: Encryption Type
- SCS: SecureClient Secure
- UI: User Information

When an application server needs information about a client or connection, the UAA client sends a query to the UAA server. This query includes a key to the connection or event. Based on this key, the UAA server retrieves the appropriate information and passes the requested data back to the client.

The UAA server and the UAA client use a separate connection for communication. This enables the application server to identify the user before responding. Communication between the UAA client and the UAA server is implemented using the OPSEC framework.

For a more detailed overview of UAA and various usage scenarios, see "OPSEC UserAuthority API Overview" on page 110.

Defining a UAA Client

The procedure for integrating a UAA Client with VPN-1 Pro can be divided into two parts:

- Configuring communication between VPN-1 Pro and the UAA Client.
- Creating queries, sending them to the UAA server, and processing the replies. This is described in detail in "OPSEC UserAuthority API Overview" on page 110.

Client Server Configuration

For information on configuring OPSEC UserAuthority clients and servers, see "Client- Server Connection" in the Check Point *VPN-1 Pro™ OPSEC API Specifications*.

For information on configuring UAA clients in the Check Point Management, see "Server Objects and OPSEC Applications" in the *Check Point SmartCenter Guide*.

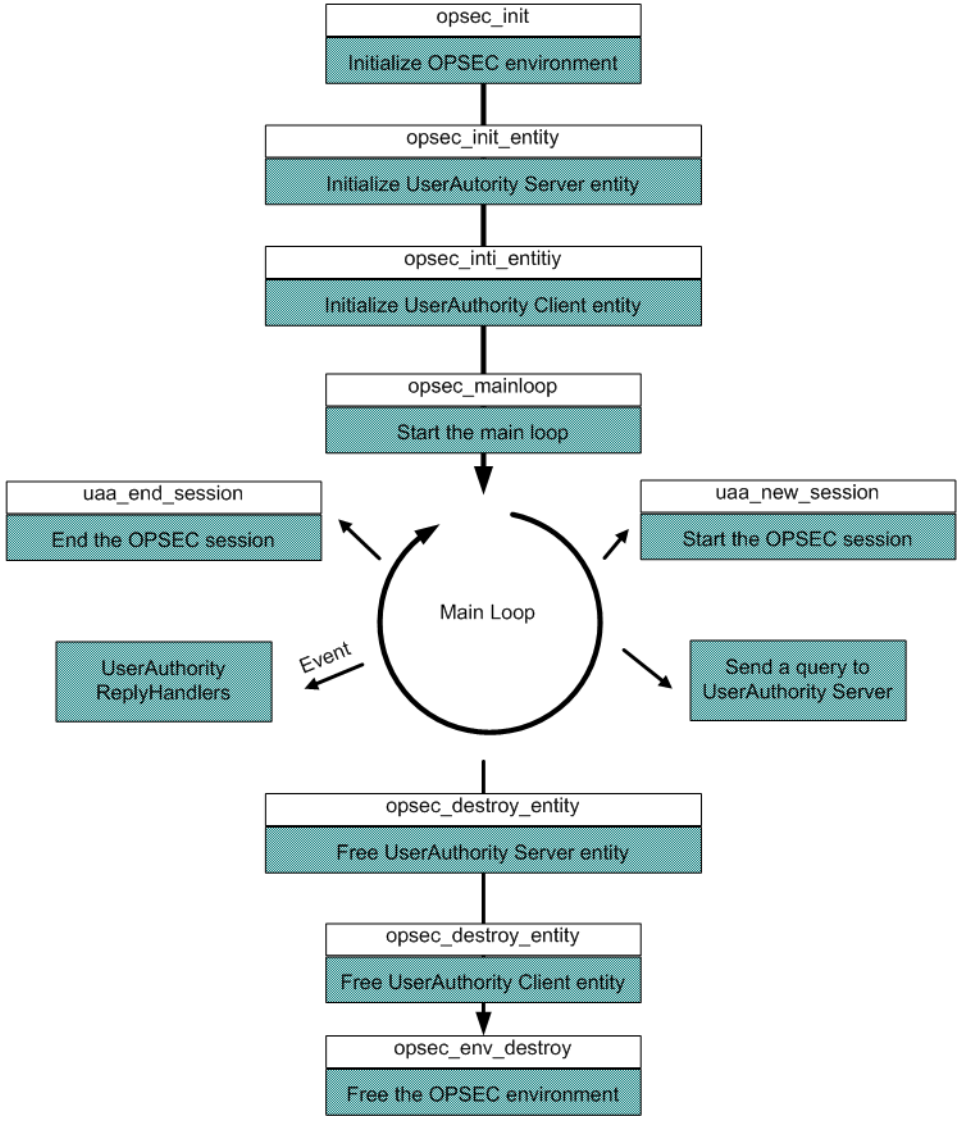
OPSEC UserAuthority API Overview

The OPSEC UserAuthority API and the OPSEC API provide functions for querying, updating and carrying out authentication against the UAA server, and processing replies.

UAA Client Application Structure

A UAA client's main function should flow as shown in Figure 8-2.

Figure 8-2 UAA Client Application Structure



When the OPSEC environment and the UAA session are initialized, a request is sent to the UAA server. The main loop then waits for a reply to arrive and processes it. Requests and replies are handled by the OPSEC UserAuthority API functions. The main loop is terminated by the underlying OPSEC level. After termination, the OPSEC entities and environment are freed.

For more information on `uaa_new_session` and `uaa_end_session`, see "Session Management" on page 121.

Event Handling

The UAA client responds to the UAA_QUERY_REPLY event handler, UAA_UPDATE_REPLY event handler, and UAA_AUTHENTICATE_REPLY event handler. These events are triggered when a reply from the server becomes available.

The response to these events is handled by the event handlers (callback functions) set in the call to opsec_init_entity for the client entity. These callbacks are set using the attributes listed in Table 8-2

Table 8-2 opsec_init_entity - UAA Entity Type Values

Value	Type	Meaning
UAA_QUERY_REPLY_HANDLER	handler	The event handler for the UAA_QUERY_REPLY.
UAA_UPDATE_REPLY_HANDLERS	handler	The event handler for the UAA_UPDATE_REPLY.
UAA_AUTHENTICATE_REPLY_HANDLER	handler	The event handler for the UAA_Authenticate_REPLY.

For more information on opsec_init_entity, see the *OPSEC API Specifications*.

Requests

A UAA request has two parts:

- Key: This is used by the UAA server to identify the appropriate connection.
- Request: This is used by the requested user and/or connection information.

Both the key and the request have one or more assertions. Each assertion has a type and a value, both of which are strings (char *).

Request Implementation

The uaa_assert_t data structure is used to pass key assertions and request assertions from the UAA client to the UAA server.

Table 8-3 shows the API functions that handle UAA requests.

Table 8-3 Request Handling Functions

Function Name	Description
UAA_send_query	Sends a query to the UAA server.
UAA_short_query	Cancels a query to the UserAuthority server
UAA_send_update	Sends an update to the UserAuthority server.
UAA_send_authenticate_request	Sends an authentication request to the UserAuthority server.

Key Assertions

Key assertions are the input to the UserAuthority server for each request. They determine the behavior of the server. Each of the different commands has a different set of key assertions. Table 8-4 shows the key assertion types and values.

Table 8-4 Key Assertions Types and Values

Command	Key Type	Key Value
Query	src	The IP address of the connection's source.
	s_port	The port number of the connection's source.
	dst	The IP address of the connection's destination.
	d_port	The port number of the connection's destination.
	ipp	The IP protocol. This assertion is optional. By default, the IP protocol is assumed to be 6 (TCP).
	snid	The Check Point session ID, a unique string stored in the HTTP_CP_SESSION_ID environment variable of the *UserAuthority Overview*.
	uid	Used for credential management queries. It specifies the username whose credentials are requested.
Update	src	The IP address of the connection's source.

Table 8-4 Key Assertions Types and Values

Command	Key Type	Key Value
	`uid`	Used for credential management updates. It specifies the username whose credentials are updated.
Authenticate	`uid`	The username to authenticate.
	`password`	The password of the user to be authenticated.

Request Assertions

Request assertions specify the information to be retrieved from the UAA server and designate how this information should be returned.

A request assertion includes a request type specifying the data to be retrieved from the UAA server (possible request types are shown in Table 8-5) and the following value:

- "*" if the reply may include multiple values corresponding to the specified type.

  Currently only used for:

  - the group assertion

- user_info/all_auth_domains_available assertion.

Table 8-5 Request Assertion Type

Command	Assertion Type	Meaning
Query	user	The ID used for authentication.
	dn	The DN (LDAP distinquished name) of the user.
	client_ip	The client IP address, which may be different from the connection's source if: • The client has undergone Network Address Translation (NAT), *or* • The connection has been redirected through a VPN-1 Pro Security Server. This attribute is returned only if: • The UAA request is included in the connection information assertion (e.g., "src", "s_port", "dst", "d_port" and "ipp"). • The connection specified in the request is passed through VPN-1 Pro.
	scheme	The type of authentication.
	group	The VPN-1 Pro groups to which the user belongs.
	enc	The type of encryption.
	scv	Indicates that the machine running SecureClient has been verified by the Policy Server running on the same machine as the UAA server.
Query	logon_time	Used to allow a client to query for a session's logon time or to include the logon time in the scope of a query.
	logoff_time	Used to allow a client to query for a session's logoff time or to include the logoff time in the scope of a query.

Table 8-5 Request Assertion Type

Command	Assertion Type	Meaning
	auth_domain/<name corresponding to authentication domain>/user	Used for credential management queries. The VPN-1 Pro User's username in the selected authentication domain.
	user_info/<name corresponding to authentication domain>/password	Used for credential management queries. The password of the VPN-1 Pro user in the selected authentication domain.
	auth_domain/all_auth_domains_available	Used for credential management queries. The reply returned for this query includes all the information stored by the credential manager for the associated user. **Note -** In order to use this type of query, use the Credential Management Web page configuration. See the "The Credentials Manager Web GUI - UA Settings" on page 15 for more information.
	win_group=*	Used to define Windows domain groups.
Update	auth_domain/<name corresponding to authentication domain>/user	Used for credential management updates. The user name of the VPN-1 Pro user in the selected authentication domain.
Update	user_info/<name corresponding to authenticatin domain>/password	Used for credential management updates. The password of the VPN-1 Pro user in the selected authentication domain.
Authenticate	user	The authenticated username.
	action	Action stage in the authentication process (i.e., failure, success, more information needed).
	message	Message suitable for the action to be taken.
	group	The VPN-1 Pro groups to which the user belongs.
	dn	The DN (LDAP Distinguished Name) of the user.
	scheme	The type of authentication.

Each request is uniquely identified by a request ID returned by the call to one of the `uaa_send_xxx` functions. The request ID is used as a parameter to be passed to other functions, for example, uaa_abort_query.

The request ID is not valid in the following cases:

- After the last reply has arrived to the user's event handler function
- After a query has been aborted by calling uaa_abort_query
- After the event handler has been called because the request has timed out (that is, the timeout specified in uaa_send_xxx expired).

The result of using the request ID in any of these cases is undefined.

Replies

A reply consists of reply assertions corresponding to the request assertions in the request. Each reply assertion consists of a type and a value, both of which are strings (char *). The reply type is identical to the corresponding request type. If there is no value corresponding to a given request type, then the assertion is not returned. If a reply type has more than one corresponding value, and the corresponding request assertion had a value of "*", then the reply contains one assertion for each value. That is, the reply contains several reply assertions of the same type.

Table 8-6 shows the assertion types and values.

Table 8-6 Reply Assertions Types and Values

Type	Reply Value
user	The user ID (name) used for authentication.
dn	The DN (LDAP Distinquished Name) of the user. Null if the user does not have a DN. This attribute can be used by LDAP-aware applications and is available only if the user entry was taken from an LDAP Server.
client_ip	The IP address of the UAA Client (which may be different than the source of the connection if the connection has been redirected through a VPN-1 Pro Security Server).
win_group	Used to define Windows domain groups.

Table 8-6 Reply Assertions Types and Values

Type	Reply Value
scheme	The type of authentication: • `NULL` - The connection is not authenticated • "`Unknown`" - exact details unknown (e.g. RADIUS, TACACS) • "`IP Based`" - such as UAM • "`Fixed password`" - Pre-shared secret, OS, VPN-1 Pro, LDAP • "`One Time Password`" - S/Key • "`Token`" - SecurID, Axent • "`Certificate`" - PKI.
group*	The VPN-1 Pro groups to which the user belongs. **Note**: Because groups are defined in the VPN-1 Pro database, LDAP groups may appear as "external groups".
enc	The type of encryption: • `NULL` - either the connection did not pass through VPN-1 Pro, or not enough information is available on the connection • "`PLAIN`" - no encryption • "`ENCRYPTED`" - encrypted, but the exact details are unknown • "`EXPORT`" - such as RC4/40 • "`DOMESTIC`" - such as DES • "`STRONG`" - such as Triple DES
scv	"`1`" if the SecureClient is currently connected to a Policy Server running on the same machine as the UAA Server. "`0`" in all other situations.

The UAA server uses the `uaa_assert_t` data structure to return reply assertions to the UAA client. The `uaa_assert_t` data structure is passed to the UAA client as one of the arguments to the event handlers. The structure is automatically freed when the event handlers return.

Connection-Based Vs. IP-Based Information in Queries

Table 8-7 UserAuthority Queries

UAA Queries on:	Use these connection key assertions	UAA Server Returns:			
		User Info.	Authentication Scheme	Encryption Type	SecureClient Secure
A connection	One of: • src s_port dst d_port and ipp • snid	user group dn client_ip	scheme	enc	
An IP address	src	user group dn client_ip win_group	scheme		scv



Tip - For detailed information on advanced UAA queries, contact OPSEC SDK Technical Services.

UAA Assertions Structure Functions

Table 8-8 shows API functions that enable you to step through the assertions in a UAA assertions structure.

Table 8-8 API Functions for Iterating through Assertions

Function Name	Description
`uaa__assert_t_iter_create`	Creates an iteration object for UserAuthority assertions.
`uaa__assert_t_iter_get_next`	Sets the iterator to the next assertion in the assertions structure.
`uaa__assert_t_iter_reset`	Resets the iterator to the first assertion.
`uaa__assert_t_iter_destroy`	Destroys the assertions iterator and frees its memory.

Processing Error Codes

Error codes can be processed using the API functions shown in Table 8-9.

Table 8-9 API Functions to Process Error Codes

Function Name	Description
`uaa__error_str`	Converts an error value to a string.

Session Management

Several queries and updates can run on a single session, but each authenticate command should run on a separate session.

Function Calls

In This Section

Session Management page 121
Assertions Management page 122
Managing Queries page 125
Managing Updates page 126
Managing Authentication Requests page 127
Assertions Iteration page 128
Managing UAA Errors page 130
Debugging page 131

This section describes the functions provided by the OPSEC UserAuthority API.

Session Management

The Session Management function calls the start and end OPSEC session APIs. Function prototypes are defined in the `uaa_client.h` file and include:

- "uaa_new_session" on page 121
- "uaa_end_session" on page 122

uaa_new_session

Description: `uaa_new_session` initializes an OPSEC session between the UAA client and the UAA server.

Usage: `OpsecSession * uaa_new_session( OpsecEntity *client, OpsecEntity *server);`

Arguments

Table 8-10 uaa_new_session Arguments

Arguments	Meaning
`client`	A pointer to the Client entity as returned by `opsec_init_entity`.
`server`	A pointer to the Server entity as returned by `opsec_init_entity`.

Return Values: Pointer to the new session, if successful, or `Null`.

uaa_end_session

Description: `uaa_end_session` ends the OPSEC session. The UAA client must call this function to correctly terminate the information exchange with the UAA server.

Usage: `void uaa_end_session (OpsecSession *session) ;`

Arguments:

Table 8-11 uaa_end_session Arguments

Arguments	Meaning
`session`	A pointer to the OPESEC session as returned by `uaa_new_session`.

Return Values: None

Assertions Management

The Assertions Management functions create, build, copy and destroy UAA assertions. Unless otherwise specified, the function prototypes are defined in the file `uaa.h`. They include:

- "uaa_assert_t_create" on page 122
- "uaa_assert_t_add" on page 122
- "uaa_assert_t_duplicate" on page 123
- "uaa_assert_t_destroy" on page 123
- "uaa_assert_t_compare" on page 124
- "uaa_asser_t_n_elements" on page 124

uaa_assert_t_create

Description: `uaa_asseret_t_create` creates a `uaa_aassert_t` data structure.

Usage: `uaa_asseret_t * uaa_asseret_t_create ();`

Arguments: There are no arguments to this function.

Return Values: Pointer to `uaa_asseret_t` structure, if successful, or `Null`.

uaa_assert_t_add

Description: `uaa_asser_t_add` adds a request assertion to the specified UAA assertions.

Usage: `int uaa_assert_t_add( uaa_assert_t *asserts, char *type, char *value);`

Arguments

Table 8-12 uaa_assert_t_add Arguments

Arguments	Meaning
`asserts`	A pointer to the `uaa_asser_t` structure containing the UAA assertions.
type	The type of assertion to be added. For more information, see "Requests" on page 112.
`value`	The value of the assertion to be added. For more information, see "Requests" on page 112.

Return Values:

- Successful - (0)
- Not successful - (-1)

uaa_assert_t_duplicate

Description: `uaa_asser_t_duplicate` creates a copy of the specified UAA assertions.

Usage: `uaa_assert_t * uaa_asser_t_duplicate( uaa_assert_t *asserts);`

Arguments

Table 8-13 uaa_assert_t_duplicate Arguments

Arguments	Meaning
`asserts`	A pointer to a `uaa_asser_t` structure.

Return Values: Pointer to the new copy of the session, if successful, or `Null`.

uaa_assert_t_destroy

Description: `uaa_asser_t_destroy` destroys the data structure containing the UAA assertions and frees its memory.

Usage: `void uaa_assert_t_destroy( uaa_assert_t *asserts);`

Arguments

Table 8-14 uaa_assert_t_destroy Arguments

Arguments	Meaning
asserts	A pointer to a uaa_asser_t structure.

Return Values: None.

uaa_assert_t_compare

Description: uaa_asser_t_compare compares two assertion structures. The user can specify a list of types to ignore.

Usage: int uaa_assert_t_compare(uaa_assert_t *a, uaa_assert_t *b, char **ignore_list);

Arguments

Table 8-15 uaa_assert_t_compare Arguments

Arguments	Meaning
a	A pointer to a uaa_asser_t structure.
b	A pointer to a uaa_asser_t structure.
ignore_list	A pointer to the Server entity as returned by opsec_init_entity.

Return Values: 0 if equal, a non-zero value if not equal.

uaa_asser_t_n_elements

Description: uaa_asser_t_n_elements returns the number of assertions in the object.

Usage: int uaa_assert_t_n_elements(uaa_assert_t *asserts);

Arguments

Table 8-16 uaa_assert_t_n_elements Arguments

Arguments	Meaning
asserts	A pointer to a uaa_asser_t structure.

Return Values: Number of assertions in the structure, if successful, or a negative value.

Managing Queries

The following Query Management functions are available:

- "uaa_send_query" on page 125
- "uaa_abort_query" on page 126

uaa_send_query

Description: `uaa_send_query` sends a query to the UAA server. The function usage is defined in the `uaa_client.h` file.

Usage: `int uaa_send_query ( OpsecSession *session, uaa_assert_t *query, void *opaque, unsigned int timeout);`

Arguments

Table 8-17 uaa_send_query Arguments

Arguments	Meaning
`session`	A pointer to the OPSEC session.
query	A pointer to the `uaa_asser_t` structure containing the UAA query.
opaque	A general purpose pointer to be passed directly to the reply handler.
`timeout`	The number of milliseconds before a UAA request times out. If a reply is not available by this time, the event handler for the event is called with the appropriate status.

Return Values:

- Successful: A unique query ID different than (-1)
- Not Successful (-1)

Note - The query ID is not valid in any of the following cases, and the result of using the query ID is undefined:



After the last reply has arrived to the user's event handler function.

After the query has been aborted by calling uaa_abort_query.

After the event handler has been called because the query has timed out (that is, the timeout specified in uaa_send_query expired).

uaa_abort_query

Description: `uaa_abort_query` cancels a request to the UAA server and the event handler for the `UAA_QUERY_REPLY` is called. The function usage is defined in the `uaa_client.h` file.

Usage: `int uaa_abort_query ( OpsecSession *session, int query_id);`

Arguments

Table 8-18 uaa_abort_query Arguments

Arguments	Meaning
`session`	A pointer to the `session`.
`query_id`	The ID of the query to be cancelled, as returned by `uaa_send_query`.

Return Values: 0 if successful, or less than 0.

Managing Updates

uaa_send_update

Description: `uaa_send_update` sends an update to the UAA server. The function usage is defined in the `uaa_client.h` file.

Usage: `int uaa_send_update ( OpsecSession *session, uaa_assert_t *update, void *opaque, unsigned int timeout);`

Arguments

Table 8-19 uaa_send_update Arguments

Arguments	Meaning
`session`	A pointer to the OPSEC session.
update	A pointer to the `uaa_asser_t` structure containing the UAA update.
opaque	A general purpose pointer to be passed directly to the reply handler.
`timeout`	The number of milliseconds before a UAA request times out. If a reply is not available by this time, the event handler for the event is called with the appropriate status.

Return Values:

- Successful: A unique query ID different than (-1)
- Not Successful (-1)



Note - The update ID is not valid in any of the following cases, and the result of using the update ID is undefined:

After the last reply has arrived to the user's event handler function.

After the event handler has been called because the update has timed out (that is, the timeout specified in uaa_send_update expired).

Managing Authentication Requests

uaa_send_authenticate_request

Description: `uaa_send_authenticate_request` sends an authentication request to the UAA server. The function usage is defined in the `uaa_client.h` file.

Usage: `int uaa_send_authenticate_request ( OpsecSession *session, uaa_assert_t *auth_info, void *opaque, unsigned int timeout);`

Arguments

Table 8-20 uaa_send_authenticate_request Arguments

Arguments	Meaning
`session`	A pointer to the OPSEC session.
auth_info	A pointer to the `uaa_asser_t` structure containing the UAA authenticate information.
opaque	A general purpose pointer to be passed directly to the reply handler (see $$$).
`timeout`	The number of milliseconds before a UAA request times out. If a reply is not available by this time, the event handler for the event is called with the appropriate status (see the $$$).

Return Values:

- Successful: A unique query ID different than (-1)

- Not Successful (-1)



Note - The update ID is not valid in any of the following cases, and the result of using the update ID is undefined:

After the last reply has arrived to the user's event handler function.

After the event handler has been called because the authentication has timed out (that is, the timeout specified in uaa_send_authenticate_reqest expired).

Assertions Iteration

Function prototypes are defined in the `uaa.h` file. The following functions step through the assertions in a UAA assertions structure:

- "uaa_assert_t_iter_create" on page 128
- "uaa_assert_t_iter_get_next" on page 129
- "uaa_assert_t_iter_reset" on page 129
- "uaa_assert_t_iter_destroy" on page 130

uaa_assert_t_iter_create

Description: `uaa_assert_t_iter_create` creates an iteration object for UAA assertions.

Usage: `uaa_assert_t_iter * uaa_assert_t_iter_create(uaa_assert_t *asserts, char *type);`

Arguments

Table 8-21 uaa_assert_t_iter_create Arguments

Arguments	Meaning
asserts	A pointer to the `uaa_assert_t` structure containing the UAA assertions.
type	If non-NULL, the iterator is typed. That is, the iterator only iterates through assertions of the specified type. Type can be one of the following: • **NULL**: Iterate through all assertions in the assertions structure. • **Any other valid string**: Iterate through assertions of the specified type (for more information, see "Key Assertions" on page 113 and "Replies" on page 117).

Return Values: Pointer to assertions iterator, if successful, or `NULL`.

uaa_assert_t_iter_get_next

Description: `uaa_assert_t_iter_get_next` sets the iterator to the next assertion in the assertions structure.

Usage: `uaa_assert_t_iter_get_next (uaa_assert_t *iter, char **value char **type);`

Arguments

Table 8-22 uaa_assert_t_iter_get_next Arguments

Arguments	Meaning
iter	A pointer to the `assertion iterator`.
value	A pointer to be set to the value of the assertion.
type	A pointer to be set to the type of the assertion.

Return Values:

- If successful: 0
- If either of the following are true then the value is (-1):
  - There are no more request assertions of the specified type (in the case of a typed iterator (see "uaa_assert_t_iter_create" on page 128).
  - An error has occurred.

uaa_assert_t_iter_reset

Description: `uaa_assert_t_iter_reset` resets the iterator to the first assertion in the assertions data structure.

Usage: `uaa_assert_t_iter_reset (uaa_assert_t *iter);`

Arguments

Table 8-23 uaa_assert_t_iter_reset Arguments

Arguments	Meaning
`iter`	A pointer to the assertions iterator.

Return Values: 0, if successful, or a non-zero value.

uaa_assert_t_iter_destroy

Description: `uaa_assert_t_iter_destroy` destroys the assertions iterator and frees its memory.

Usage: void `uaa_assert_t_iter_destroy (uaa_assert_t *iter);`

Arguments

Table 8-24 uaa_assert_t_iter_destroy Arguments

Arguments	Meaning
`iter`	A pointer to the assertions iterator.

Return Values: None.

Managing UAA Errors

This section describes the error utility functions. The function usage is defined in the `uaa_error.h` file.

uaa_error_str

Description: `uaa_error_str` converts the status of a reply to an error message.

Usage: `char *uaa_error_str(uaa_reply_status status);`

Arguments

Table 8-25 uaa_error_str Arguments

Arguments	Meaning
`status`	The reply status, as returned by `status` argument of the `reply` event handler.

Return Values: A string indicating the error, if successful, or `NULL`.

Debugging

This section describes utility functions for debugging. To enable these functions, the `OPSEC_DEBUG_LEVEL` environment variable must be set to 3. For further details about the `OPSEC_DEBUG_LEVEL`, see *OPSEC API Specification*.

Function prototypes are defined in the `uaa.h` file.

uaa_print_assert_t

Description: `uaa_print_assert_t` prints the contents of the `uaa_print_assert_t` structure.

Usage: `void uaa_print_assert_t(uaa_assert_t *asserts);`

Arguments

Table 8-26 uaa_print_str Arguments

Arguments	Meaning
`asserts`	A pointer to the uaa_assert_t structure to be printed.

Return Values: None

Event Handlers

This section describes the functions that need to be written to implement a UAA Client. All of these functions take a pointer to `OpsecSession` as an argument.



Note - Memory allocated for function arguments is managed by the OPSEC environment, and the arguments hold valid data only during the execution of the handler functions. For this reason, you should not, for example, save a static pointer to this data for use after the handler function returns.

UAA_QUERY_REPLY Event Handler

Description: `UAA_QUERY_REPLY` is called when a reply to a UAA query becomes available.



Note - The name QueryReplyHandler is a placeholder. You can assign any name to this function.

Usage: `int QueryReplyHandler( OpsecSession *session, uaa_assert_t *reply, void *opaque, int query_id, uaa_reply_status status, UaaReplyIsLast last);`

Arguments

Table 8-27 QueryReplyHandler Arguments

Arguments	Meaning
session	A pointer to an `OpsecSession` structure, as returned by `uaa_new_session` (see"uaa_new_session" on page 121).
reply	A pointer to the `uaa_asser_t` structure containing the reply assertions.
opaque	The general-purpose pointer copied from the corresponding call to `uaa_send_query` (see "uaa_send_query" on page 125).

Table 8-27 QueryReplyHandler Arguments

Arguments	Meaning
query_id	The ID returned by the corresponding call to `uaa_send_query` (see "uaa_send_query" on page 125).
status	The reply status: • `UAA_REPLY_STAT_OK` if no errors have occured • Otherwise, a value that can be converted to an error message using `uaa_error_str` (see "uaa_error_str" on page 130).
last	The value `UAA_REPLY_LAST` indicates that this is the last reply for the specific query and the value `UAA_REPLY_NOT_LAST` indicates that the server will send additional replies.

Return Values:

- `OPSEC_SESSION_OK` if the session can continue.
- `OPSEC_SESSION_END` if the session must be closed.
- `OPSEC_SESSION_ERR` if the session must be closed due to an error.

UAA_UPDATE_REPLY Event Handler

Description: `UAA_UPDATE_REPLY` is called when a reply to a UAA update becomes available.



Note - The name UpdateReplyHandler is a placeholder. You can assign any name to this function.

Usage:
```
int UpdateReplyHandler( OpsecSession *session, uaa_assert_t *reply,
void *opaque, int cmd_id, uaa_reply_status status;
```

Arguments

Table 8-28 UpdateReplyHandler Arguments

Arguments	Meaning
session	A pointer to an `OpsecSession` structure, as returned by `uaa_new_session` (see"uaa_new_session" on page 121).
reply	A pointer to the `uaa_asser_t` structure containing the reply assertions.
opaque	The general-purpose pointer copied from the corresponding call to `uaa_send_update` (see "uaa_send_update" on page 126).
cmd_id	The ID returned by the corresponding call to `uaa_send_update` (see "uaa_send_update" on page 126).
status	The reply status: • `UAA_REPLY_STAT_OK` if no errors have occured • Otherwise, a value that can be converted to an error message using `uaa_error_str` (see "uaa_error_str" on page 130).

Return Values:

- `OPSEC_SESSION_OK` if the session can continue.
- `OPSEC_SESSION_END` if the session must be closed.
- `OPSEC_SESSION_ERR` if the session must be closed due to an error.

UAA_AUTHENTICATE_REPLY Event Handler

Description: `UAA_AUTHENTICATE_REPLY` is called when a reply to a UAA authentication request becomes available.



Note - The name AuthenticationReplyHandler is a placeholder. You can assign any name to this function.

Usage: `int AuthenticationReplyHandler( OpsecSession *session, uaa_assert_t *reply, void *opaque, int cmd_id, uaa_reply_status status;`

Arguments

Table 8-29 UpdateReplyHandler Arguments

Arguments	Meaning
session	A pointer to an `OpsecSession` structure, as returned by `uaa_new_session` (see"uaa_new_session" on page 121).
reply	A pointer to the `uaa_asser_t` structure containing the reply assertions.
opaque	The general-purpose pointer copied from the corresponding call to `uaa_send_authenticate_request` (see "uaa_send_authenticate_request" on page 127).
cmd_id	The ID returned by the corresponding call to `uaa_send_autheticate_request` (see "uaa_send_authenticate_request" on page 127).
status	The reply status: • `UAA_REPLY_STAT_OK` if no errors have occured. • Otherwise, a value that can be converted to an error message using `uaa_error_str` (see "uaa_error_str" on page 130).

Return Values:

- `OPSEC_SESSION_OK` if the session can continue.
- `OPSEC_SESSION_END` if the session must be closed.
- `OPSEC_SESSION_ERR` if the session must be closed due to an error.

Chapter 9

Monitoring the UserAuthority Environment

In This Chapter

Overview page 138
System Monitoring page 139
User Monitoring page 144

Overview

Monitoring allows the system administrator to view the system status for debugging and problem solving in the system. For example, an administrator might receive a complaint that a user is unable to access a Web application. The administrator can use the monitoring tools to determine if this is due to a problem in the system (such as a server is offline) or a problem in the system configuration, or because the user does not have the necessary authorization to access the requested application.

There are two types of monitoring in UserAuthority:

- **System monitoring** is used to check the status and state of the UserAuthority System at any time. The system is monitored to determine if any component is offline or if there are problems in the system's configuration. See "System Monitoring" on page 139.
- **User monitoring** is used to determine if there are any problems specific to the user. Logs are used to follow the user's requests and see how the system responds (e.g., what queries are made by the UserAuthority Server (UAS)). See "User Monitoring" on page 144.

This chapter describes the two types of monitoring and how to carry out monitoring activities.

System Monitoring

In This Section

Monitoring the System Status page 139

Monitoring the System Status

UAS protects the local network and specific Web applications from access by unauthorized and unauthenticated users. For this reason, it is important to know whether all components in the system are operating. Check Point's SmartCenter server gathers information on all system components. This information can be monitored using the SmartView Monitor console. For more information on how to use SmartView Monitor, see the *Check Point SmartCenter Guide*.

SmartView Monitor reports system status information for all Check Point and OPSEC modules in the system, including UAS. The possible module status types are:

- **OK**: The module installed on the object is responding to status update requests indicating that everything is working correctly.
- **Untrusted**: Secure Internal Communication failed.
- **Problem**: The module installed on the object is responding to status update requests, but there is a problem in the status. There can be different types of problems, such as the UAS is not responding.
- **Attention**: The module is active although there might be a problem on a product installed on the module.

For more information on monitoring statuses, see the *Check Point SmartCenter Guide*.

You can display specific information about a module by:

- Clicking the icon in the toolbar, which displays a window containing information for all modules for the selected product (UAS).
- Clicking on the module in the Modules pane, which displays detailed information about the UAS installed on that module.

The following sections describe the detailed information displayed for the UAS .

UAS

SmartView Monitor lists the modules that are deployed in your network. Each product that is installed on a module is listed in the tree under the module. When you select a UAS in the module tree, details for the selected UAS are displayed in the Details pane on the right side of the window.

Table 9-1 UAS Details

Detail	Description
Status	The status for the selected UAS. See "Monitoring the System Status" on page 139 for a list of possible statuses.
Description	A description of the UAS on that module.
UserAuthority Server Version	The software version for the selected UAS.
Policy Name	The name of the policy installed on the selected UAS.
Installed At	The date and time that the last UserAuthority policy was installed.
License	The license number and information for the selected UAS.
UserAuthority Server Type	The type of UAS (installed on VPN-1 Pro, on a Windows Domain Controller (DC), or on a Citrix/Terminal Services).
Configuration	A list of items included in the configuration that relate to the selected UAS: • Log Server IP Addresses. • Windows domains trusted by VPN-1 Pro. • Other UASes that provide identity information.
Run-Time Information	A list of run-time items: • The IP addresses for UserAuthority OPSEC clients. • Number of requests processed. • Average response time per request (in seconds).

Using UAS Logs for System Monitoring

UAS has three types of logs. The log type is displayed in the **Type** column of the Records pane or the **Record Details** window. The log types are:

- **Log**: Standard logs that describe what is happening or whether a query is carried out for each user request. For example, `Authentication Success` is a log entry that indicates that the user was authenticated, and appears in a regular log file.
- **Alert**: Alerts are displayed in red and call attention to potential problems in the system. An example of an alert is `Web server is stopping`. This indicates that the Web server is not online.
- **Control**: Control logs indicate a standard system activity. For example, when the system is turned on or configured there must be a connection between different components.

The Alert and Control logs are helpful for system monitoring. They can show potential problems or indicate whether standard communication activities have occurred, and can be used to troubleshoot system problems.

The actual messages displayed in the logs can be edited to fit the needs of your organization.

Using UAS Logs

UAS logs provide information on queries to and from the UASes in the deployment, as well as information on the chaining (shared identity) between computers. To use UserAuthority logs, verify that there is a Log Server and then configure the logging level (see "Configuring the Logging Level for the UAS on the FireWall Gateway" on page 141). For information on log servers, see the *Check Point SmartCenter Guide*.

UAS logs are useful for solving user access problems.

Configuring the Logging Level for the UAS on the FireWall Gateway

UAS logs can be configured to work on three levels:

Low: UAS generates Alert and Control logs only.

Medium: UAS generates logs on UAS query failures, in addition to the Alert and Control logs.

High: UAS generates logs with detailed information about UAS queries, including failures in identity sharing, in addition to the logs generated on the Low and Medium levels.

To configure the level of UAS logs, do the following in SmartDashboard:

1. In the Network Object tree, double click the network object for the VPN-1 Pro gateway with UAS installed.

   The **Check Point Gateway** window is displayed.

2. In the tree pane, select **UserAuthority Server**.



Note - You must separately configure the logging level for each UAS on a VPN-1 Pro gateway.

UASes on Windows DCs are configured to create logs by default. To change the logging configuration for UASes on Windows DCs, you must edit the `netso.ini` file on the Windows DC. For information, see "Configuring Logs for UASes not on a FireWall Gateway" on page 142.

3. In the **Logging Level** area, select a logging level from the **UserAuthority Logging Level** drop-down list.
4. Click **OK** to close the window.
5. Save and install the policy on the VPN-1 Pro gateway.

Configuring Logs for UASes not on a FireWall Gateway

By default, UASes on Windows DCs and Citrix/Terminal services are configured to generate logs. The log generation configuration is found in the `netso.ini` file on the Windows DC or Citrix/Terminal Services machine. If logs are not being created or you want to turn off logging for the UAS on the Windows DC, you must edit the `netso.ini` file.

To configure UAS Logging on a Windows DC:

1. In the Windows DC or Citrix/Terminal Services machine, browse to the UAS installation directory (by default `C:\\Program Files\Check Point\UAG\R55\Conf`).

2. From the `Conf` folder, open the `netso.ini` file.



Note - You must open the `netso.ini` file with WordPad. You cannot open it with NotePad.

3. In the `[NETSO_Configuration]` section, find the line `log server=`
4. After the equal (=) sign, enter the IP address or net bios name for the machine with the log server (if you want the logs to be created on the management server, enter `DN_Mgmt`). In the event of multiple log servers, enter the IP addresses (or net bios) for each one separated by commas (,).
5. Save and close the file.
6. Run `UAS renconf` to restart the UserAuthority Service and activate the changes to the file.

   The following is an example of the `netso.ini` file configured to create logs on the management computer.

```
Log Server = DN_Mgmt
```

For more information on user monitoring, see "User Monitoring" on page 144.

User Monitoring

In This Section

Monitoring User Activities page 144
Monitoring Example: SecureAgent Cannot Provide User Identity page 145

Monitoring User Activities

UAS logs enable user monitoring to troubleshoot user problems in the system. These logs provide a description of the activities that occur in the system when a user makes a request. For example, UAS logs indicate when a query for the user's identity is made.

If you want to compare the UAS activities for the same user request, you should create a filter to display only logs with the same UserAuthority Session ID (UA Session ID). This ID will be the same for both types of logs.

By examining these logs you can monitor many user activities, such as:

- User requests to Web applications
- User authentication
- User credential injection
- Replies to user requests

Each of the processes or queries in the flow is represented by a UserAuthority log. The logs indicates where the initial request came from, where it is going and what is happening. In some cases the result is also indicated.

This information can be used to determine why a user might be unable to access applications or benefit from SSO. Configuration problems can then be corrected so that the user can continue to use the Web applications on the network as usual. See "Monitoring Example: SecureAgent Cannot Provide User Identity" on page 145 for an example on monitoring user activities.

Monitoring Example: SecureAgent Cannot Provide User Identity

You can use UAS logs to determine why user identity is not achieved. The following is true in this example:

- The user must be identified by the Windows DC.
- SecureAgent is not active.

In this case, when a user attempts to access a Web application, the user receives a message that the service is not available. This is because the UAS is unable to identify or authenticate the user. When this problem is reported in the system, it is easy to determine why this happens with the logs. The following UAS logs are generated when a user requests a Web application in this situation:

Figure 9-1 Unsuccessful Attempt to Access a Web Application

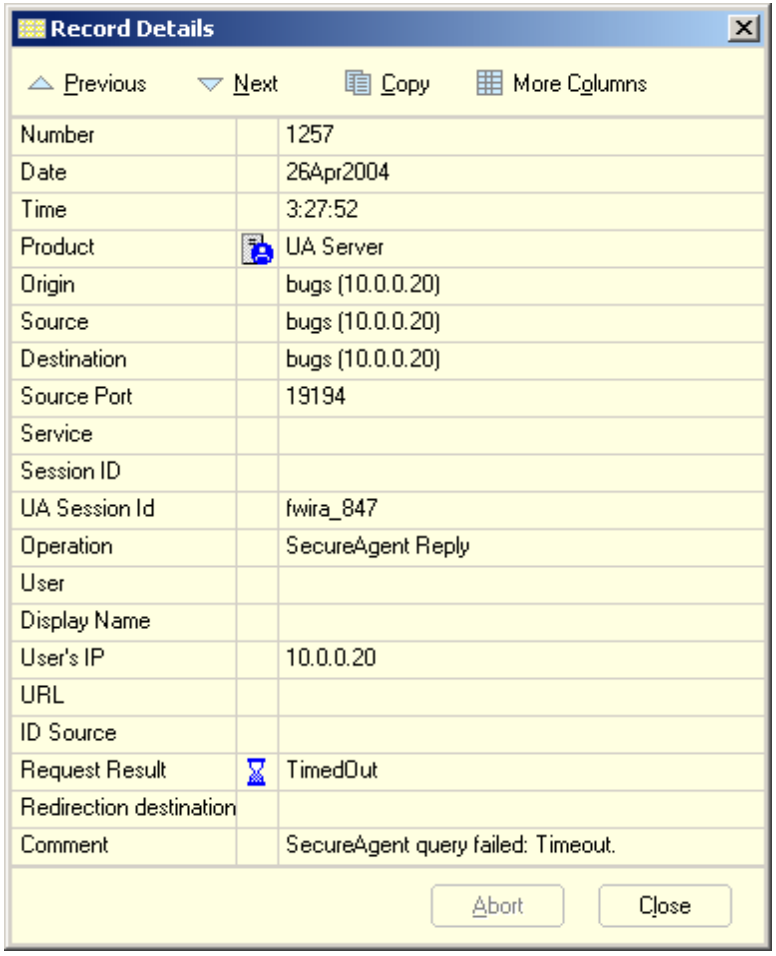
UA Server	Win_DC	FireWall..	Win_DC	FireWall_2	UAA Query	194.29.37.115	Success	New UAA query request.
UA Server	Win_DC	FireWall..	Win_DC	FireWall_2	UAA Query	194.29.37.115	Success	New UAA query request.
UA Server	Win_DC	Win_DC	FireWall..	FireWall_2	SecureAgent Query	194.29.37.115	Success	SecureAgent query.
UA Server	Win_DC	Win_DC	FireWall..	FireWall_2	SecureAgent Query	194.29.37.115	Success	SecureAgent query.
UA Server	Win_DC	FireWall..	Win_DC	FireWall_2	SecureAgent Reply	194.29.37.115	TimedOut	SecureAgent query failed: Timeout.
UA Server	Win_DC	FireWall..	Win_DC	FireWall_2	SecureAgent Reply	194.29.37.115	TimedOut	SecureAgent query failed: Timeout.
UA Server	Win_DC	Win_DC	FireWall..	FireWall_2	UAA Query Reply		Success	UAA query empty reply.
UA Server	Win_DC	Win_DC	FireWall..	FireWall_2	UAA Query Reply		Success	UAA query empty reply.

The logs in this example indicate the following:

1. The UAS on the VPN-1 Pro gateway queries the UAS on the Windows DC. The log information indicates the two machines and that the query was successful.
2. The UAS on the Windows DC queries SecureAgent for the user's identity. This happens because the system is not configure for Windows Integrate Authentication. In this case, it is necessary to install the UAS on the Windows DC and retrieve the user identity with SecureAgent.
3. An alert indicating that the system is not active is returned because SecureAgent is not responding.

   The following **Record Details** window shows the information returned for this alert.

Figure 9-2 SecureAgent Query Timed Out



The comment clearly states that the SecureAgent query failed and the system timed out.

4. The last log shows that the UAS on the Windows DC sent an empty query back to the UAS on the VPN-1 Pro gateway. An empty query indicates that there is no identification information for the user requesting the Web application. Therefore, the VPN-1 Pro cannot forward the request and the user receives a message indicating that the service is not available.

Chapter 10
Troubleshooting UserAuthority

In This Chapter

Overview page 148
General Problems page 149
User-Related Problems page 153

Overview

This chapter provides help for common problems that might arise when using UserAuthority. Problems in UserAuthority can be divided into two categories:

- **General Problems:** These are problems that effect the system as a whole, such as a system failure or bad configuration.
- **User problems:** These are problems that effect a single user, such as improper configuration of the user's SecureAgent.

In addition to the information provided in this chapter, you can also read the logs generated to identify a problem. For more information on using logs to monitor system errors, see Chapter 9, "Monitoring the UserAuthority Environment"".

General Problems

This section provides information on common problems in the overall system.

In This Section

Why is there no established SIC? page 149
Why are Domain Controller Queries not Sent Properly? page 152

Why is there no established SIC?

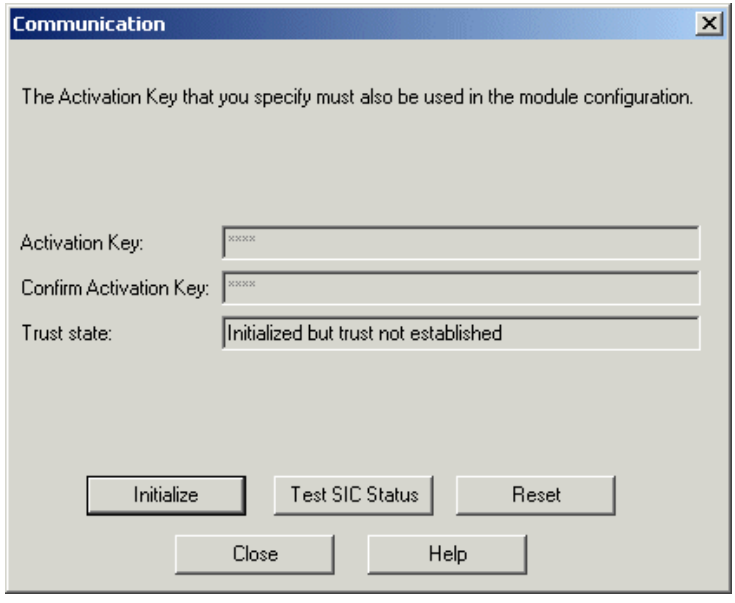
Symptom

There is a problem in the Secure Internal Communication (SIC) configuration in the UAS.

Problem

When completing the SIC configuration in SmartDashboard, you receive a SIC Failure message in the **Communication** window.

Figure 10-1 SIC Failure Message



Solutions

Verify the SIC status.

To verify SIC status:

1. From SmartDashboard, double click the relevant network object. The **Network Host** window is displayed.
2. In the **Secure Internal Communication** area at the bottom of the window, click **Communication**.

   The **Communication** window is displayed.
3. Click **Test SIC Status**.

   Make sure that the **Trust state** is **Communicating**. If the **Trust state** is not **Communicating**, then SIC is not established.

   If SIC is not established, do one or more of the following as necessary:

   - Make sure that the Check Point SVN Foundation service is started on the relevant network object.
   - Make sure that the relevant network object can be reached from the Check Point SmartCenter management server and that communication is not blocked by a VPN-1 Pro module.

     Note that VPN-1 Pro inserts an implied rule for this communication.
   - Make sure that there is time and time zone synchronization between the VPN-1 Pro gateway and the relevant network object.
   - Re-establish SIC (see "Re-establish SIC" on page 150).

Re-establish SIC

You must re-establish SIC on the VPN-1 Pro gateway where the UAS is installed and in SmartDashboard.

To re-establish SIC on the relevant machine:

On a Windows machine:

1. From the **Start** menu, select **Programs > Check Point SmartConsole NGX_R61 > Check Point Configuration** to open the **Check Point Configuration** window.
2. Click the **Secure Internal Communications** tab.
3. Click **Reset** and then confirm by clicking **Yes**.

4. Enter a password key in the **Activation Key** field and then enter it again in the **Confirm Activation Key** field to confirm it. Be sure to remember your key, you need to enter it in the SmartDashboard configuration.
5. Click **OK**.
6. Click **Yes** to restart Check Point services.

If this is a Linux or Unix machine:

1. From a command line, type `sysconfig`.
2. From the `Configuration` menu, type `7`; `Products Configuration` and then press Enter.
3. From the `Products Configuration` menu, type `3`; `Secure Internal Communication` and then press Enter.
4. At the prompt, `Would you like to re-initialize communication?`, type `y` and then press Enter.
5. Type your password as described in the Windows procedure and follow the on-screen instructions to close and save your configuration.

To re-establish SIC in SmartDashboard:

1. Double click the relevant network object. The **Network Host** window is displayed.
2. In the **Secure Internal Communication** area at the bottom of the window, click **Communication**.
3. Click **Reset** and then click **Yes**.
4. Click **OK** in the **Reset is done** window.
5. In the **Activation Key** field, enter the activation key that you created when you re-initialized SIC on the relevant machine.
6. Enter the activation key again in the **Confirmation** field.
7. Click **Initialize**.

   If the Operation is successful, the words **Trust established** are displayed in the **Trust state** field.

Why are Domain Controller Queries not Sent Properly?

Symptom

A Domain Controller end-user with SecureAgent is not authenticated by the firewall. The SecureAgent rejects the authentication requests from the Domain Controller.

Problem

The Domain Controller has 2 enabled network interfaces on the same subnet, where one is connected to the LAN and the other is disconnected (but not disabled).

The Domain Controller attempts to send the authentication query to the SecureAgent with the disconnected interface's IP.

As a result, the SecureAgent rejects the authentication request, and the user is not authenticated by the firewall.

Solutions

Disable the interface that is disconnected. Disabling the disconnected interface on the Domain Controller and restarting the clients SecureAgent process resolves this problem.

User-Related Problems

This section provides information on common problems related to individual users.

In This Section

Why does SecureAgent not identify the user? page 153
Why are Terminal Server Clients not Identified by UAS? page 156
Why does the Firewall Report Identify Users as Unknown? page 157

Why does SecureAgent not identify the user?

Symptom

A user with SecureAgent is not identified by VPN-1 Pro. In this case, the user is denied access to all external resources (resources on the other side of the gateway).

Problem

SecureAgent is not retrieving the user's identity.

Solutions

- Make sure that SecureAgent is running by doing the following:
  - Check that the SecureAgent icon is in the taskbar and that it is still active. The SecureAgent icon looks like .
  - From the Windows Task Manager, click the **Processes** tab and make sure that the uatc.exe process is running.
- Make sure that SecureAgent is installed on the user's PC.
  - Make sure that the user is logged on to the Windows Domain Controller (DC) and not to a local machine account.
  - Make sure that the user is not using cached credentials (this occurs when the machine cannot connect to the Windows DC when logging on).
  - Make sure that "Configure SecureAgent automatic installation through a Windows Logon Script" was configured. See "Configuring SecureAgent Automatic Installation" on page 50.
  - Make sure that the SecureAgent scripts are in the NETLOGON directory (see Table 2-1 on page 51).

  - Make sure that the client machine has the `MSVCP60.dll` (this DLL is available from Microsoft).
  - Make sure that the user has sufficient rights to install programs on the PC (i.e., the user is an administrator on the target machine).
- Make sure that SecureAgent is communicating with the UAS:
  - Make sure that there is network connectivity between the Windows DC and the desktop.
  - Check the UAS logs to make sure that the UAS on the Windows DC is sending queries to the SecureAgent.
  - Make sure that Client IPs are not hidden from the VPN-1 Pro gateway by an intermediate VPN-1 Pro NAT Hide rule.
  - Make sure that SecureClient/SecuRemote or a Personal Firewall are not blocking the query (UDP port 19190).
  - For clients running both SecureClient and SecureAgent, the Desktop Policy must contain the following rule:

Table 10-1 Desktop Policy

Source	Desktop	Service	Action
Windows DC(s)	LAN computer	CP_SecureAgent-udp	Accept

  - Make sure that the **Client for Microsoft Networks** checkbox is selected in the **Windows Network Settings** window.
- If SecureAgent flashes red when trying to access an external resource, then make sure the server that is attempting to query the SecureAgent is defined in the `acl.txt` file.

  To define the server:

  1. On the Windows DC where the UAS is installed, open the file `uatcs-acl.txt` in Windows WordPad.

2. Edit the following file parameters:

   [hostname]: The host name of the UAS.

   [ipaddress]: The IP address of the UAS.

   [port]: The UAS UDP source port (this should always be 19195).

   The following is an example of a uatcs-acl.txt file configured to accept queries from a Windows DC with the name DC, IP address 10.0.0.2, and port number 19195.

```
#
#hostname              ipaddress              port
#
DC                     10.0.0.2               19195
```



Note - Normally you would modify this file on the Windows DC and have it distributed to clients automatically. If this file is modified directly on a client machine, then SecureAgent must be restarted.

- Make sure that the SecureAgent installation is completed before browsing for external resources. This is verified when the **Command Prompt** window that is running the script appears and then closes.
- Make sure that there are no HTTP (cache) proxies between Web browsers and the VPN-1 Pro gateway. If you are using an HTTP proxy, then you must do one of the following:
  - Use a special configuration file to make requests from specific DNS entries bypass the HTTP proxy.

    To do this:

1. From **Internet Explorer**, select **Tools > Internet Options**. The **Internet Options** window is displayed.
2. Click the **Connections** tab.
3. Click **LAN Settings**. The **Local Area Network (LAN) Settings** window is displayed.
4. In the **Automatic Configuration** area, select **User automatic configuration script**.
5. In the **Address** field, enter the FQDN or IP address where your configuration file is located.
6. Click **OK**, and then **OK** again to close the windows.

7. Make sure the configuration file contains the DNS entries that you want to bypass the HTTP proxy. The following is an example of a configuration file:

```
function FindProxyForURL(url, host
{
if (isPlainHostName(host) ||
dnsDomainIs(host, ".checkpoint.com") ||
dnsDomainIs(host, ".checkpoint.co.jp") ||
isInNet(host, "172.31.0.0", "255.255.0.0") ||
isInNet(host, "192.168.0.0", "255.255.0.0") ||
isInNet(host, "10.0.0.0", "255.0.0.0") ||
isInNet(host, "127.0.0.1", "255.255.255.255") ||
dnsDomainIs(host, ".us.checkpoint.com") ||
dnsDomainIs(host, ".ts.checkpoint.com"))
return "DIRECT";
else
return "PROXY proxy-scan1.checkpoint.com:8080; PROXY
proxy5.checkpoint.com:8080; DIRECT"; }
```

Why are Terminal Server Clients not Identified by UAS?

Symptom

A Terminal Server Client is not identified by VPN-1 Pro. In this case, the Ternminal Server Client is denied access to all external resources (resources on the other side of the gateway).

Problem

UAS is not retrieving the user's identity.

Solutions

In order to identify users coming from a Terminal Server Client, each session must be authenticated. To authenicate a session use a Session Authentication rule with SSO.

Why does the Firewall Report Identify Users as Unknown?

Symptom

Although a user is identified by the UAS and is reported to the firewall, the firewall logs show the user as **unknown**. This causes the user to be dropped by the firewall rule.

Problem

UserAuthority identifies the user and presents the data to the firewall. The firewall then checks its user databases to verify the existence of the user in question. If the user does not exist in any internal or external database (for example, LDAP) the user will be blocked as an **unknown** user.

Solutions

To assure that the user is identified by the firewall, the administrator must provide a user database that can be accessed by SmartCenter Management (for example, external database or the local firewall user database). Only users defined in such databases will be identified by the firewall.

Appendix A
Integrating UserAuthority with Meta IP

In This Appendix

Overview page 160
Required Components page 161
Preliminary Steps page 162
Windows DC Configuration page 163
VPN-1 Pro Policy Configuration page 164
DHCP Server Configuration page 166

Overview

Meta IP has a DHCP plugin that monitors a DHCP Server IP subscription. UserAuthority can easily be integrated with the Meta IP product to provide authenticated IP addresses from an authenticated IP pool to authenticated users.

Required Components

- Check Point NG with Application Intelligence, UserAuthority Server.
- A Microsoft Windows NT or Windows 2000 Server Domain Controller (DC).
- A DHCP relay installed on the router.
- Check Point SmartDashboard installed on a management server.
- Meta IP Feature Pack 2 Hotfix 1 (Professional or Enterprise edition).
- Meta IP UAA Programmable Extension component.

Preliminary Steps

- Install Check Point VPN-1 Pro, UserAuthority Server, and Check Point SMART Clients (see “Installing and Configuring UAS on VPN-1 Pro” on page 31).
- Install Check Point UserAuthority Server on the Windows DC (see “Installing and Configuring the UAS on the Windows DC” on page 43).

Windows DC Configuration

1. After installation, verify that `uatcs.bat` script and its associated files have been installed in the `netlogon` shared folder on the Windows DC. These files should reside in the same folder that is used to store user logon scripts for the Windows domain. For example, on a Windows 2000 Server, the path to this folder is:

   `<windows_dir>\SYSVOL\<win_domain_name>\scripts`

   Note - The following files should reside in `netlogon` shared folder:

   - `instuatc.exe`
   - `uatc.exe`
   - `uatcs.bat`
   - `uatcs-acl.txt`

2. Edit the uatcs-acl.txt file to include an entry for your Windows DC. If your Windows DC has multiple interfaces, add an entry for each IP address associated with the Windows DC. For example, to add a Windows DC called DOMAINCONTROL with IP addresses 172.16.10.21 and 10.11.1.1, add the following entries to the uatcs-acl.txt file:

   - `DOMAINCONTROL 172.16.10.21 19195`
   - `DOMAINCONTROL 10.11.1.1 19195`

3. Using Active Directory Users and Computers (Windows 2000 Server) or User Manager For Domains (NT 4.0 Server), configure each user's profile to run the uatcs.bat logon script.

VPN-1 Pro Policy Configuration

See the *Check Point Firewall and SmartDefense Guide* for details on how to perform the following procedures in SmartDashboard:

1. Add the Windows DC:
   a. From SmartDashboard, click the **Network Objects** tab and create an object for your Domain Controller.
   b. Right click Check Points and select **New > Check Point > Host**. The **Check Point Host** window is displayed.
   c. Enter the name of the Windows DC.
   d. Enter the IP address of the Windows DC.
   e. Under Check Point Products, select the **UserAuthority Server** checkbox and click **Communication**. If trust has not been established, provide an activation key and click **Initialize** to establish trust between the Windows DC and VPN-1 Pro. After initialization, click **Close**.
   f. Click **OK** to close the window.
2. Create an entry for each Meta IP DHCP server under nodes in the Check Point SmartDashboard. Right-click **Nodes**, and select **New > Host**.
   a. Enter a name for the host.
   b. Enter an IP address.
   c. Click **OK** to close the window.
3. Open the **OPSEC Applications** tab, right click **OPSEC Application**, and select **New OPSEC Application.** For more information on configuring OPSEC applications, see Chapter 11, "UserAuthority OPSEC APIs".
   a. Enter a name for the OPSEC application (for example, `dhcp_uaa`).
   b. Select a DHCP server object as the host for the OPSEC application.
   c. In the **Client Entities** area, select the **UAA** checkbox. Click **Communication** and specify an activation key, then click **Initialize**. After initialization, click **Close**. Note that trust will not be established between the DHCP server and the OPSEC Application object on the firewall until the DHCP server has pulled the certificate from the Certificate Authority.

4. For networks that use UAA communications, configure a rule on the VPN-1 Pro to allow communications over the following ports:
   - UDP Communications between the VPN-1 Pro and the Windows DC on ports 19194 and 19195 (you may choose pre-defined service: CP_SecureAgent).
   - TCP Communications between the DHCP Server(s) and VPN-1 Pro on ports 19191 and 18210 (you may choose pre-defined services: FW_uaa and FW_ica_pull).
5. Install the policy.

DHCP Server Configuration

1. Run the `opsec_pull_cert` utility, which establishes trust with the Certificate Authority (VPN-1 Pro). The required command line parameters are:
   - -h <VPN-1 Pro IP address>
   - -n <OPSEC application object name>
   - -p <activation key>

     Example:

   - `opsec_pull_cert.exe -h 172.16.10.20 -n UAAPE_NT -p` *activation_password*

     The –n parameter must match the name of the OPSEC application object created in the Firewall Policy Configuration procedure.

     The –p parameter must match the activation key specified in the OPSEC Communications Properties dialog.



Note - On Windows NT 4.0 Servers, it may be necessary to provide the FQDN of the VPN-Pro instead of the IP address of the -h parameter.

2. Install the DHCP UAA programmable extension on the DHCP Server and on the computer hosting the Meta IP NG FP2 Management Console by running the installation package.
   - On Windows Platforms, run the `mip_uape51.msi` file to install the programmable extension and UI Updates.
   - On Solaris platforms, do the following:

   a. Copy `miusrauth.tgz` to a directory on the DHCP Server.

   b. unzip `miusrauth.tgz`

   c. `tar -xvf miusrauth.tar`

   d. `ipkgadd -d`

   e. Choose to install the `miusrauth` package.

f. Answer `yes` to the following prompt:

```
The following files are already installed on the system and are
being used by another package:
/opt/metaip51/bin/dhcsim
/opt/metaip51/sbin/dhcpd
Do you want to install these conflicting files [y,n,?,q] y
```

g. Add the Check Point SVN Foundation library path to the metaip51 profiles: `metaip51_profile.sh` and `metaip51_profile.csh`.

h. After installation completes, restart the SMC service:

```
/etc/init.d/mip-smc-51 start
```

- On Linux (7.3 and later)

a. Copy the file `miusrauth-51-00.i386.rpm` to a directory on the DHCP server ii rpm -i `miusrauth-51-00.i386.rpm`

b. Add the Check Point SVN Foundation library path to the `metaip51` profiles: `metaip51_profile.sh` and `metaip51_profile.csh`.

c. After installation completes, restart the SMC service:
`/etc/init.d/mip-smc-51 start`



Note - If you have a secondary DHCP server, you must configure the secondary DHCP server to authenticate with the same UserAuthority server that the primary DHCP server uses.

3. On Unix platforms, modify the LD_LIBRARY_PATH in the Meta IP profiles to include the CPShared library directory. This enables DHCP to dynamically link to the OPSEC libraries.

a. Open /`opt/metaip51/etc/metaip51_profile.sh`. in a text editor.

- On Linux platforms, change:

```
LD_LIBRARY_PATH="${CPIPIDIR}/lib:${LD_LIBRARY_PATH}"
```

to

```
LD_LIBRARY_PATH="${CPIPIDIR}/lib:/opt/CPshrd-55-03/
lib:${LD_LIBRARY_PATH}"
```

- On Solaris platforms, change:

```
LD_LIBRARY_PATH="${CPIPIDIR}/lib:${LD_LIBRARY_PATH}"
```

to

```
LD_LIBRARY_PATH="${CPIPIDIR}/lib:/opt/CPshrd-55/
lib:${LD_LIBRARY_PATH}"
```

b. Open `/opt/metaip51/etc/metaip51_profile.csh`. in a text editor.

- On Linux platforms, change:

```
setenv LD_LIBRARY_PATH "${CPIPIDIR}/lib:${LD_LIBRARY_PATH}"
```

to

```
setenv LD_LIBRARY_PATH "${CPIPIDIR}/lib:/opt/CPshrd-55-03/
lib:${LD_LIBRARY_PATH}"
```

- ii On Solaris platforms, change:

```
setenv LD_LIBRARY_PATH "${CPIPIDIR}/lib:${LD_LIBRARY_PATH}"
```

to

```
setenv LD_LIBRARY_PATH "${CPIPIDIR}/lib:/opt/CPshrd-55/
lib:${LD_LIBRARY_PATH}"
```



Note - Check Point SVN Foundation R 55 with Application Intelligence uses the CPshrd-55 directory. If the directory name for CPshared changes, you must update the Meta IP profile files to reflect the new path.

c. Stop and restart the SMC service:

`/etc/init.d/mip-smc-51 stop`

`/etc/init.d/mip-smc-51 start`

4. To configure DHCP, enter the following information in the UserAuthority window:

a. **The complete Path to the UserAuthority Extension on the DHCP Server machine** (for example, `Program Files\MetaIP\5.1\lib\uaauth.dll or /opt/metap51/ lib/uaauth.so`)

b. **The IP Address of the UserAuthority Server**: This server is usually located on the same computer that is running the VPN-1 Pro. Specify the IP address of this server.

c. T**he Port that UserAuthority Server Listens on**: The default port is 19191. If UserAuthority is configured to listen on a different port, enter that port number instead.

d. **Timeout (in seconds) for UA Queries (1-300)**: The maximum time that the DHCP server should wait for a response from the UserAuthority server. Do not change this value unless you have a specific reason.

e. **Client SIC Name**: The Secure Internal Communication (SIC) name of the OPSEC Client returned by the Certificate Authority. To find this name, open the OPSEC Application Properties dialog for your OPSEC application object in the Check Point Policy Editor. The SIC name for the OPSEC client appears near the bottom of the dialog in the "DN:" edit box.

Example:
CN=UAAPE_NT,O=SAGITTARIUS.uagdomain.metainfo.com.sct29n

f. **Server SIC Name**: The SIC name of the Certificate Authority (VPN-1 Pro). To find this name, open the VPN-1 Pro server's **Properties** window in the Check Point Policy Editor. You can open this dialog by clicking on the **Network Objects** tab in the Policy Editor, selecting the Check Point object corresponding to your VPN-1 Pro, and selecting **Edit** from the pop-up menu for that object. The Secure Internal Communication (SIC) name for the Certificate Authority (FireWall-1) appears near the bottom of the dialog in the "DN:" edit box.

Example: `CN=cp_mgmt,O=SAGITTARIUS.uagdomain.metainfo.com.sct29n`

g. **Complete Path to the "p12" file on the DHCP Server machine**: Enter the path to the certificate file that was created when you ran the opsec_pull_cert utility (for example, `Program Files\MetaIP\5.1\etc\opsec.p12 or /opt/metap51/etc/opsec.p12`).

h. **Logging**: Set the desired logging level. Logging levels include (listed from most detailed to least detailed):

- **Debug**: Client authentication debug messages.
- **Info**: Client authentication information messages.arn — Warning messages on client authentication.
- **Error**: Error messages on client authentication.
- **Std. Error**: Logs messages to STDERROR.

5. Create a Shared network object containing at least two lease pools: one unauthenticated and one authenticated.

a. In the *authenticated* lease pool, set the following parameters and options:

- DHCP Parameter **Client Request Handling Authentication level** = Authenticated One Lease Per Client = True
- DHCP Parameter **Lease Time Default Lease Time** = desired lease time for authenticated clients

- DHCP Options:

  (3) Routers = the IP address of the router to reach the Domain Controller and WINS server

  (44) NetBIOS Name Server = the IP address of the WINS server

  (46) NetBIOS Node Type = the desired NETBIOS node type (P, M, or H Node)

b. In the *unauthenticated* lease pool, set the following parameters and options:

- DHCP Parameter **Client Request Handling Authentication level** = Unauthenticated One Lease Per Client = True.
- DHCP Parameter **Lease Time Default Lease Time** = 30–60 seconds (short lease time recommended).
- DHCP Options:

  (3) Routers = the IP address of the router to reach the Domain Controller and WINS server

  (44) **NetBIOS Name Server** = the IP address of the WINS server

  (46) **NetBIOS Node Type** = the desired NETBIOS node type (P, M, or H Node)

6. Right click the DHCP service and select **Export DHCP Service**.



Note - Client SIC Name and Server SIC Name are case-sensitive.

Appendix B
Glossary

In This Appendix

Acronyms and Abbreviations page 172

Acronyms and Abbreviations

The following acronyms and abbreviations are used in this guide.

Table B-1 Glossary of Terms

OPSEC	Open Platform for Security
OWASP	Open Web Application Security Project
SIC	Secure Internal Connection
SSO	Single Sign On
TIP	Trusted Identification Point
UAA	UserAuthority API
UAS	UserAuthority Server
Windows DC	Windows Domain Controller

THIRD PARTY TRADEMARKS AND COPYRIGHTS

Entrust is a registered trademark of Entrust Technologies, Inc. in the United States and other countries. Entrust's logos and Entrust product and service names are also trademarks of Entrust Technologies, Inc. Entrust Technologies Limited is a wholly owned subsidiary of Entrust Technologies, Inc. FireWall-1 and SecuRemote incorporate certificate management technology from Entrust.

Verisign is a trademark of Verisign Inc.

The following statements refer to those portions of the software copyrighted by University of Michigan. Portions of the software copyright © 1992-1996 Regents of the University of Michigan. All rights reserved. Redistribution and use in source and binary forms are permitted provided that this notice is preserved and that due credit is given to the University of Michigan at Ann Arbor. The name of the University may not be used to endorse or promote products derived from this software without specific prior written permission. This software is provided “as is” without express or implied warranty. Copyright © Sax Software (terminal emulation only).

The following statements refer to those portions of the software copyrighted by Carnegie Mellon University.

Copyright 1997 by Carnegie Mellon University. All Rights Reserved.

Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of CMU not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission.CMU DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL CMU BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

The following statements refer to those portions of the software copyrighted by The Open Group.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE OPEN GROUP BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The following statements refer to those portions of the software copyrighted by The OpenSSL Project. This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (http://www.openssl.org/).

THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT ``AS IS'' AND ANY * EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE OpenSSL PROJECT OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The following statements refer to those portions of the software copyrighted by Eric Young. THIS SOFTWARE IS PROVIDED BY ERIC YOUNG ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE. Copyright © 1998 The Open Group.

The following statements refer to those portions of the software copyrighted by Jean-loup Gailly and Mark Adler Copyright (C) 1995-2002 Jean-loup Gailly and Mark Adler. This software is provided 'as-is', without any express or implied warranty. In no event will the authors be held liable for any damages arising from the use of this software. Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.

2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.

3. This notice may not be removed or altered from any source distribution.

The following statements refer to those portions of the software copyrighted by the Gnu Public License. This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version. This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.

The following statements refer to those portions of the software copyrighted by Thai Open Source Software Center Ltd and Clark Cooper Copyright (c) 2001, 2002 Expat maintainers. Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions: The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software. THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE. GDChart is free for use in your applications and for chart generation. YOU

MAY NOT re-distribute or represent the code as your own. Any re-distributions of the code MUST reference the author, and include any and all original documentation. Copyright. Bruce Verderaime. 1998, 1999, 2000, 2001. Portions copyright 1994, 1995, 1996, 1997, 1998, 1999, 2000, 2001, 2002 by Cold Spring Harbor Laboratory. Funded under Grant P41-RR02188 by the National Institutes of Health. Portions copyright 1996, 1997, 1998, 1999, 2000, 2001, 2002 by Boutell.Com, Inc. Portions relating to GD2 format copyright 1999, 2000, 2001, 2002 Philip Warner. Portions relating to PNG copyright 1999, 2000, 2001, 2002 Greg Roelofs. Portions relating to gdttf.c copyright 1999, 2000, 2001, 2002 John Ellson (ellson@graphviz.org). Portions relating to gdft.c copyright 2001, 2002 John Ellson (ellson@graphviz.org). Portions relating to JPEG and to color quantization copyright 2000, 2001, 2002, Doug Becker and copyright (C) 1994, 1995, 1996, 1997, 1998, 1999, 2000, 2001, 2002, Thomas G. Lane. This software is based in part on the work of the Independent JPEG Group. See the file README-JPEG.TXT for more information. Portions relating to WBMP copyright 2000, 2001, 2002 Maurice Szmurlo and Johan Van den Brande. Permission has been granted to copy, distribute and modify gd in any context without fee, including a commercial application, provided that this notice is present in user-accessible supporting documentation. This does not affect your ownership of the derived work itself, and the intent is to assure proper credit for the authors of gd, not to interfere with your productive use of gd. If you have questions, ask. "Derived works" includes all programs that utilize the library. Credit must be given in user-accessible documentation. This software is provided "AS IS." The copyright holders disclaim all warranties, either express or implied, including but not limited to implied warranties of merchantability and fitness for a particular purpose, with respect to this code and accompanying documentation. Although their code does not appear in gd 2.0.4, the authors wish to thank David Koblas, David Rowley, and Hutchison Avenue Software Corporation for their prior contributions.

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at http://www.apache.org/licenses/LICENSE-2.0

The curl license

COPYRIGHT AND PERMISSION NOTICE

Copyright (c) 1996 - 2004, Daniel Stenberg, <daniel@haxx.se>.All rights reserved.

Permission to use, copy, modify, and distribute this software for any purpose

with or without fee is hereby granted, provided that the above copyright

notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name of a copyright holder shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization of the copyright holder.

The PHP License, version 3.0

Copyright (c) 1999 - 2004 The PHP Group. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, is permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. The name "PHP" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact group@php.net.

4. Products derived from this software may not be called "PHP", nor may "PHP" appear in their name, without prior written permission from group@php.net. You may indicate that your software works in conjunction with PHP by saying "Foo for PHP" instead of calling it "PHP Foo" or "phpfoo"

5. The PHP Group may publish revised and/or new versions of the license from time to time. Each version will be given a distinguishing version number. Once covered code has been published under a particular version of the license, you may always continue to use it under the terms of that version. You may also choose to use such covered code under the terms of any subsequent version of the license published by the PHP Group. No one other than the PHP Group has the right to modify the terms applicable to covered code created under this License.

6. Redistributions of any form whatsoever must retain the following acknowledgment:

"This product includes PHP, freely available from <http://www.php.net/>".

THIS SOFTWARE IS PROVIDED BY THE PHP DEVELOPMENT TEAM ``AS IS'' AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE PHP DEVELOPMENT TEAM OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This software consists of voluntary contributions made by many individuals on behalf of the PHP Group. The PHP Group can be contacted via Email at group@php.net.

For more information on the PHP Group and the PHP project, please see <http://www.php.net>. This product includes the Zend Engine, freely available at <http://www.zend.com>.

This product includes software written by Tim Hudson (tjh@cryptsoft.com).

Copyright (c) 2003, Itai Tzur <itzur@actcom.co.il>

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistribution of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Neither the name of Itai Tzur nor the names of other contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR

CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS

INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright (c) 1998, 1999, 2000 Thai Open Source Software Center Ltd

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions: The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright © 2003, 2004 NextHop Technologies, Inc. All rights reserved.

Confidential Copyright Notice

Except as stated herein, none of the material provided as a part of this document may be copied, reproduced, distrib-uted, republished, downloaded, displayed, posted or transmitted in any form or by any means, including, but not lim-ited to, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of NextHop Technologies, Inc. Permission is granted to display, copy, distribute and download the materials in this doc-ument for personal, non-commercial use only, provided you do not modify the materials and that you retain all copy-right and other proprietary notices contained in the materials unless otherwise stated. No material contained in this document may be "mirrored" on any server without written permission of NextHop. Any unauthorized use of any material contained in this document may violate copyright laws, trademark laws, the laws of privacy and publicity, and communications regulations and statutes. Permission terminates automatically if any of these terms or condi-tions are breached. Upon termination, any downloaded and printed materials must be immediately destroyed.

Trademark Notice

The trademarks, service marks, and logos (the "Trademarks") used and displayed in this document are registered and unregistered Trademarks of NextHop in the US and/or other countries. The names of actual companies and products mentioned herein may be Trademarks of their respective owners. Nothing in this document should be construed as granting, by implication, estoppel, or otherwise, any license or right to use any Trademark displayed in the document. The owners aggressively enforce their intellectual property rights to the fullest extent of the law. The Trademarks may not be used in any way, including in advertising or publicity pertaining to distribution of, or access to, materials in

this document, including use, without prior, written permission. Use of Trademarks as a "hot" link to any website is prohibited unless establishment of such a link is approved in advance in writing. Any questions concerning the use of these Trademarks should be referred to NextHop at U.S. +1 734 222 1600.

U.S. Government Restricted Rights

The material in document is provided with "RESTRICTED RIGHTS." Software and accompanying documentation are provided to the U.S. government ("Government") in a transaction subject to the Federal Acquisition Regulations with Restricted Rights. The Government's rights to use, modify, reproduce, release, perform, display or disclose are

restricted by paragraph (b)(3) of the Rights in Noncommercial Computer Software and Noncommercial Computer Soft-ware Documentation clause at DFAR 252.227-7014 (Jun 1995), and the other restrictions and terms in paragraph (g)(3)(i) of Rights in Data-General clause at FAR 52.227-14, Alternative III (Jun 87) and paragraph (c)(2) of the Commer-cial

Computer Software-Restricted Rights clause at FAR 52.227-19 (Jun 1987).

Use of the material in this document by the Government constitutes acknowledgment of NextHop's proprietary rights in them, or that of the original creator. The Contractor/Licensor is NextHop located at 1911 Landings Drive, Mountain View, California 94043. Use, duplication, or disclosure by the Government is subject to restrictions as set forth in applicable laws and regulations.

Disclaimer Warranty Disclaimer Warranty Disclaimer Warranty Disclaimer Warranty

THE MATERIAL IN THIS DOCUMENT IS PROVIDED "AS IS" WITHOUT WARRANTIES OF ANY KIND EITHER EXPRESS OR IMPLIED. TO THE FULLEST EXTENT POSSIBLE PURSUANT TO THE APPLICABLE LAW, NEXTHOP DISCLAIMS ALL WARRANTIES,

EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NON INFRINGEMENT OR OTHER VIOLATION OF RIGHTS. NEITHER NEXTHOP NOR ANY OTHER PROVIDER OR DEVELOPER OF MATERIAL CONTAINED IN THIS DOCUMENT WARRANTS OR MAKES ANY REPRESEN-TATIONS REGARDING THE USE, VALIDITY, ACCURACY, OR RELIABILITY OF, OR THE RESULTS OF THE USE OF, OR OTHERWISE RESPECTING, THE MATERIAL IN THIS DOCUMENT.

Limitation of Liability

UNDER NO CIRCUMSTANCES SHALL NEXTHOP BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES, INCLUDING, BUT NOT LIMITED TO, LOSS OF DATA OR PROFIT, ARISING OUT OF THE USE, OR THE INABILITY TO USE, THE MATERIAL IN THIS DOCUMENT, EVEN IF NEXTHOP OR A NEXTHOP AUTHORIZED REPRESENTATIVE HAS ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. IF YOUR USE OF MATERIAL FROM THIS DOCUMENT RESULTS IN THE NEED FOR SERVICING, REPAIR OR CORRECTION OF EQUIPMENT OR DATA, YOU ASSUME ANY COSTS THEREOF. SOME STATES DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THE ABOVE LIMITATION OR EXCLUSION MAY NOT FULLY APPLY TO YOU.

Copyright © ComponentOne, LLC 1991-2002. All Rights Reserved.

BIND: ISC Bind (Copyright (c) 2004 by Internet Systems Consortium, Inc. ("ISC"))

Copyright 1997-2001, Theo de Raadt: the OpenBSD 2.9 Release

PCRE LICENCE

PCRE is a library of functions to support regular expressions whose syntax and semantics are as close as possible to those of the Perl 5 language. Release 5 of PCRE is distributed under the terms of the "BSD" licence, as specified below. The documentation for PCRE, supplied in the "doc" directory, is distributed under the same terms as the software itself.

Written by: Philip Hazel <ph10@cam.ac.uk>

University of Cambridge Computing Service, Cambridge, England. Phone:

+44 1223 334714.

Copyright (c) 1997-2004 University of Cambridge All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of the University of Cambridge nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Index

A

APIs
  Assertions Management 126
  event handlers 136
Auditing
  configuring, for external resources 92
  Outbound Traffic using UserAuthority Outbound Access Control 88
  overview 11, 86
  using logs 87
Automatic synchronization 98
  using db_sync Script 98

C

Citrix MetaFrame deployment 32
Citrix MetaFrame Server or Windows Terminal Services deployment
  sample deployment 32
  workflow 33
Citrix MetaFrame Servers
  Outbound Access Control 73
Citrix MetaFrame Terminals
  Outbound Access Control 67
Clustering 94
  VPN-1 Pro 97
Clusters
  VPN-1 Pro 97
Configuring
  UserAuthority Server 41
  UserAuthority Server properties 51
Credentials Manager
  automatic synchronization 98
  synchronizing 97

D

Databases
  using local Check Point 81
db_sync script 98
DC 167
Debugging 135
Deployment
  Citrix MetaFrame Server or Windows Terminal Services 32
  Outbound Access Control 27
  overview 26
  SSO for VPN-1 Pro 21
Deployments
  typical 21
DHCP server configuration 170
Domain Controllers
  adding 72
Domain equality
  configuring 73

E

Event Handlers 136
  UAA_AUTHENTICATE_REPLY 138
  UAA_QUERY_REPLY 136
  UAA_UPDATE_REPLY E 137
Event Handling 116
External database 82
External resources
  auditing requests 92
  monitoring access to 88

F

Function calls 125
  assertions Iteration 132
  assertions management 126
  debugging 135
  managing authentication requests 131
  managing queries 129
  managing UAA Errors 134
  managing updates 130
  session management 125

G

Groups
  defining 66

H

High availability 94
  using a VPN-1 Pro cluster 97
  using Multiple Domain Controllers 96

I

Identification
  using SecureAgent 61
Identity sharing 61
  configuring manual 62
Installing
  UserAuthority license 35
  UserAuthority on UNIX/Linux-based machine 40
  UserAuthority Server 35
  UserAuthority Server on Domain Controller 47

K

Key assertions 117

L

LDAP database 82
License
  installing 35
Load balancing 94
Logs
  configuring for UserAuthority Server on the FireWall Gateway 145
  configuring for UserAuthority Servers not on a FireWall Gateway 146
  use in auditing 87
  viewing 87

M

Manual Identity sharing
  configuring 62
Meta IP 164
  DHCP server configuration 170
  Domain Controller configuration 167
  VPN-Pro policy configuration 168
  Windows Domain Controller configuration 167
Module status types 143
Monitoring 142
  system 143
  system status 143
  user 148
Multiple Domain Controllers 96
  Outbound Access Control 68
Multiple domains
  Outbound Access Control 70

O

OPSEC APIs 23, 109
  overview 114
OPSEC protocols 23
Outbound Access Control 27, 57
  identity sharing 61
  multiple domains 70
  on Citrix MetaFrame Servers 73
  UserAuthority solution 59
Outbound Access Control deployment
  components 27
  sample deployment 28
  workflow 28

P

Policy
  VPN-1 Pro 80
Programming model 111

R

Request Assertions 118
Requests for external resources
  configuring auditing 92

S

SecureAgent
  automatic installation 54
  Outbound Access Control 61
SIC
  reestablishing 154
  verifying status 154
SmartDashboard
  creating groups 66
SmartView Tracker interface 87
SSO
  establishing for VPN-1/Firewall-1 29
  UserAuthority solution for VPN-1 57
SSO for VPN-1 Pro 57
  on Citrix terminals 67
  UserAuthority solution 59
SSO for VPN-1 Pro deployment 21
  adding SSO rule 28
  on Windows Terminal Services 73
SSO rules
  creating 29
  creating for SSO for Citrix MetaFrame or Windows Terminal Services 33
System monitoring 143
  UserAuthority Server 144
  using UserAuthority Server logs 145
  using WebAccess logs 145

T

Testing deployment
  Outbound Access Control 29, 33
Troubleshooting
  general problems 153
  no established SIC 153
  SecureAgent does not identify the user 157
  User-related problems 157
Trusted Identification Points 20, 61

U

UAA Assertions structure functions 124
UAA Client
  application structure 114
  event handling 116
  key assertions 117
  request assertions 118
  requests 116
  server configuration 114
UAA errors 134
UAS Groups
  creating 65
User Groups
  in UserAuthority 79
User Identity
  providing for VPN-1 Pro 59
User monitoring 148
  example of unsuccessful access attempt 149
UserAuthority
  advantages 18
  installing license 35

integrating with Meta IP 164
introduction 18
Queries 123
underlying concept 20
UserAuthority API 109
function calls 125
overview 114
programming model 111
UserAuthority CLIs 102
UserAuthority Server
configuring 41
configuring SecureAgent
automatic installation 54
installing on a Windows
gateway 36
installing on Domain
Controller 47
installing on VPN-1 Pro 35
monitoring 144
UserAuthority Server
properties 51
Users
in UserAuthority 79
managing 78

V

VPN-1 Pro
clusters 97
defining authentication
actions in authentication
policy 80
deployment with multiple
domain controllers 68
deployment with multiple
domains 70
Outbound Access Control 57
policy configuration 168

W

WebAccess
configuring to recogize
Windows user groups 83
Windows Groups
retrieving with
UserAuthority 66
Windows Terminal Services 73
deployment 32
Windows user identity
using 83