SecuRemote / SecurID
Implementation on Nokia VPN-1 Appliance

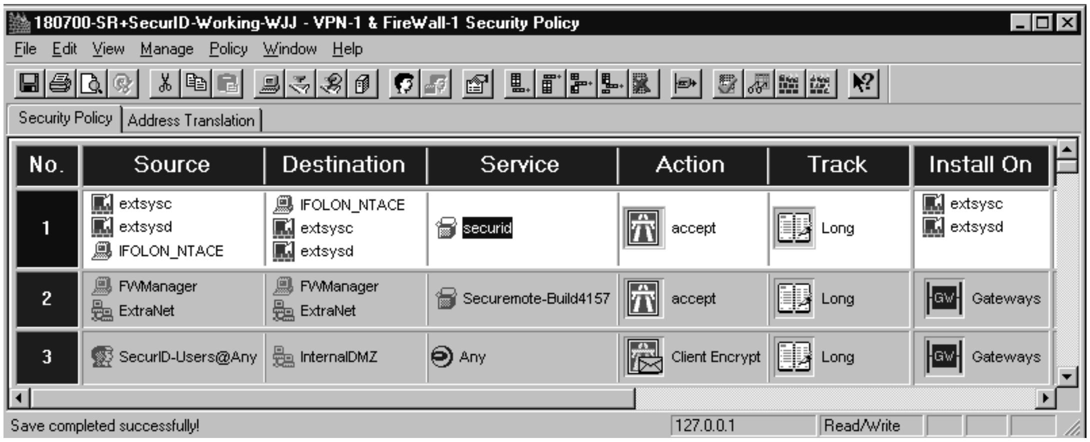
Author: Will Jones(widge_it@msn.com)
Date: 19 July 2000

Tested Configuration:
Securemote Build: 4157
SecurID Version: 4
SecuRemote License: srlight (25 users)
Encryption License: DES
IPSO release: 3.2.1
FW Version: 4.0 sp4

SecuRemote provides VPN capabilities for remote connections. Specifically it allows for encrypted connections either across private networks, or tunneled over the public Internet.

SecurID works in conjunction with SecuRemote to provide token based one-time-password authentication. It requires users to be authenticated by a separate Authentication Server, rather than the Firewall, and provides a far stronger method of authentication than user passwords provide.

Rulebase and Explanation


180700-SR+SecurID-Working-WJJ - VPN-1 & FireWall-1 Security Policy
File Edit View Manage Policy Window Help
Security Policy
Address Translation

No.	Source	Destination	Service	Action	Track	Install On
1	extsysc extsysd IFOLON_NTACE	IFOLON_NTACE extsysc extsysd	securid	accept	Long	extsysc extsysd
2	FWManager ExtraNet	FWManager ExtraNet	Securemote-Build4157	accept	Long	Gateways
3	SecurID-Users@Any	InternalDMZ	Any	Client Encrypt	Long	Gateways

Save completed successfully! 127.0.0.1 Read/Write

The Firewall contains three rules which are relevant to the operation of SecuRemote and SecurID.

SecurID Rule
Rule 1:   This rule is required to allow the SecurID protocol (UDP 5500) to pass between the external FW's and the ACE Server on the Internal DMZ. To check that this is working correctly, TCPDUMP port 5500 on the interface on the Firewall to which the ACE Server is attached. Here is the output of what you should see from the Firewall(the IP addresses will depend on your own IP configuration):

```
extsysc[admin]# tcpdump -i eth-s1p1 port 5500
tcpdump: listening on eth-s1p1
07:14:16.120865 198.198.198.1.2203 > 10.68.3.5.5500: udp 124
07:14:16.121316 10.68.3.5.5500 > 198.198.198.1.2203: udp 124
07:14:16.129878 198.198.198.1.2204 > 10.68.3.5.5500: udp 124
07:14:17.146527 10.68.3.5.5500 > 198.198.198.1.2204: udp 124
```

To get the above output on the Firewall, at the client machine I tried to establish an FTP session to an FTP server in my Encryption Domain. As a result, as soon as the request hit the Firewall, the Firewall passed the request off to the ACE Server for authentication. Back on the client machine, I was prompted by SecuRemote for my username/passcode credentials. If I didn't have rule 1 in my Firewall, this authentication pass-off would not have worked, as there wouldn't have been a rule to allow connections to the ACE Server on port 5500. I will explain more about how SecurID works after explaining more about the other two rules.

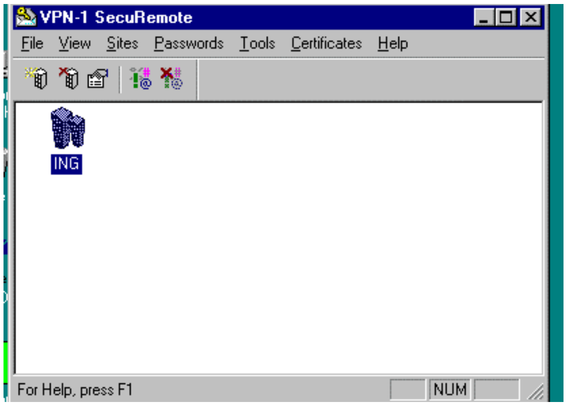
SecuRemote Rule

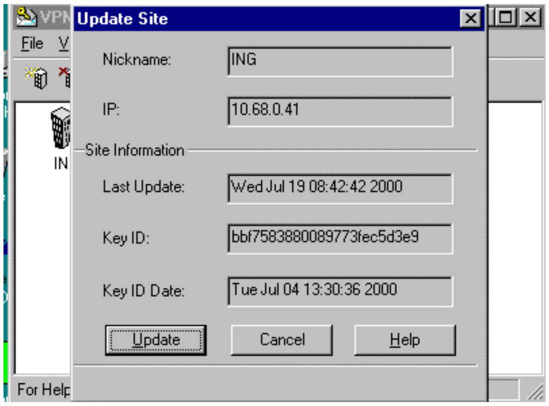
Rule 2: This rule enables connectivity between the SecuRemote client (in this case situated on the Extranet) and the Firewall Management Console in order for the SecuRemote Clients to get Topology information from the Management Console. With SecuRemote build 4157 the ports required to be opened on the Firewall are:

TCP 264
TCP 256

These ports are required to be open as before the SecuRemote client can establish encrypted sessions with the Firewall, it needs to receive topology information (what the encryption domain is) and encryption keys. This is what happens when you first install SecuRemote on the client machine.

SecuRemote on the Client





Nickname: ING
IP: 10.68.0.41
Site Information
Last Update: Wed Jul 19 08:42:42 2000
Key ID: bbf7583880089773fec5d3e9
Key ID Date: Tue Jul 04 13:30:36 2000

Upon installation of SecuRemote on the client machine, you will be asked to input the IP address (and "nickname" if you wish – this is now available with build 4157) of the Firewall Mangagement Console. Once you have done this, the SecuRemote client attempts to connect to the Management Console in order to receive Topology information and encryption keys. (When you initially set this up it would be wise to log this as "long" on the Firewall. By doing this you can see the key exchange happening within the Firewall log.

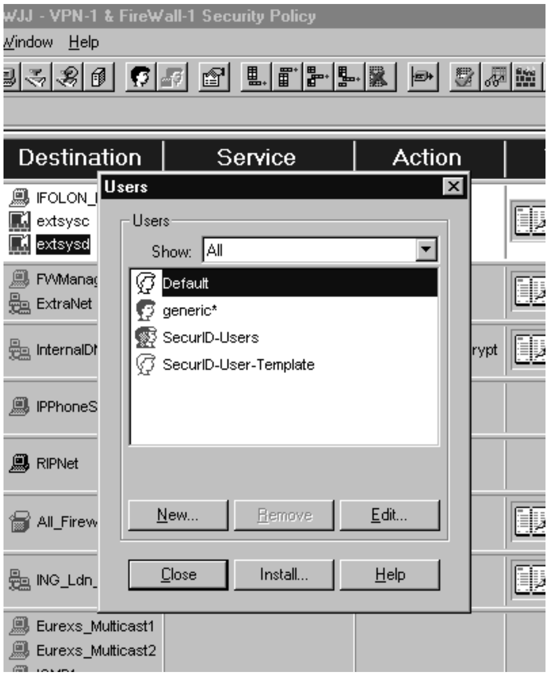
The second screenshot (right) shows a successful receipt of topology and encryption key exchange on the client machine. If you are unable to get this to work it is likely that the cause is a connectivity problem between your client and the Firewall Management Console (refer to rule 2).

User Access Rule

The third rule is the rule that determines access rights to the Encryption Domain setup on your firewall. In our example we are using SecurID as the authentication mechanism, and SecuRemote as the encryption/VPN mechanism.

Third-Party Authentication mechanisms such as SecurID require a separate Authentication database to be used instead of the Firewall itself. In order to do this the Firewall must know which users need to authenticated by the third-party, and which don't. Firewall-1 provides a user called "**generic***" for just this purpose.

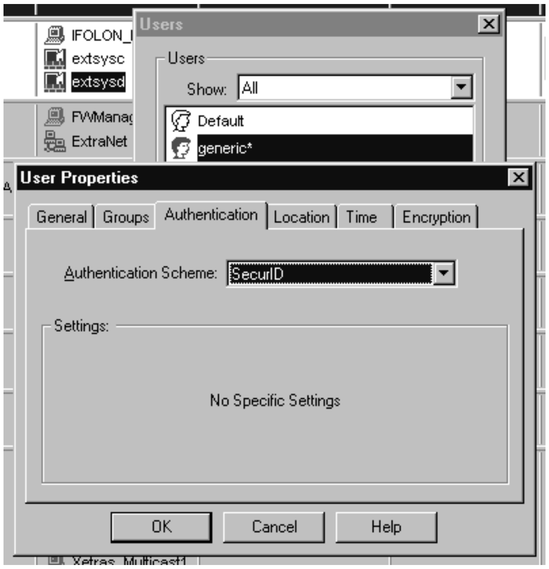
The Firewall Users Database (showing generic* user and details)



Users

Show: All

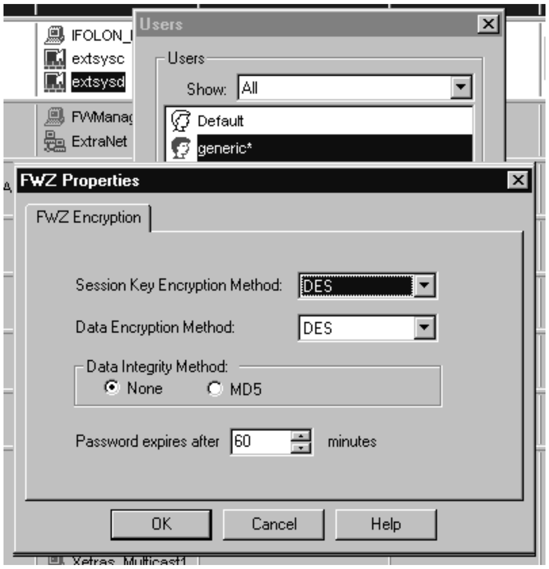
- Default
- generic*
- SecurID-Users
- SecurID-User-Template



Authentication Scheme: SecurID

Settings:

No Specific Settings



Session Key Encryption Method: DES

Data Encryption Method: DES

Data Integrity Method: ● None ○ MD5

Password expires after 60 minutes

The screenshots show that the generic* user has been created within the Firewall User Database. The screenshots also show that for this user the authentication process to use is SecurID, whilst the encryption to use is DES. Also note the group called SecurID-Users. This is the group to which generic* is a member – the only member.

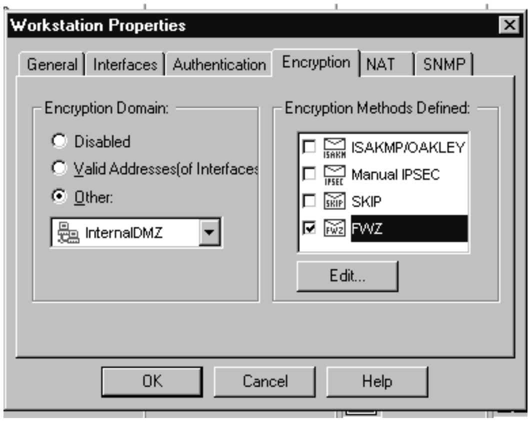
Lets look closer at the rule:

Source	**Destination**	**Service**	**Action**
Securid-Users@Any	InternalDMZ	Any	Client Encrypt

This rule states that for any connection originating from a user within the SecurID-Users group trying to connect to a device within the InternalDMZ using any service, the connection should be authenticated by a third-party (remember who the user in the group is), and should be encrypted. Its important to note that the link is encrypted because the destination is the InternalDMZ. The InternalDMZ is the Encryption Domain that is setup on the Firewall object within the Firewall database. Without this information SecuRemote would not work, as it wouldn't know which hosts required encryption and which hosts didn't. The Encryption Domain is the subnet that you require to be encrypted.

Scenario:
You have remote locations that wish to connect to your webserver and your ftpserver. For security reasons you don't want to put these servers on your internal network, so you put them on a separate subnet (DMZ) in order to manage who gets access and who doesn't. Connections from remote networks into your DMZ need to traverse your Firewall in order to get access. On your Firewall object in the Firewall database you need to set your Encryption Domain to be the DMZ network. By doing this, you are telling the firewall that any remote connection to the DMZ network needs to be encrypted.

Encryption Domain Setup within the Firewall Object



Workstation Properties

Encryption Domain:
- ○ Disabled
- ○ Valid Addresses(of Interfaces
- ● Other: InternalDMZ

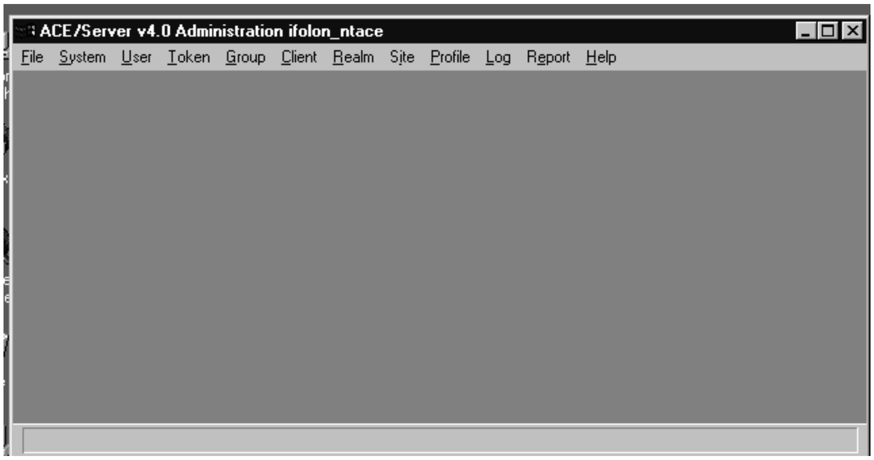
Encryption Methods Defined:
- ☐ ISAKMP/OAKLEY
- ☐ Manual IPSEC
- ☐ SKIP
- ☑ FWZ

Whilst setting up the Encryption Domain is vital for successful use of SecuRemote, you must also make sure the external interfaces are correct. That is to say, you have correctly licensed your firewall so that the license is applied to the external interface (public facing) IP address. On the Nokia platform you need to enter the external interface into the EXTERNAL.IF file located in the $FWDIR/conf directory. The entry should contain the physical interface name plus c0 (eg. eth-s1p1c0). If you don't do this, the Firewall will not know which interface is public facing, which is relevant to connections into the encryption domain.

SecurID Implementation

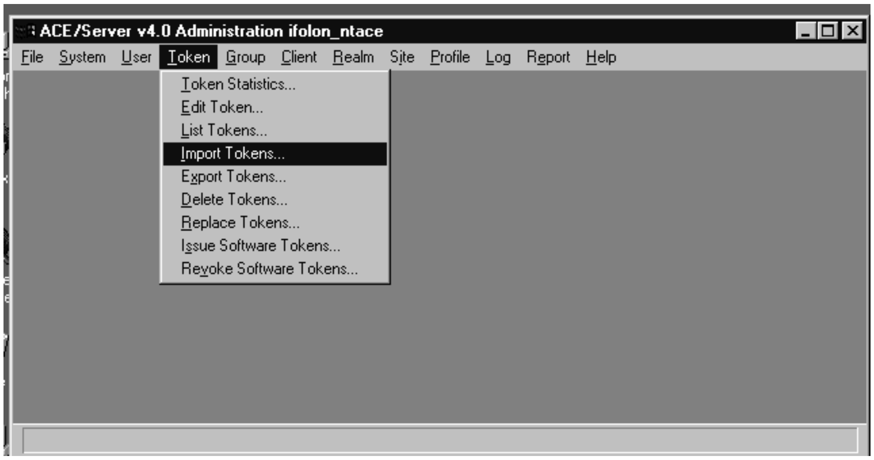
So now we have the Firewall configured for SecurID (and SecuRemote), we need to concentrate on getting the ACE Server up and configured. Before attempting to install the software you need to get the decrypt key from RSA in order for you to register your seed record and license file. For each batch of Tokens you will receive a floppy disk. This disk contains a .ASC file that is encrypted. When you register your product purchase with RSA they will send you a decrypt key in the form of an executable file. Double click this file, enter the passphrase you have received with the executable and press enter. This creates a separate file which contains the decrypted data for each encrypted file.

Once you have installed the software and registered your seed records/tokens, you need to go into Database Configuration - Host Mode – this is off of the ACE Server menu:


ACE/Server v4.0 Administration ifolon_ntace
File System User Token Group Client Realm Site Profile Log Report Help

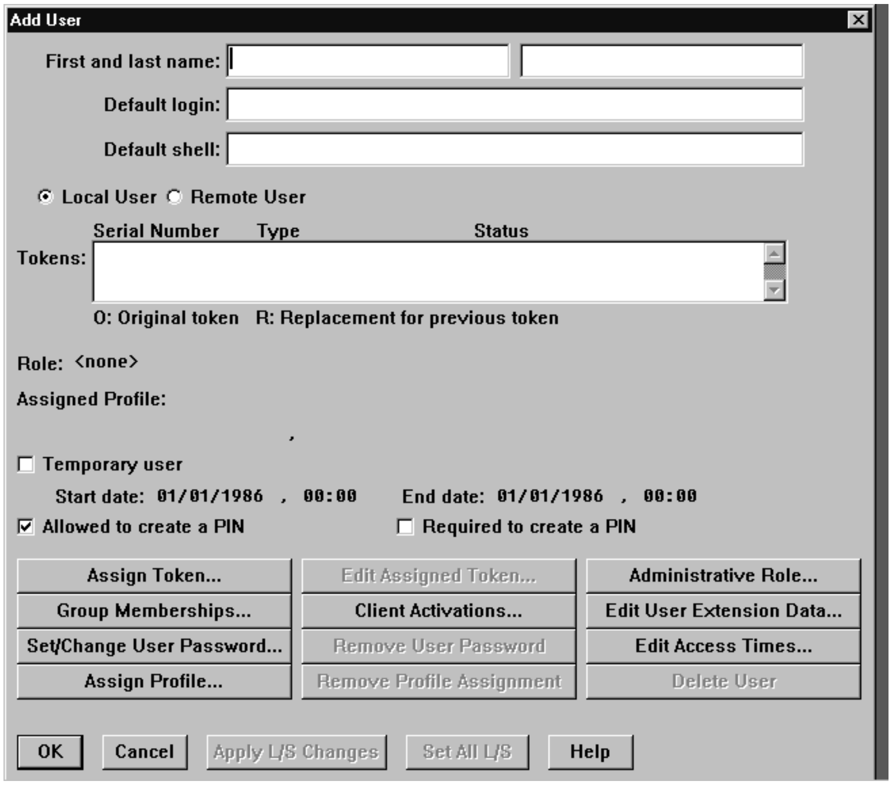
Firstly, import the tokens into the database. To do this, from the menu select Token – Import. Next, select the .ASC file you have saved on the server. This will import the token information from the .ASC file into the ACE Database.

Importing Tokens into the ACE Database


ACE/Server v4.0 Administration ifolon_ntace
File System User Token Group Client Realm Site Profile Log Report Help
Token Statistics...
Edit Token...
List Tokens...
Import Tokens...
Export Tokens...
Delete Tokens...
Replace Tokens...
Issue Software Tokens...
Revoke Software Tokens...

Next, create a dummy user for testing purposes. Select “Add User” from the User menu. Once in the Add User screen, enter the Users’ first and last name, and more importantly, enter the default login. The default login is the username that will be used to authenticate the remote User. Next, select whether or not the user will be allowed to create a PIN to go with their Token, if this isn’t selected, one will be generated automatically when they first connect to a device in the Encryption Domain. Next, select “Assign Token” – you are then able to link the User to a Token. Each token has a unique serial number (usually on the back of the keyfob). You will see a list of these serial numbers, to which you can assign your user to (make sure you give the user the correct keyfob once you have done this !).

Add User Screen



Add User

First and last name:

Default login:

Default shell:

● Local User ○ Remote User

Tokens: Serial Number Type Status

O: Original token R: Replacement for previous token

Role: <none>

Assigned Profile:

☐ Temporary user

Start date: 01/01/1986 , 00:00 End date: 01/01/1986 , 00:00

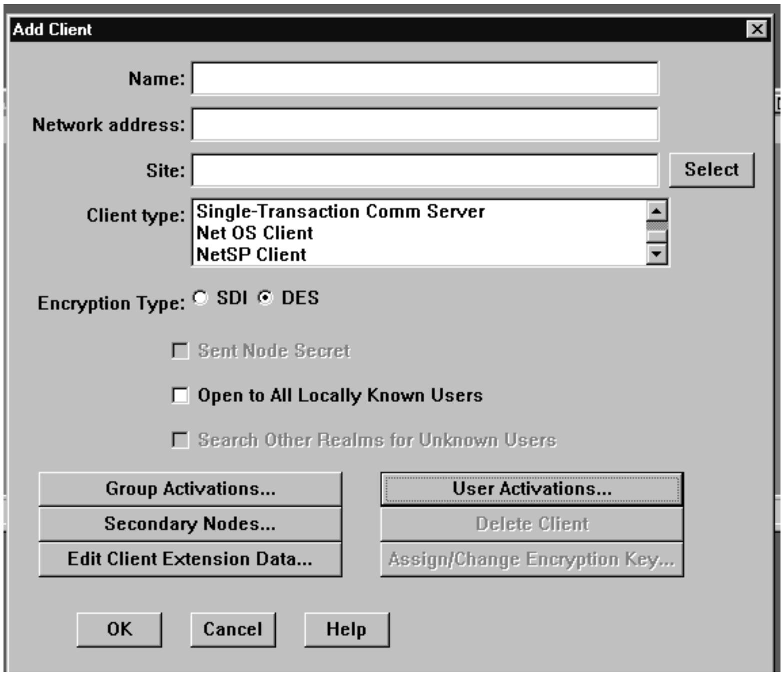
☑ Allowed to create a PIN ☐ Required to create a PIN

Once you have created Clients, you will then be able to activate the User on individual or groups of Clients. Do this by going into the Client Activations… screen of the Add User menu. Obviously, you could create the Clients first and then add the Users. It doesn't matter which way round you do this.

Adding clients

As far as the ACE Database is concerned, Clients are not users they are actual machines. Clients can range from NT Workstations to Firewall Interfaces. Effectively, devices within your Encryption Domain (that you allow remote connections to) need to be added as Clients in the ACE Database. It is IMPORTANT that both the Firewall and the ACE Server can resolve correctly the names of the Clients in the ACE Database – Authentication may not work if this is not the case. If you're not using DNS (or you wish to resolve names as quick as possible), enter the names and IP addresses of all the your ACE Database Clients into the ACE Server's hosts file. Make sure to also enter these Client names into the Firewall. Next, select the Client Type. This will obviously depend on what the Client is. The Nokia platform needs to be input as a Communication Server, whilst the NT web and ftp servers are input as NET OS Clients. Next select DES as the encryption method (which is what we are using). Finally, whilst testing, select the "Open to All Locally Known Users" checkbox. This means that the Client will accept connections from any and all SecurID Users. This is good to use whilst testing, but once you have finished testing you should deselect this option and instead, select the User Activations… screen whereupon you can determine which SecurID users are allowed access to each Client.

Add Client



Name:

Network address:

Site:

Client type: Single-Transaction Comm Server
Net OS Client
NetSP Client

Encryption Type: ○ SDI ● DES

□ Sent Node Secret

□ Open to All Locally Known Users

□ Search Other Realms for Unknown Users

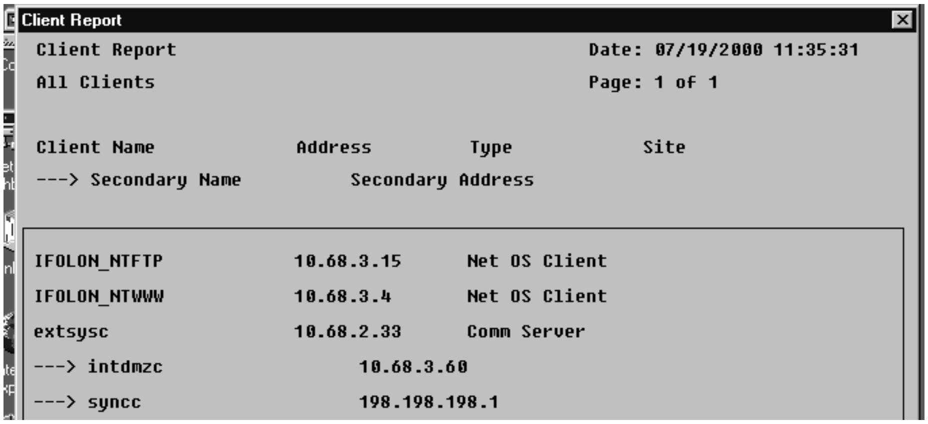
Primary Interface

This is the interface referenced by the EXTERNAL.IF file in $FWDIR/conf. The ACE Server needs to know this in order for authentication to work at all.

Secondary Nodes

Secondary Nodes are any other Firewall interface other than the primary interface. The ACE Server needs to know about all the Firewall interfaces, not just the primary interface. Again, without this information, authentication will not work.

ACE Server – Known Clients



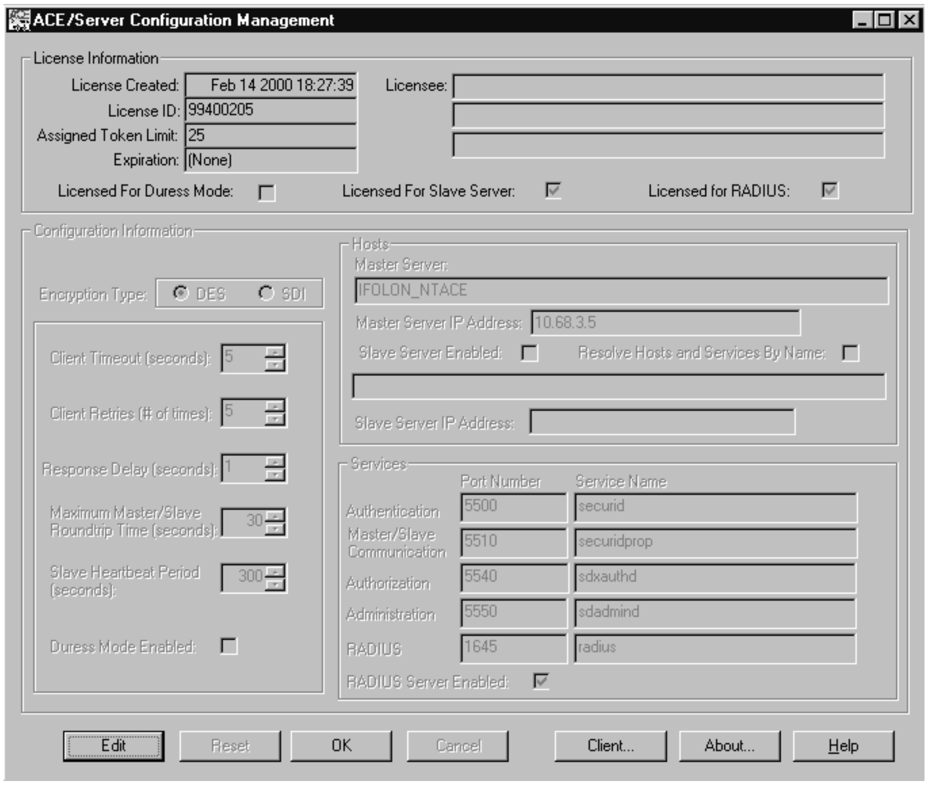
```
Client Report                                         Date: 07/19/2000 11:35:31
All Clients                                           Page: 1 of 1

Client Name            Address         Type           Site
---> Secondary Name          Secondary Address

IFOLON_NTFTP           10.68.3.15      Net OS Client
IFOLON_NTWWW           10.68.3.4       Net OS Client
extsysc                10.68.2.33      Comm Server
---> intdmzc                 10.68.3.60
---> syncc                   198.198.198.1
```

In the screenshot above I have generated a Client Report (from select Client – List Client). This shows what Clients are configured within the ACE Server database. Underneath the Comm Server, notice the two indented Clients; these are Secondary Nodes (interfaces) of the Comm Server (Firewall). From this screenshot you can also determine the external interface of the Firewall is 10.68.2.33, as this is the Comm Server (extsysc).

SDCONF Generation

The final part of the SecurID implementation is to generate the SDCONF.REC file. As you can see from the screenshot below, the SDCONF.REC file contains crucial parameters required by the SecurID. These parameters are all adjustable from within the Configuration Management screen, which can be accessed from the Start menu, by selecting ACE Server, Configuration Management.

Configuration Management



ACE/Server Configuration Management

License Information

License Created: Feb 14 2000 18:27:39
License ID: 99400205
Assigned Token Limit: 25
Expiration: (None)
Licensee:

Licensed For Duress Mode: □ Licensed For Slave Server: ☑ Licensed for RADIUS: ☑

Configuration Information

Encryption Type: ● DES ○ SDI

Client Timeout (seconds): 5
Client Retries (# of times): 5
Response Delay (seconds): 1
Maximum Master/Slave Roundtrip Time (seconds): 30
Slave Heartbeat Period (seconds): 300
Duress Mode Enabled: □

Hosts

Master Server: IFOLON_NTACE
Master Server IP Address: 10.68.3.5
Slave Server Enabled: □ Resolve Hosts and Services By Name: □
Slave Server IP Address:

Services

	Port Number	Service Name
Authentication	5500	securid
Master/Slave Communication	5510	securidprop
Authorization	5540	sdxauthd
Administration	5550	sdadmind
RADIUS	1645	radius

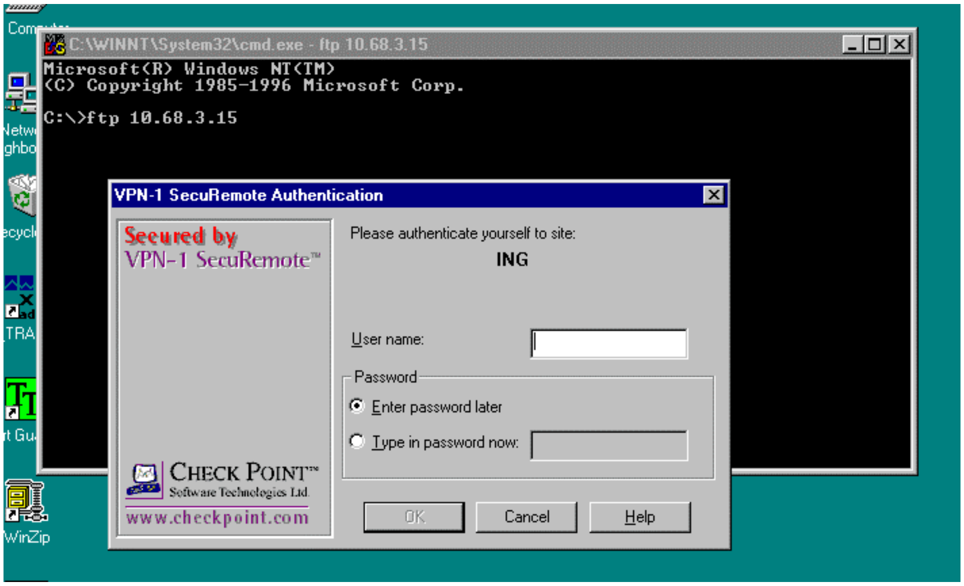
RADIUS Server Enabled: ☑

Before generating the file, make sure the Master Server entry is correct and resolvable and that the correct encryption type is selected. Also, make sure the appropriate services are allowed through the firewall !

When this file is generated it is placed in the \ACE\DATA directory on the ACE Server. This file needs to be copied to the Firewall and placed in a directory called \VAR\ACE (the ACE subdirectory needs to be created). Remember, if copying the file to the Firewall using FTP, be sure to set BIN mode before sending the file (its not an ascii file).

Final Stage
Now you are ready to test the complete setup ! Install the SecuRemote Client on the user workstation, input the IP address (and nickname if you want) of the Firewall Management Console and select enter. If your connectivity and rules are correct (if not read this document again!) you will receive topology information and encryption keys from the Management Console. Once you have this you are ready to test the SecuRemote/SecurID configuration. If you are allowing any service into the your Encryption Domain (as long as the user is authenticated), try to start an FTP or HTTP session to a server in the DMZ. The next screen you see after typing the FTP or HTTP request is something like this:

SecuRemote VPN Authentication Screen



C:\WINNT\System32\cmd.exe - ftp 10.68.3.15

```
Microsoft(R) Windows NT(TM)
(C) Copyright 1985-1996 Microsoft Corp.

C:\>ftp 10.68.3.15
```

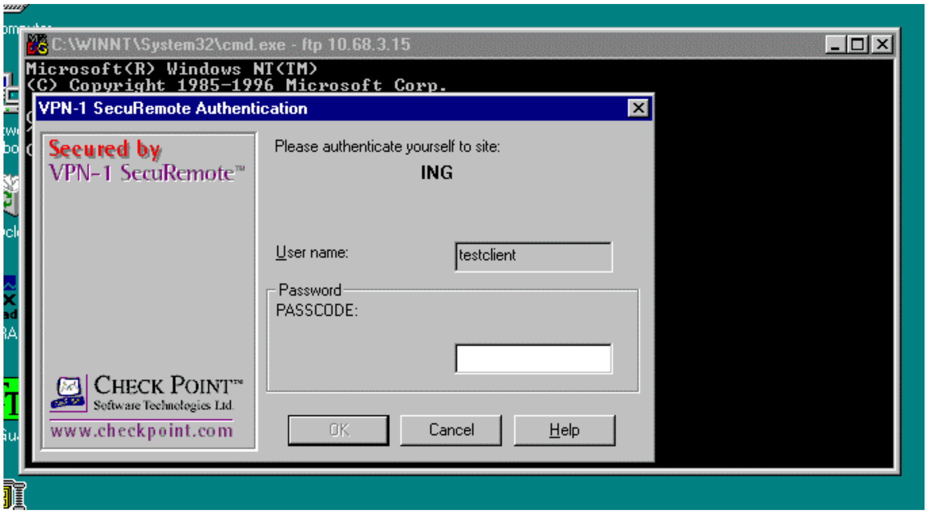
VPN-1 SecuRemote Authentication

Please authenticate yourself to site:
ING

User name:

Password
● Enter password later
○ Type in password now:

As the FTP request is pointing to a server in the Encryption Domain you are challenged to authenticate. In the User Name box enter the user name you have created in the ACE Server database relating to the keyfob issued to that machines user.



C:\WINNT\System32\cmd.exe - ftp 10.68.3.15

```
Microsoft(R) Windows NT(TM)
(C) Copyright 1985-1996 Microsoft Corp.
```

VPN-1 SecuRemote Authentication

Please authenticate yourself to site:
ING

User name: testclient

Password
PASSCODE:

Upon entering the User Name, select the Enter password later radial button. You will now be presented with another box asking you to enter your passcode.

IMPORTANT: The very first time you connect from a client with a new keyfob, in the PASSCODE field, enter just the 6 digit number from the keyfob. Once you have done this, you will be asked to generate a 4-digit PIN number (if you have allowed users the right to do this within the ACE Server User configuration screen). Once you have chosen and input your PIN, every time you are asked to authenticate again, you must enter your PIN followed by whatever the 6-digit number is on the keyfob at that time. Once authentication has been granted the first time, a file is generated on the Firewall, in the \ACE\DATA directory called SECURID. If for some reason you have problems in the future with Clients not being authenticated, you could delete this file on the Firewall which will set all Users to New Pin mode. This means the next time users connect it will be as if they have connected to your site for the first time. Therefore they will have to go through the PIN generation process again.