Global Information Assurance Certification Paper

GSEC PRACTICAL ASSIGNMENT
Version 1.4b, Option 1
(Pre-approved)

Configuring a Cisco PIX to use TACACS+ for authentication of a remote user VPN

By Charles Brodsky
September 3, 2004

Table of Contents

Abstract

The primary objective of this document is to configure a Cisco PIX Firewall to use TACACS+ when authenticating individual users that connect through a Cisco software VPN client. The secondary objective is to explain how Cisco implements IPSec and VPNs as well as examine what each command does.

First a foundation is established by discussing the key features and protocols of VPNs and IPSec. Then Cisco's IPSec implementation and "Easy VPN" design is examined. Next the PIX Firewall is configured to allow remote access VPN connections and the software VPN client is installed and configured. Following that, the Cisco Secure Access Control Server (CSACS) is installed and configured to provide user authentication. Finally the additional PIX commands necessary for Extended Authentication (Xauth) are entered. These commands allow for individual user authentication through the PIX.

The following equipment was used in the creation of this document.

- PIX 515, OS version 6.3(3) with a 3DES license[1]
- Cisco Software VPN Client version 4.0.2(b) on a Windows XP system
- CSACS version 3.3.1 (90 day eval) on a Windows 2000 SP4 server.

[1] **Note:** All the 500 series PIX firewalls use the same commands. The information in this document applies to any model PIX.

IPSec and VPN Basics

A VPN (Virtual Private Network) is essentially a secured connection between two devices over an unsecured network like the Internet. This is typically done by encrypting and/or encapsulating the original packets as data within another packet. The primary benefit is that this allows us to ensure that the data cannot be read by anyone other than the intended recipient or changed in transit. An analogy of this is sending a postcard through the mail. Anyone can read or modify your postcard before it gets to your intended recipient and it would be difficult to tell if something had been changed and who had changed it. If we put the postcard in a sealed, clear envelope (encapsulated it) before we mailed it we could be sure that the postcard wasn't changed by checking that the envelope wasn't tampered with. If we put the postcard in a brown envelope (encryption) we could also be sure that nobody would be able to read what's inside as well.

The way that we make sure the packet (envelope) isn't changed along the way is by using very complex mathematical calculations and a key that is known only to the sender and receiver. The attributes of the packet itself are used in the mathematical formulas and a value is generated and sent with the packet to the destination. That value is called a message digest. The message digest is always a fixed size. This is done by taking a part of the result of the calculations and dropping, or truncating, the rest of the output. The process of creating a message digest is called hashing. Because the hashing process only gives you part of the output it is a one-way only operation. You can't take the message digest and recreate the original packet.

The recipient will repeat the same set of calculations on the packet it receives and then compare its results to the message digest sent with that packet. If both are identical then the packet is considered to be unaltered. The formulas used to generate the message digest are complex enough that the chances of being able to change the packet and still have the same message digest are extremely low. Optionally the packet can be encrypted (brown envelope) to ensure that not only was the data unaltered but that unauthorized individuals can't see what is inside.

The concept of using a VPN is often compared to going through a tunnel. If you are in a tunnel others outside the tunnel can't interact with you. This protects you from anything or anyone outside. By encapsulating our data we get a virtual tunnel for our private information to travel through.

Frequently IPSec is the protocol used to create these VPN tunnels. IP Security (IPSec) (RFC 2401), like the Internet Protocol (IP) suite, is really a collection of many other protocols that work together. The two main protocols in IPSec are Authentication Header (AH) and Encapsulation Security Payload

(ESP). AH ensures that the data has not been changed, authenticates the sender and protects from replay attacks. A replay attack is when an attacker captures some of your packets and resends them. It's like using a tape recorder to play back your words attempting to trick someone else into thinking they are talking to you.

ESP can do all the things that AH can plus it will encrypt your data so that only the intended recipient can read it. If you want to keep your information confidential you need to use ESP. If you only want to make sure that your information isn't changed and you don't care if other people can read it, like sending a post card, you can use AH. Realize that the process of encrypting and decrypting packets takes extra system resources so you will generally see better performance from your devices using AH instead of ESP. This can be an issue if you are using slow or overburdened VPN devices.

IPSec has two modes of operation. The first is called transport mode and is outside the scope of this paper. The second is tunnel mode and this is what we have been describing so far.

Security Associations (SA) are a key concept to understanding and using IPSec. Basically an SA is a relationship between two devices. It functions similarly to a TCP or UDP port in IP. It allows setting up and tracking each conversation and the agreed parameters such as encryption keys, encryption settings and IPSec peers among other things. It is important to realize that Security Associations only work in one direction. If you want to have both sides able to send information to each other you need two SAs.

Another protocol that is frequently used with IPSec is the Internet Key Exchange (IKE) protocol. This is a hybrid protocol made up of the ISAKMP and Oakley standards. It is frequently used to authenticate IPSec peers (end points) and establish encryption keys.

We'll discuss these protocols and their implementation in more detail when we configure our PIX Firewall.

Cisco "Easy VPN" Overview

In order to create VPN connections both sides must agree on the rules of the conversation. Some of these rules include how the end points will prove their identity before creating the tunnel, if encryption will be used, how the keys used for encryption will be generated, how long the keys will be valid before they must be changed, etc.

For organizations with many remote users this can be difficult and time consuming. For smaller organizations with few technical people it can also be a challenge. Enter Cisco's Easy VPN.

Cisco's Easy VPN is built on the Cisco Unified Client Framework and it is designed to push VPN settings to the remote clients automatically upon connection. It enables setting a policy and having it updated and enforced immediately. This results in very little client configuration.

There are two components of the Cisco Easy VPN design: the Easy VPN Server and the Easy VPN Remote. The server can be a router running IOS 12.2(8)t or later, a PIX Firewall running OS 5.0 or later or a dedicated VPN device called a VPN Concentrator. The Remote can be any of the server devices as well as a dedicated VPN Hardware client or Software client on your Windows (Win 9.x-XP), Linux (Intel based), Mac (OS X 10.2) or Solaris (UltraSparc 32 & 64 bit) system. The VPN software client is free from Cisco.

The server will automatically push the required settings such as encryption, key lifetime, etc. to the client during each connection. So if you decide that DES isn't strong enough and you change that parameter to 3DES on the Easy VPN server all Easy VPN remote clients will automatically be configured to use 3DES during their next connection. This can be a great time saver for large deployments while ensuring that all VPN clients are compliant with your VPN security policy.

Basic PIX Remote Access VPN Configuration

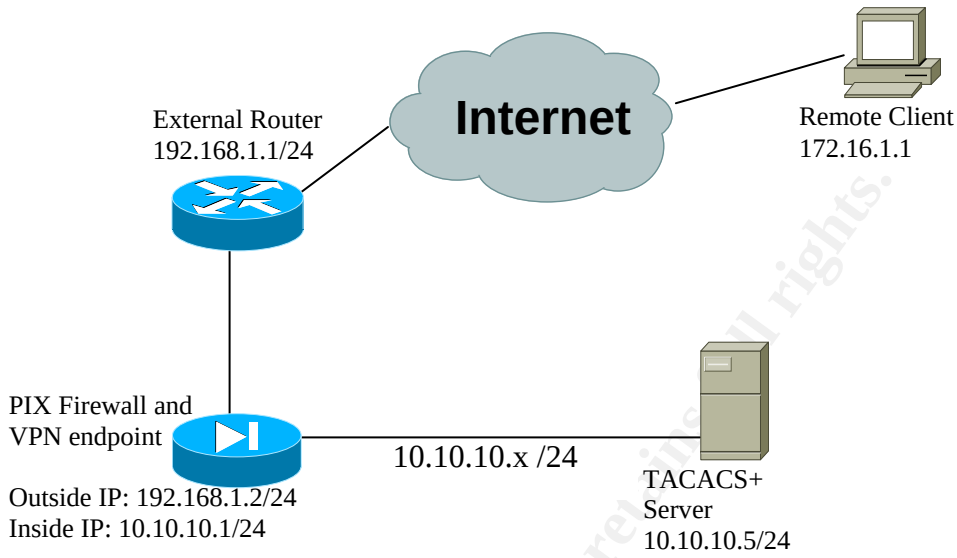
Overview

IKE is a hybrid protocol that provides utility services for IPSec such as authentication of IPSec peers, negotiation of IKE and IPSec Security Associations and the establishment of keys for encryption algorithms. By using IKE the system automatically manages and negotiates many of the parameters that would otherwise have to be configured and modified manually.

Cisco implements IPSec and Remote Access VPNs with five basic steps.

1. Request a VPN Connection (Client Software)
2. IKE Phase One
3. IKE Phase Two
4. Data Transfer
5. IPSec Tunnel Termination

Before we look at each step and the necessary commands let's take a look at the sample network topology. We'll pretend that all the IP addresses and address blocks listed below are publicly routable.



Request a VPN Connection (Client Software)

When a remote user wants to connect to the private network they will first connect to the Internet and then start the VPN client software. The software contacts the VPN device and attempts to authenticate with it. Assuming that the authentication is successful the two devices will create a virtual tunnel for the client to communicate with the internal network.

The next two sections, IKE Phase One and IKE Phase Two, will explain what we need to do to allow these devices to successfully authenticate and connect.

IKE Phase One

During IKE Phase One the peer devices are authenticated and a secure channel is setup between the two devices. This channel is later used to negotiate the IKE Phase Two (IPSec) Security Associations.

IKE Phase One can operate in one of two modes. The first is called main mode. Main mode uses three two-way data exchanges. The first exchange is used to agree on the hash and encryption algorithms for the IKE SA. The next exchange uses the Diffie-Hellman algorithm to generate the shared secret keys.

Finally the peer's identity is verified in the third exchange by sending information in encrypted form.

The second mode is called aggressive mode. In this mode there are only two exchanges. Basically all the information needed is sent in the first exchange. Then the receiver responds and the exchange is complete. Although this mode is faster, it also sends some sensitive information before a secure channel is created.

By default the PIX uses main mode. Since it is more secure than aggressive mode we won't need to change the configuration.

The following PIX commands are used to create the Phase One tunnel. MD5 is the hashing algorithm, 3DES for data encryption of the session keys and a 1024 bit key for the Diffie-Helman key exchange (DH group 2). The key lifetime will be the default 86400 seconds (24 hours.) Shortening the key lifetime will make the session more secure by changing the keys quicker but it also generates more overhead. You have to determine the proper balance for your needs.

```
isakmp policy 100 authentication pre-share
isakmp policy 100 encryption 3des
isakmp policy 100 hash md5
isakmp policy 100 group 2
isakmp policy 100 lifetime 86400
```

The last thing we need to do is enable IKE on the interface(s) that will be used to terminate the tunnels. Since we're using the outside interface, this is done with the following command:

```
isakmp enable outside
```

IKE Phase Two

IKE Phase Two is where the actual IPSec Security Associations that the data travels through are created. For this reason configuring IKE Phase Two is often referred to as configuring IPSec.

Unlike IKE Phase One there is only one mode for IKE Phase Two called "quick mode." It uses the existing, protected tunnel setup by IKE Phase One to securely negotiate its session keys. The keys and key lifetime used for these SAs and tunnels are different than the Phase One keys and they are independently configured. Remember that SAs are unidirectional so one is formed for each direction that data must travel.

In order for this to work correctly we have to give the client an address that systems on the internal LAN can route to. You need to make sure that these addresses are routed to the PIX. If the internal routers already have their default gateways pointing to the PIX you won't need to change any routing statements. The pool of addresses we want to use will be configured on the PIX device itself. This pool will function like a DHCP scope. We will name the pool "clients2" and give them the address range of 10.10.20.1 through 10.10.20.254 with a 255.255.255.0 mask, also known as /24 in bitmask notation. You do not need to include the mask in the configuration. The command to create the pool is:

```
ip local pool clients2 10.10.20.1-10.10.20.254
```

There are several fundamental differences between a Cisco Router and a PIX Firewall. One of them is the fact that the PIX, by design, will want to hide and protect the internal systems from the outside by using NAT and/or PAT to obscure the internal network addresses. NAT stands for Network Address Translation. This is when an internal IP address is translated into a public address. PAT stands for Port Address Translation. This is when one port is translated to another. PAT is often used to allow several inside systems to share a single public IP address. The PIX keeps track of each conversation and the ports used so it knows where to send the response packets too.

We need to tell the PIX not to NAT/PAT the VPN addresses. This is done by putting them in a special group on the PIX called "NAT 0" (pronounced NAT zero.) Any traffic matching an IP address in NAT 0 will be sent through the PIX without changing the source addresses. We can do this by typing the address block in the command. Since our traffic is coming from the inside interface we need to specify that in the command as well.

```
nat (inside) 0 10.10.20.0 255.255.255.0
```

FYI: unlike Cisco Routers, PIX Firewalls use the 'real' netmask not a wildcard mask where the bits are reversed.

FYI: It is also possible to create an access list and reference it in the NAT 0 group if there are several address blocks that need to bypass NAT/PAT.

The next step is to decide how the data should be protected. Since this changes the data to 'cypher text' (encrypted text) the settings are called a 'transform set.' A transform set for a Cisco PIX can contain up to three individual transforms, one AH and two ESPs. The transforms supported by the PIX are as follows:

ah-md5-hmac	AH with MD5 authentication
ah-sha-hmac	AH with SHA authentication
esp-md5-hmac	ESP with MD5 authentication
esp-sha-hmac	ESP with SHA authentication
esp-des	ESP encryption using 56 bit DES
esp-3des	ESP encryption using 168 bit triple DES

As I mentioned earlier, AH transforms are good if you are only concerned about people changing the data in transit and for authenticating the sender. If you want to encrypt the information be sure to use ESP.

The following command will configure a transform set called 'myset' with 3DES and MD5 authentication.

```
crypto ipsec transform-set myset esp-3des esp-md5-hmac
```

Crypto maps define what traffic will be protected and exactly how IPSec will protect it. The crypto map uses access lists to identify that traffic. Because I am configuring a remote access VPN I'll use something called a "dynamic access list." This type of access list is automatically created when the client connects then removed when they disconnect.

To create a dynamic crypto map called 'VPNusers' with a priority of 10 that uses the myset transform set enter the following command:

```
crypto dynamic-map VPNusers 10 set transform-set myset
```

Then I need to bind this dynamic crypto map to a static crypto map. I will call this crypto map 'newmap' and give it a priority of 100. I'll also set it to use IKE Phase 1 for peer authentication. This is done with the following command:

```
crypto map newmap 100 ipsec-isakmp dynamic VPNusers
```

Now I'll bind this crypto map to the outside interface.

```
crypto map newmap interface outside
```

The next set of configuration commands is for the Easy VPN remote clients. These commands will tell them what VPN IP address to use, the WINS/DNS servers and the group password for connecting to our Easy VPN server (i.e. PIX). We will also set our idle time to 1800 seconds and use the address pool we created earlier called 'clients2'. These commands are pretty much self explanatory. Our group name will be 'USERS' and the password will be 'mypasswd'.

> FYI: Both the VPN group name and password are case sensitive so remember that capitals count. When you configure the VPN client(s) make sure you enter the group name and password correctly.

```
vpngroup USERS address-pool clients2
vpngroup USERS dns-server server1
vpngroup USERS wins-server server2
vpngroup USERS idle-time 1800
vpngroup USERS password mypasswd
```

The last thing I need to do is to allow IPSec traffic to pass into the PIX from the outside. Allowing IPSec traffic is done with the following command:

```
sysopt connection permit-ipsec
```

Data Transfer

This is the phase where the tunnel is established and the data is actively going back and forth.

IPSec Tunnel Termination

When the users are finished this is the point that they disconnect the VPN client and the dynamic access list is removed.

Basic VPN Client Configuration

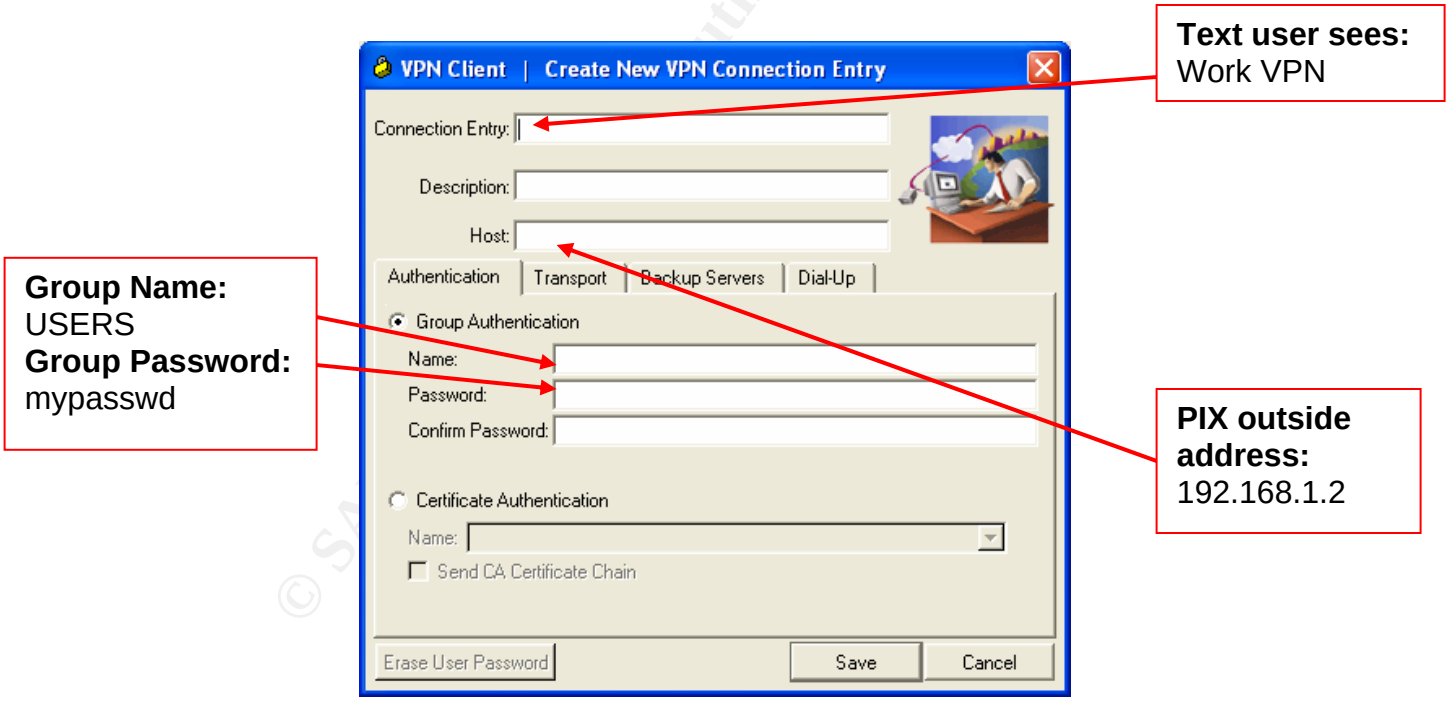
The next step is to install and configure the Cisco software VPN client. I'll install version 4.0.2 (b) on a Windows XP system. Cisco provides a good document on installing the client. This page also has links to other useful information ranging from understanding the user interface through managing digital certificates.

> FYI: If you are using a 3.0 or newer client you must be running PIX OS 6.0 or later on your firewall.

Please review the following link for the system requirements and installation procedures. It's an easy, straight forward process.

http://www.cisco.com/en/US/products/sw/secursw/ps2308/products_user_guide_chapter09186a008015ce7b.html

After finishing the software installation and rebooting it's time to start the client and click the "new" button to create a connection entry.



All the settings that have to be configured are on this screen. The default settings on the other tabs will work fine for our needs. The connection entry and description can be whatever you want. The text you type in the connection entry box will be what the user sees in the client window. This should be short and meaningful, especially if there are several different connections configured on the

client. I'll call mine 'Work VPN'. The host is the Public IP address of the VPN endpoint: in this case the outside PIX interface (192.168.1.2).

The Group Authentication section is critical. The group name is the name configured in the vpngroup section on the PIX. I used 'USERS'. The password is the group password configured in that section. I used 'mypasswd'. Remember that the group name and password are case sensitive so be careful when you type it in and let your users know that capitals count.

Click the 'Save' button for the settings to be saved on the client. To connect just double click the connection or highlight that connection entry and then click the connect button on the tool bar.

So far this has been pretty intuitive. The only thing that isn't immediately obvious is how to disconnect the client. When the client is open, before connecting, there is a lock in the system tray near the clock. That lock will appear unlocked or open.

When connected to the VPN with an established tunnel the VPN client window on the screen will disappear by default. The only visual indication that there is a connection to a VPN will be the lock in the locked, or closed, position.

While they are connected to the VPN users will only be able to access VPN resources and not the rest of the Internet with our configuration. Don't be surprised when new VPN users contact the help desk because they can't get to the Internet or some other resource. In my experience they tend to forget they are connected to a VPN early on but eventually they get used to it.

FYI: It is possible to allow VPN connections and Internet access at the same time. This is called 'split tunneling' and I don't recommend using it. Attackers have been known to use systems with split tunneling enabled to bypass firewalls and attack internal systems.

To disconnect from the VPN either right click on the lock and choose 'disconnect' or double click the lock and then click the disconnect button on the menu.

Simply by taking the defaults for the installation and entering minimal information the users can connect to the corporate network. If I later decide that I want to use SHA instead of MD5 for authentication I simply have to change the Easy VPN Server configuration on the PIX and all clients will use those settings on their next connection. Changing 1 or 1000 clients isn't a problem. I also know that my policy is consistently enforced.

At this point VPN access for users with a shared group name has been configured. Next I'll configure the authentication server and finally examine what needs to be modified on the PIX to allow for individual user authentication.

Basic Cisco Secure Access Control Server (CSACS) Configuration

Let's look at installing and configuring the Cisco Secure Access Control Server (CSACS). If you don't already own a copy of this software an evaluation copy is available from the Cisco website to allow you to test the features and functionality of the product before purchasing. You will need a Cisco Connection Online (CCO) login to download the evaluation. If you click the "download software image" link on the bottom of the following URL you will be directed to a page that will allow you to login. If you don't already have a CCO login there is a link to begin the process of registering for one. This page also provides links to a significant amount of documentation on the CSACS product and its features.

http://www.cisco.com/en/US/products/sw/secursw/ps2086/index.html

For this paper we are using the full featured, 90 day evaluation copy of the CSACS version 3.3.1 running on a Windows 2000 SP 4 server.

CSACS will provide Authentication, Authorization and Accounting services, often referred to as AAA (sometimes pronounced 'triple A') services. An AAA server is a database that is checked by network devices to confirm the identity of a user (Authentication), verify that the user is allowed to do what they are attempting (Authorization), and track what each user does (Accounting.)

There are two basic components that make this work: the AAA server (CSACS) and the AAA client (PIX). It is worth mentioning that some documentation refers to an AAA client as a Network Access Server (NAS.) Any device that the user connects through that is configured to redirect the user authentication process to an AAA server is a NAS or AAA client. I feel that the term AAA client is a better expression of what it does. Remember that an AAA client is not a user; it is the device the user connects through.

The PIX Firewall needs to be running version 4.0 or later for full AAA support. AAA clients can also be Cisco Routers or other devices that provide network access. Full TACACS+ and RADIUS AAA client support is available on Cisco IOS devices running version 11.1 or later.

TACACS+ and RADIUS are the two protocols that the Cisco PIX can use to communicate AAA information and provide AAA services. Let's look at RADIUS and TACACS+ and see why I feel TACACS+ is superior.

> FYI: In addition to TACACS+ and RADIUS Cisco Routers can also support Kerberos for authentication. Kerberos is outside the scope of this paper.

RADIUS stands for Remote Authentication Dial-In User Service. It was developed by the Internet Engineering Task Force (IETF) and is an open standard (RFC 2865 which obsoletes RFC 2138.) It is basically the universal standard for authentication. Almost all devices that perform this type of function support it although the implementation of some vendors may not be fully compatible with others. RADIUS has been around for a long, long time. It is a good choice when you have products from multiple vendors that need a central authentication system.

TACACS+ stands for Terminal Access Controller Access Control System Plus. It was developed by Cisco (RFC 1492.) It offers several advantages over RADIUS. Securing Cisco IOS Networks[2] has a nice comparison of these two protocols. Some of the more important differences are as follows:

TACACS+: Uses TCP (guarantees packet delivery)
RADIUS: Uses UDP (doesn't confirm packet delivery)

TACACS+: Supports more protocols
RADIUS: Won't support AppleTalk Remote Access (ARA), NetBEUI, Novell Asynchronous Services Interface (NASI) and X.25 PAD connections.

TACACS+: Encrypts the entire packet
RADIUS: Only encrypts passwords

TACACS+: Fully independent architecture for AAA
RADIUS: Authentication and authorization are combined.

Obviously there are several advantages to using TACACS+ but you need to be using Cisco equipment or equipment that is TACACS+ compatible. Since I am using all Cisco equipment this is the protocol I will be using. If I needed, or wanted, to use RADIUS the configuration would be almost identical. CSACS supports both protocols.

[2] Lammle, p.95.

FYI: Cisco has a nice document on their site that compares the advantages and disadvantages of these two protocols.

http://www.cisco.com/en/US/tech/tk583/tk547/technologies_tech_note09186a0080094e99.shtml

First let's look at how the AAA server (CSACS) and AAA client (PIX) communicate using TACACS+. The user attempts to connect to or through a device that is configured to use AAA authentication. That device, the AAA client, then sends a message to the AAA server that a user is trying to connect. The server then asks the client for the username. The client gets the username from the user and sends that information to the server. The server then asks for the password which the client also gets from the user and relays it to the server. After that the server compares the username and password to the information stored in the database and sends the client either a pass or a fail message. If the information matches an authorized user it passed and the user is successfully authenticated.

Now I need to install the CSACS software. I installed the Windows version on a Windows 2000 SP4 server. One thing I noticed was that the minimum hardware requirements displayed during the install and the ones in the installation guide from the website were slightly different. Since the webpage had more detailed information and slighter higher hardware requirements I'd prefer to be cautious and go with the website recommendations. Here is a link to those specs and the installation guide.

http://www.cisco.com/en/US/products/sw/secursw/ps2086/prod_installation_guide09186a0080238b18.html#wp980498

Now that I've confirmed that my server meets the minimum hardware requirements let's install the software. Don't forget that you will need to be logged in as an Administrator to install this software on the server.

Early in the install process we get a window that tells me to make sure the following items are complete before installing this software.

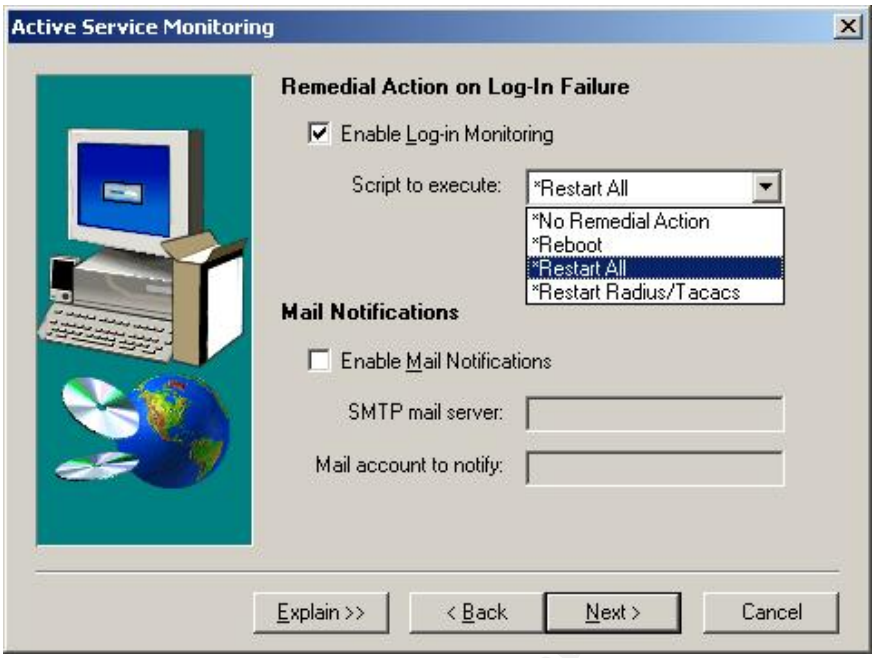
This screen's function is to make sure that both the users and the CSACS server can ping the PIX. Confirm that all IOS devices are running code that supports the AAA features and ensure that the web browser meets the minimum requirements for the administration console's web interface to work properly.

After I check off these boxes and click next I am asked to choose the installation folder. The default location is fine so click next once again.

The CSACS has the option of using its own database for user authentication or an external database like Active Directory. My configuration will use the local database.

Next the installer program asks which, if any, advanced options it should display. I won't need to use any of these options but you can check them if you like.

FYI: You can enable/disable access to these individual options after installation.

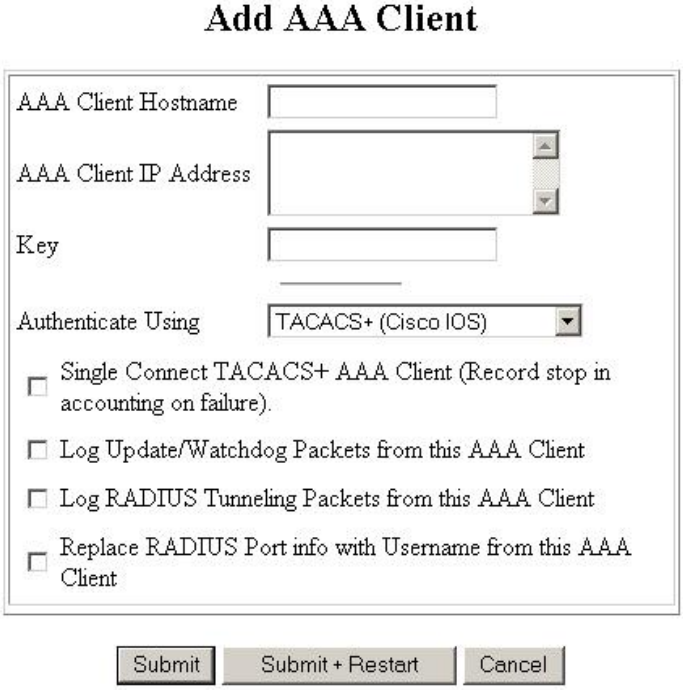


The next screen (shown above) gives me the option to have login attempts monitored and allows a script to be run in the event of a problem. For example, if the server stops allowing people to authenticate and login then it can run a script to reboot the system, restart the application, restart only the RADIUS/TACACS+ services or do nothing. It can also notify me of these events through email. If I wanted more than one person to be notified I can create mail groups on my mail server that will act as a distribution list to alert the proper Administrators.

After this step simply accept the defaults for the rest of the installation.

The next thing I need to do is to configure my AAA Server (CSACS) to work with my AAA client (PIX). First I need to launch the console. I do this by double clicking the “ACS Admin” shortcut on the desktop that was created during my install. There is a list of buttons on the left side. This is how I can navigate to different sections of the CSACS. The main part of the window will be where I do the actual configuration. To get into the window where I can add my AAA client devices I click the “Network Configuration” button on the left side. A window opens that has an area for me to enter my information and another large area with context sensitive help.

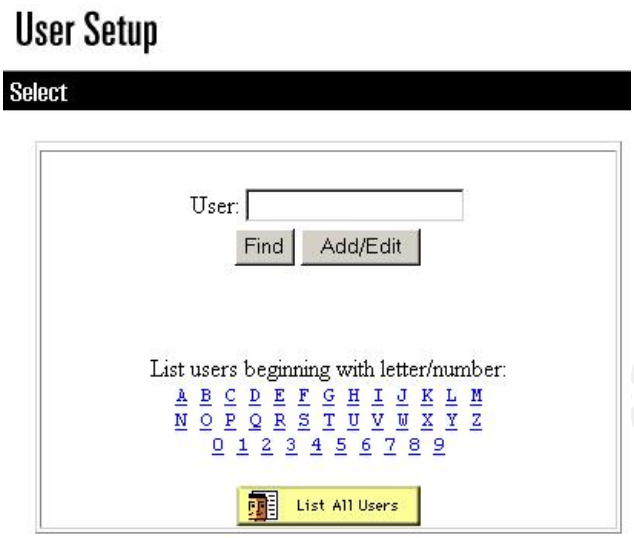
The top part of that window will have a section for AAA Clients. The list of clients will be blank but there will be a button immediately below that section called “Add Entry”. Click this button and the following window will appear.



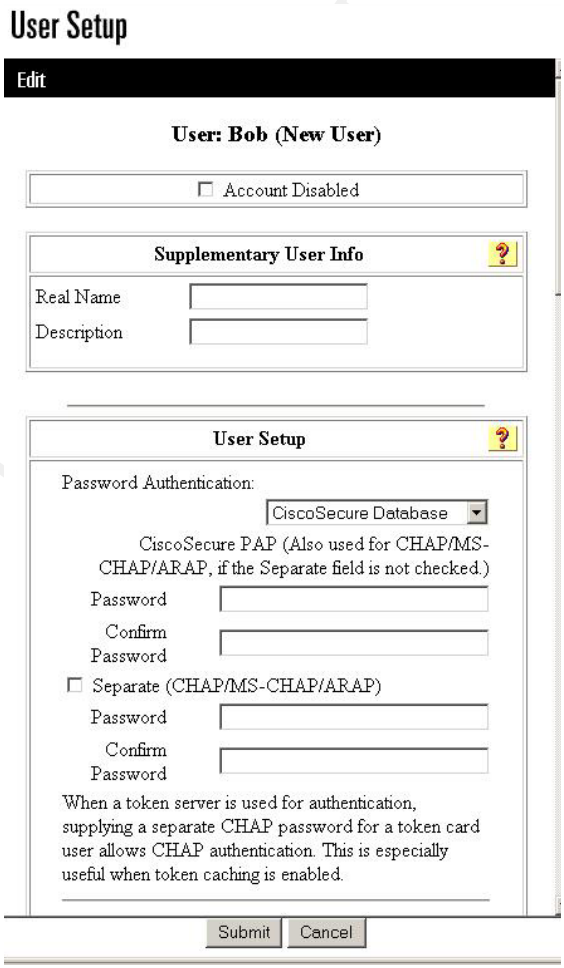
I only need to change the three fields on the top: “AAA Client Hostname”, “AAA Client IP Address” and “Key.” The client hostname is what I want to name this particular client. The IP address is the PIX’s IP address. Be aware that you can have multiple physical clients with a single AAA client entry on the CSACS. Depending on your needs and the number of AAA client devices in your environment this can be a big time saver. If you have more than one device per AAA client entry all devices will need to use the same protocol and key. Both the AAA Server and AAA Client(s) need to be using the same key in order for this to work properly. The key can be any alphanumeric keyword up to 127 characters without spaces (other special characters are okay) and it is case-sensitive. After entering the following information click the “Submit + Restart” button to save and immediately apply your changes. It will only restart the service not the server.

```
AAA Client Hostname:     PIX
AAA Client IP Address:   10.10.10.1
Key:                     mypass
```

Now I need to add the users. First I have to click the “User Setup” button on the top left of the navigation bar. That will open a window that will let me search for existing users to update/modify them or to add new users. The window looks like this:



To add a new user type the username you want to add then click the Add/Edit button. I'll add a user named Bob. When I type his name and click the Add/Edit button we'll see a screen like this one:



The Real Name and Description fields are self explanatory. Since I don't need different passwords for PAP, CHAP, etc I won't check the box to enable separate passwords. I can now enter the password for this user.

This is all we need to do to enable basic authentication on the CSACS for each user. The last step is to configure the PIX to use the CSACS for authentication.

Modifying the PIX to use CSACS for Xauth

Xauth is supported on PIX OS 5.2 or later. It allows individual user authentication between the IKE Phase One and IKE Phase Two stages. If the user is successfully authenticated then the PIX will establish the IKE Phase Two/IPSec SA. If authentication fails, the PIX will stop the creation of the tunnel and teardown the IKE Phase One connection as well. Here's what I need to do to use it.

First I need to ensure that TACACS+ is enabled as an AAA protocol. This is done by default. If you don't see the following command you must enter it.

```
aaa-server tacacs+ protocol tacacs+
```

The next step is to configure the PIX to use the AAA server. The PIX can have up to 16 independent groups of AAA servers allowing it to perform AAA functions differently based on the type of traffic (telnet, VPN, http, etc) or direction (inbound, outbound.)
Each group can have up to 16 individual AAA servers configured as members to allow for redundancy and failover. This allows for up to 256 TACACS+ and/or RADIUS servers, in total, between the two protocols.

Now I want to create an AAA group called "mytacacs" and have it use the 10.10.10.5 AAA server connected through the inside interface. I also want to use "mypass" as our secret key for communicating. The last setting for this is the timeout value which I will set to the default value of 5 seconds.

The timeout value can range from 5 to 30 seconds. When the AAA client tries to connect to the AAA server it will send a request and wait until the time out value expires. After the timeout expires it will send another request and wait until the timeout value expires again. It does this four times in total before attempting to contact the next server in the group. So with the default value of 5 seconds the AAA client will try for a total of 20 seconds to reach the first server in the group before attempting to contact the next server.

```
aaa-server mytacacs protocol tacacs+
aaa-server mytacacs (inside) host 10.10.10.5 mypass timeout 5
```

The last step is to tell the PIX to use Xauth for authentication of my VPN users. I also need to make sure it uses the correct AAA server group. To do this I need to add one command to my existing configuration.

```
crypto map newmap client authentication mytacacs
```

As you can see, enabling Xauth on my PIX was not very difficult. After only adding three additional commands I was able to authenticate my remote VPN users individually.

References

Chapman Jr., David and Fox, Andy. Cisco Secure PIX Firewalls. Indianapolis: Cisco Press, 2002.

Cisco Systems Inc. "About IPSec." November 5, 2002 http://www.cisco.com/univercd/cc/td/doc/product/iaabu/pix/pix_v52/ipsec/ipsec.htm (September 3, 2004)

Cisco Systems Inc. "CISCO EASY VPN." http://www.cisco.com/en/US/products/sw/secursw/ps5299/prod_brochure09186a00800a4b36.html (September 3, 2004)

Cisco Systems Inc. "How to Add AAA Authentication (Xauth) to PIX IPSec 5.2 and Later." Cisco PIX 500 Series Firewalls. May 4, 2004 http://www.cisco.com/en/US/products/hw/vpndevc/ps2030/products_tech_note09186a008010a206.shtml (September 3, 2004)

Cisco Systems Inc. "Installation Guide for Cisco Secure ACS for Windows Server Version 3.3." http://www.cisco.com/en/US/products/sw/secursw/ps2086/prod_installation_guide09186a0080238b18.html (September 3, 2004)

Cisco Systems Inc. "Installing the VPN Client." Cisco VPN Client. http://www.cisco.com/en/US/products/sw/secursw/ps2308/products_user_guide_chapter09186a008015ce7b.html (September 3, 2004)

Cisco Systems Inc. "Introduction." Cisco Secure Access Control Server for Windows. http://www.cisco.com/en/US/products/sw/secursw/ps2086/index.html (September 3, 2004)

Cisco Systems Inc. "Managing VPN Remote Access." Cisco PIX Firewall Software. http://www.cisco.com/en/US/products/sw/secursw/ps2120/products_configuration_guide_chapter09186a0080172787.html (September 3, 2004)

Cisco Systems Inc. "TACACS+ and RADIUS Comparison." May 4, 2004 http://www.cisco.com/en/US/tech/tk583/tk547/technologies_tech_note09186a0080094e99.shtml (September 3, 2004)

Lammle, Todd and Timm, Carl. Securing Cisco IOS Networks. Alameda: Sybex, 2003.

Mason, Andrew. Cisco Secure Virtual Private Networks. Indianapolis: Cisco Press, 2002.

Last Updated: May 27th, 2017

Upcoming Training

SANS Stockholm 2017	Stockholm, Sweden	May 29, 2017 - Jun 03, 2017	Live Event
SANS San Francisco Summer 2017	San Francisco, CA	Jun 05, 2017 - Jun 10, 2017	Live Event
SANS Houston 2017	Houston, TX	Jun 05, 2017 - Jun 10, 2017	Live Event
Security Operations Center Summit & Training	Washington, DC	Jun 05, 2017 - Jun 12, 2017	Live Event
Community SANS Ottawa SEC401	Ottawa, ON	Jun 05, 2017 - Jun 10, 2017	Community SANS
SANS Rocky Mountain 2017	Denver, CO	Jun 12, 2017 - Jun 17, 2017	Live Event
SANS Charlotte 2017	Charlotte, NC	Jun 12, 2017 - Jun 17, 2017	Live Event
SANS Rocky Mountain 2017 - SEC401: Security Essentials Bootcamp Style	Denver, CO	Jun 12, 2017 - Jun 17, 2017	vLive
Community SANS Portland SEC401	Portland, OR	Jun 12, 2017 - Jun 17, 2017	Community SANS
SANS Secure Europe 2017	Amsterdam, Netherlands	Jun 12, 2017 - Jun 20, 2017	Live Event
SANS Minneapolis 2017	Minneapolis, MN	Jun 19, 2017 - Jun 24, 2017	Live Event
SANS Columbia, MD 2017	Columbia, MD	Jun 26, 2017 - Jul 01, 2017	Live Event
SANS Cyber Defence Canberra 2017	Canberra, Australia	Jun 26, 2017 - Jul 08, 2017	Live Event
SANS Paris 2017	Paris, France	Jun 26, 2017 - Jul 01, 2017	Live Event
SANS London July 2017	London, United Kingdom	Jul 03, 2017 - Jul 08, 2017	Live Event
Cyber Defence Japan 2017	Tokyo, Japan	Jul 05, 2017 - Jul 15, 2017	Live Event
SANS Cyber Defence Singapore 2017	Singapore, Singapore	Jul 10, 2017 - Jul 15, 2017	Live Event
Community SANS Minneapolis SEC401	Minneapolis, MN	Jul 10, 2017 - Jul 15, 2017	Community SANS
SANS Los Angeles - Long Beach 2017	Long Beach, CA	Jul 10, 2017 - Jul 15, 2017	Live Event
Community SANS Phoenix SEC401	Phoenix, AZ	Jul 10, 2017 - Jul 15, 2017	Community SANS
SANS Munich Summer 2017	Munich, Germany	Jul 10, 2017 - Jul 15, 2017	Live Event
Mentor Session - SEC401	Macon, GA	Jul 12, 2017 - Aug 23, 2017	Mentor
Mentor Session - SEC401	Ventura, CA	Jul 12, 2017 - Sep 13, 2017	Mentor
Community SANS Atlanta SEC401	Atlanta, GA	Jul 17, 2017 - Jul 22, 2017	Community SANS
Community SANS Colorado Springs SEC401	Colorado Springs, CO	Jul 17, 2017 - Jul 22, 2017	Community SANS
SANSFIRE 2017	Washington, DC	Jul 22, 2017 - Jul 29, 2017	Live Event
SANSFIRE 2017 - SEC401: Security Essentials Bootcamp Style	Washington, DC	Jul 24, 2017 - Jul 29, 2017	vLive
Community SANS Charleston SEC401	Charleston, SC	Jul 24, 2017 - Jul 29, 2017	Community SANS
Community SANS Fort Lauderdale SEC401	Fort Lauderdale, FL	Jul 31, 2017 - Aug 05, 2017	Community SANS
SANS San Antonio 2017	San Antonio, TX	Aug 06, 2017 - Aug 11, 2017	Live Event
SANS Prague 2017	Prague, Czech Republic	Aug 07, 2017 - Aug 12, 2017	Live Event