alpha micro PRODUCTS **VAR *Tech Advisory***

VTB2002-04

October 28, 2002

Secure Your AMOS Telnet Sessions With SSH Encryption

Remote Connections Enjoy Increased Security

Dear Alpha Micro Dealer:

Would you like to increase the security of Telnet sessions? Want to increase protection against hackers beyond the usual password security? Are some of your customers demanding encryption of TCP/IP communications with the outside world? Need to meet the new HIPAA standards for medical records security? Don't want to hassle with setting up a VPN? One convenient solution to these requirements is to employ SSH encryption on your Telnet sessions.

SSH vs. SSL Encryption

In a previous VAR Tech Advisory (VTB2002-03), we discussed how to implement Secure Sockets Layer (SSL) encryption with the AMOS® web server. Secure Shell (SSH) encryption is a "cousin" of SSL. **Whereas SSL is typically used with HTTP and occasionally FTP servers, SSH is typically used with Telnet, remote control, and occasionally FTP applications. We will limit our discussion to implementing SSH in conjunction with Telnet.**

In the AMOS world, many of us have learned of SSH through the SSH client (TTSSH) that is built into the ZTERM terminal emulator. We understand that U.A. Systems is also preparing an SSH client to be introduced in a future version of AlphaLAN®.

The SSH protocol involves the use of a client and a server. The client resides on the PC hosting the terminal emulation software. The server resides on the opposite end of the wide area connection, on the same LAN as the AMOS Telnet server, as shown in Figure 1 below.

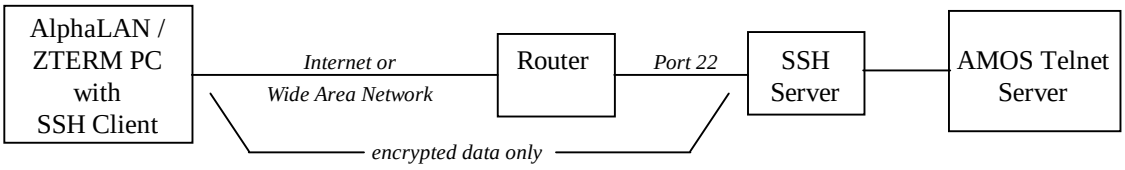

Figure 1: SSH Data Flow

Choosing an SSH Server

There are many SSH servers on the market, both commercial products and freeware. Each has their pros and cons. Some commercial products have an easy-to-use GUI setup. For our example, we will use a freeware product available for both Windows and Linux, OpenSSH. OpenSSH has the functionality we desire, however it does not have a GUI wizard for setup. You may download the Windows version at www.networksimplicity.com/openssh/, and the Linux version at www.openssh.com. Our discussion will employ the Windows version. Note that the OpenSSH server requires NT, 2000, or XP, whereas the OpenSSH client will also work on 9x and ME.

Installing the SSH Server

After downloading OpenSSH for Windows from the above link, follow the usual Windows installation process. Install both the Server and the Client. Then open a DOS prompt, and enter the following commands:

```
C:
cd Program Files
cd Networksimplicity
cd ssh
mkpasswd  -l  -u  "localuser"  >>  ..\etc\passwd
```

[The –l is a lower case L.]
["localuser" is the Windows User Name you will be logged on as on the local PC]
[Do this again for every User Name that will be permitted to log in to the SSH server.]

```
mkgroup  -l  >>  ..\etc\group
```

Edit the file `C:\Program Files\Networksimplicity\etc\group` with Notepad, and remove any duplications. For example, you might notice the bottom half of the file being the same as the top half.

Edit the file `C:\Program Files\Networksimplicity\ssh\sshd-config` with Wordpad (not Notepad). Change the line RSAAuthentication from "No" to "Yes". Change the Protocol line from "2" to "2,1". These changes enable the SSH server to recognize SSH1 protocol, which ZTERM uses.

In spite of what is stated in the OpenSSH documentation, generating authentication keys using ssh-keygen is an optional step that you may skip. We will be using Windows User Names as the authentication method instead of cryptographic keys.

Reboot the PC. Make sure that the OpenSSH service is started. It should be set to Automatic start.

The Puzzle of Port Forwarding

The trickiest aspect of setting up SSH is properly configuring Port Forwarding. The SSH client must be configured to pass the Telnet session “through” the SSH server at the other end of the WAN, and forward an unencrypted Telnet session onto the AMOS Telnet server.

The confusing aspect is that you must set up two sessions, one being an SSH session to the SSH server, the second being a conventional Telnet session that rides on top of the SSH session. Think of the SSH session as the track, and the Telnet session as the train riding on the track.

Configuring ZTERM’s SSH Client

This involves the following steps:

- Set up the SSH protocol
- Set up SSH Authentication
- Set up SSH Forwarding
- Entering the SSH password
- Making the first connection
- Enabling Auto Connect
- Saving the configuration
- Telnetting via the local SSH client

ZTERM has a built-in SSH client, TTSSH. We will use it to create the “track” connection to the SSH server. Set up a new connection for ZTERM where the protocol is SSH, not Telnet. Specify the IP address of the remote SSH server. Use port 22, the standard SSH port. The terminal emulation type does not matter; choose any one you like. Do not set it for “Auto Connect at Startup” for now.

Click on Settings, SSH Authentication. Click the button for “Use plain password to log in.” Enter the default User Name present on the SSH server PC which you will be using. Keep in mind that authentication is done against the Windows User Names on the SSH server PC, not on the PC running ZTERM. You might be logged on as “Mary” on the ZTERM PC, but might have a user name of “Mary Jones” on the SSH server PC and "Ms. Jones" on the AMOS system.

Click on Settings, SSH Forwarding. Click on Add, and set up one Local Port Forwarding entry for port 23 on the local PC to the port and IP address of the AMOS Telnet server. This can be a local network only IP such as 192.168.0.x, because presumably the SSH server and the AMOS Telnet server are on a LAN. You may substitute a port other than 23 if you are running a Telnet server on the local PC that also claims port 23.

Click on the Action, Connect, OK. You will be prompted for the password matching the Windows User Name you are using on the SSH server PC. If everything has been set up correctly, you will be presented with a pseudo-DOS prompt on the remote SSH server.

The first time you make a connection with ZTERM, it may tell you that the host to which you want to connect is not a known host. After you click OK the first time, you will not be asked this question in the future.

Once you establish that you are connecting to the SSH server successfully, you may enable Auto Connect at startup by checking the box under Configuration, Options, Automatic Connection.

Note that ZTERM has a bug wherein the TTSSH settings are not saved if the connection was not successful. Once you get an SSH connection working under ZTERM, be sure to close out ZTERM, and test the connection a second time to ensure that all of the TTSSH settings did, indeed, get saved. Specifically, the User Name, Port Forwarding, and Automatic Connection settings should all be saved, and you should immediately come up to the pseudo-DOS prompt of the SSH server.

At this point, have ZTERM connect to the SSH server, then minimize the screen. This SSH session will need to be idle in the background to provide the "track" on which your Telnet session will run.

Next, you will need to set up a second ZTERM connection icon for the Telnet session. It should be set up as a typical Telnet session with one key difference: You will be connecting to the local IP address of 127.0.0.1, at port 23. When you connect to "yourself," the TTSSH client running on the other ZTERM session will remotely Port Forward the Telnet session over the SSH-encrypted path, through the SSH server, and on to the AMOS Telnet server. And that is what we are after: a Telnet session that is encrypted while passing over the wide area connection.

Note that if you accidentally disconnect the SSH session before the Telnet session, the Telnet session will be automatically disconnected as well.

Configuring the Pre-SSH AlphaLAN or Other Terminal Emulators Without SSH

Soon, AlphaLAN promises to have SSH built into the product. Until that time, you can add SSH capabilities to AlphaLAN and any other terminal emulation software lacking built-in SSH via the use of the OpenSSH client program.

Create a .BAT file with the following commands:

```
echo on
cd  \
cd  Program Files
cd  Networksimplicity
cd  ssh
ssh  -l  "username"  -L  23:amosbox:port  sshserver
```

Where "username" is the Windows User Name to log on to on the SSH server, "amosbox" is the IP address of the AMOS Telnet server, "port" is the port of the Telnet server on the AMOS system, and "sshserver" is the IP address of the SSH server. You may use domain names in place of the IP addresses if you wish.

When you execute this batch file, the SSH client will attempt to make a connection to the SSH server. If it is successful, it will prompt you for the Windows Password that matches the Windows

User Name. Upon successful entry of the password, you will be presented with a pseudo-DOS prompt on the remote SSH server. Minimize this DOS box.

Start up AlphaLAN, and connect to the loopback port 127.0.0.1, port 23. The SSH client will listen for the incoming connection on port 23 on the local PC, and will remotely Port Forward it through the SSH server on to the port on the AMOS system where the Telnet server is running. You have thus achieved the goal of a Telnet session that is SSH-encrypted as it passes through the wide area connection.

When you are done using the Telnet session, disconnect AlphaLAN. You can then type “exit” at the pseudo-DOS prompt where the SSH client is running. It will exit out to your local DOS prompt. Note that if you try to exit the SSH client prior to disconnecting the Telnet session, the SSH client will hang until the Telnet session is closed.

Other Installation Considerations

If you are using a NAT or similar firewall, configure the router on the SSH server side to route port 22 (the SSH port) to the PC which will be running the SSH server. The Telnet traffic will be encrypted on top of the SSH session on this port.

In this example, we have used the Windows User Name and Password as the method of authentication. It is easy to understand and set up. SSH also allows other methods of authentication, such as the use of a “public key” file, which must be present on both the client and server PCs. You can read more about alternative authentication methods in the documentation for OpenSSH in C:\Program Files\Networksimplicity\docs.

Once an SSH session from the local PC to the SSH server is established, you can also “ride” other TCP/IP protocols on top of the session. For example, you could set up encrypted FTP or POP3 transmissions. Simply set up additional Local Port Forwarding entries in the SSH client on the local PC to forward those ports to the appropriate ports on the FTP or POP3 server. Then have the FTP or POP3 clients connect to IP address 127.0.0.1, so that the SSH client can pick up and encrypt the transmission.

Wish List

On the AMOS side of the equation, our goal is to add a native SSH server to AlphaTCP down the road, eliminating the need for a SSH server running on a PC.

On the AlphaLAN/ZTERM side, it would be nice if there were a “composite” SSH client, which would establish the underlying SSH connection and the second tier Telnet session in one swoop. Setting up dual sessions manually is somewhat inconvenient .

Contact Us For Further Assistance

As with SSL, setting up SSH can get complicated. Don’t hesitate to contact our Technical Support Staff for assistance in configuring SSH for your specific requirements. Your comments and suggestions are welcomed

Lastly, while VPN also provides additional security, it is a more complex and costly option. This is why we believe that with users increasingly asking for advanced security, SSH is clearly the best solution for your customers!