Description of SSHBRIDGE

SSHBRIDGE is an auxiliary program that allows your Windows applications to provide secure communications SSHv2 of extremely simple way.

Today is very convenient to leave the network communication techniques based on plain text, such as TELNET protocol, for encrypted communications using the standard SSHv2. On the other hand, this new protocol is very complex to implement and provide this new functionality to your applications already developed is a job really hard to do.

SSHBRIDGE offers the possibility of using this new communication system without changing your programs. It runs as an auxiliary program that serves as a bridge TCP-SSH. If launched with the parameter "/silent", runs hidden and can not be handled by the user from the Windows programs bar. Otherwise displays a window showing the TCP and SSH listened ports and the number of active connections for each of them.

To better understand the mechanics of SSHBRIDGE below are described three different scenarios. In all cases your application and SSHBRIDGE reside on the same machine to prevent the insecure communication between the two will be exposed in the network:

- **Client application.** The goal is to let your client application to access a remote SSH server. It's necessary to define in the configuration file what unsafe port (TCP) will be used for the bridging function TCP -> SSH. SSHBRIDGE acts as a server for that TCP port. In the configuration you can define the parameters relative to the remote server for each local TCP port: remote IP, remote port (SSH), login and password. You can configure as many local TCP ports as necessary. When SSHBRIDGE accepts a connection on any local TCP port, it performs an SSH connection to the remote SSH server doing a bidirectional bridge between the two connections. The parameters for the remote server that are not in the configuration will be requested to the client interactively:

  ```
  SSH Host...:
  SSH Port...:
  Login......:
  Password...:
  ```

- **Server application.** Now we want a remote SSH client can communicate with your server application that listens an unsafe TCP port. For this purpose, SSHBRIDGE acts as an SSH server. In the configuration there is also a list of SSH listening ports indicating which local TCP ports must do the bridges SSH -> TCP. When accepting an SSH connection from a remote client, SSHBRIDGE makes a TCP connection to the local port obtained in the configuration, which is the one your application listens, making the two-way bridge between them. Keep in mind that the process of establishing an SSH connection leads inevitably implicit the authentication of the client with username and password. SSHBRIDGE requires this two facts as SSH server but always accepts the connection without to do any checking. It's

responsibility of your server application to accept, or not, the authentication data. For this reason, SSHBRIDGE sends this information to the server application after establishing the TCP connection and before to make the bidirectional bridge. The format of this data is: name+\r+password+\r (\r is the symbol for Carriage Return, ASCII 13). If your application does not require any authentication you can disable this process by configuration.

- **SSH tunnel.** The job is to establish secure communication tunnels (SSH) for applications that are not safe. In this case we must have one SSHBRIDGE at each end of the communication for to redirect safely as many ports as desired. In the client application side is used the TCP -> SSH functionality and SSH -> TCP in the other end (server). In example: the machine A has a pop3 mail client and we want to reach the server in machine B safely. The SSHBRIDGE in machine A is listening the TCP port 110. It forwards the traffic to the SSHBRIDGE of the machine B, by an SSH connection, and then the original traffic is delivered to the server application. Notice that in this example we must disable the authentication sending to the server application because this process is done by the pop3 protocol, not by the connection procedure.

SSHBRIDGE not allow the execution of more than one instance simultaneously. If you launch the execution of the program, and there is already a running instance, the first closes automatically running the new operating. Your applications can modify the configuration of SSHBRIDGE and an easy way to apply the changes is to launch the execution of the program again. To close definitively the program you must call it whit the parameter “/close”. This is especially useful if you use the hidden execution mode. It is also possible terminate the process from the Windows Task Manager.

Configuration of SSHBRIDGE

As explained in the scenarios described, we must make certain settings for the proper functioning of the program. All settings are defined in the file SSHBRIDGE.INI which must reside in the operating system directory (usually C:\WINDOWS), or in the directory where you installed the program. Notice that there is not objection that the applications do changes in this configuration file without human intervention.

The configuration has three sections:

- TCP table. List of TCP ports that SSHBRIDGE listens, with the respective parameters of the remote SSH servers. They are the ports that your client applications use to communicate with SSHBRIDGE. TCP -> SSH direction.
- SSH table. List of SSH ports that SSHBRIDGE listens to allow remote SSH clients can talk with your TCP server applications. SSH -> TCP direction.
- General configuration

Below is an example of the configuration file with comments explaining the various parameters. Notice that sometimes leave empty fields in the TCP table (“,,”) to ask them interactively to the client application.

```
[TCPTABLE]
; tcp_port=ssh_ip,ssh_port,login,password

; all defined in the configuration
23=192.168.1.150,22,user1,puser1
; the client will send the authentication data
1023=192.168.1.151,22,,
; the client will send all fields interactively
1024=,,,

[SSHTABLE]
; ssh_port=tcp_port,¿send auth to server? (0=not, 1=yes)

; with authentication
22=23,1
; without authentication
1022=23,0

[GENERAL]
; hidden execution mode (0=not, 1=yes)
; activated also if launched with the paramenter “/silent”
SilentMode=0

; time in seconds to write data in the interactive way
LoginTime=30
```

License of use

This program is provided "as is" and the author does not accept any responsibility that may result from its use, or for the proper operation of it. You are completely free to use it privately or commercially, and have available as long as you want for his evaluation. Hidden execution mode is not operational in unregistered copies and in the window of the program is indicated that fact. To use this mode of execution, hiding to the end user the existence of this auxiliary program, you should contact the author of the program by sending an email to: info@ingelek.com and make a donation of 30 € (payment via Paypal .) Then you will receive a file encrypted with your name that allows the hidden execution mode and removes the “Unregistered copy” message in the program window.