Utility name:
TCPRELAY.EXE version 1.0.0

Purpose:
Marshals TCP traffic between clients and a server.

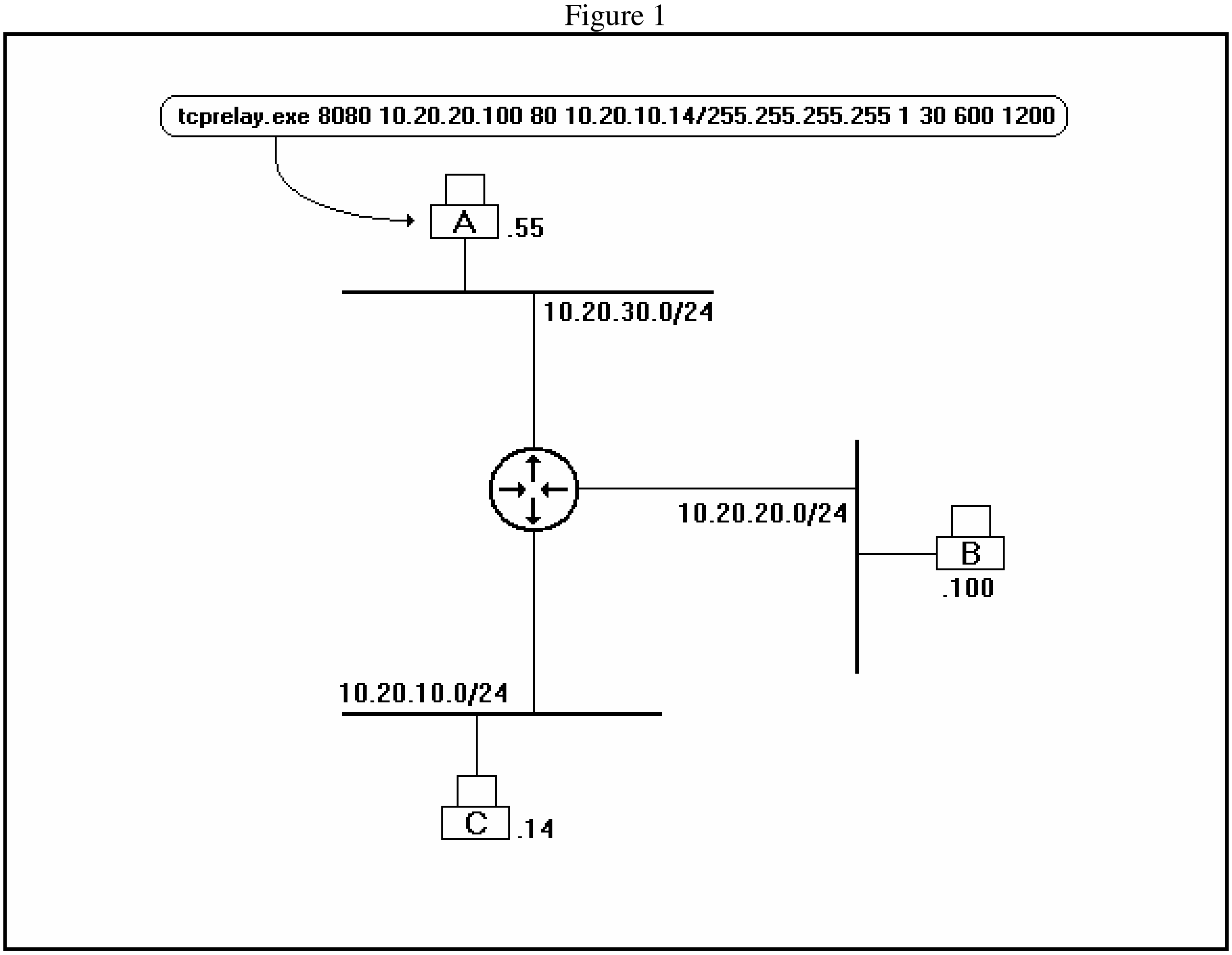
Author:
Bill Chaison, BC.Soft@yahoo.com, http://chaison.freewebpage.org

Development environment:
MS Visual C++ 6.0, console application, Winsock 2, multithreaded, 802 lines of code.

License and terms of use:
This program is provided as freeware. It may be used by any individual or organization for any lawful purpose free of charge. This program may not be resold, nor may it be bundled or repackaged with any application or compilation that will be exchanged for profit. This program does not contain malicious code or surreptitious features. This program is made available as-is and users of it are not entitled to product support. The author will not be held liable for any damages resulting from the use of this program.

Example applications:
Dynamic / ephemeral gateway, host-based proxy, network traffic path exception, etc.

Figure 1



```
tcprelay.exe 8080 10.20.20.100 80 10.20.10.14/255.255.255.255 1 30 600 1200
```

The scenario in figure 1 illustrates how host (C), which does not have direct access to host (B), can access the web server on host (B) through host (A). The parameters used with tcprelay.exe in this example indicate that host (A) will listen on TCP port 8080 and relay all of the data it receives on that port to host (B) over destination TCP port 80. The ACL used by tcprelay.exe in this example restricts access to just host (C). The example application in figure 1 also indicates that incomplete TCP connection attempts to host (B) will be dropped after 1 second, completed TCP sessions to host (B) that have been idle for more than 30 seconds will also be dropped, and tcprelay.exe will terminate itself after 10 minutes from the time it was launched. Some additional considerations – if host (B) must be called by name as "http://www.mysite.net" then host (C) may need a "hosts" file entry or some other name resolution provision to ensure that "www.mysite.net" resolves to the address for host (A) instead of the address for host (B). Host (C) will also need to be sure to call the resource as "http://www.mysite.net:8080". This utility does not provide transparent forwarding. All traffic marshaled between host (C) and host (B) by host (A) will use the IP address of host (A). For instance, the HTTP logs of host (B) will show the IP address of host (A) for web requests made from host (C) that are relayed through host (A).

To get additional help with the command line arguments, open a shell and execute the program without parameters.

Messages and output:

All error and informational messages are logged to STDOUT.

1. "Error: This system does not provide a high resolution timer."

   The program cannot continue because a 64-bit high-resolution timer is not available on your computer.

2. "Error: <fwd port> must be between 1 and 65535 inclusive."

   The program cannot continue because you supplied an invalid port number on which to listen.

3. "Error: Invalid IP address specified for <trg host>."

   The program cannot continue because you supplied an invalid IP address for the host to which the relayer will forward data.

4. "Error: <trg port> must be between 1 and 65535 inclusive."

   The program cannot continue because you supplied an invalid target port on the destination host to which the relayer will forward data.

5. "Error: <ACL> needs to be keyword ANY, or dotted decimal IP/mask notation."

   The program cannot continue because the ACL supplied was inappropriate. The ACL may be either the keyword ANY, which means that all requesters can relay through the host, or an IP/mask filter. Only one entry is supported. For example, 192.168.100.30/255.255.255.255 for the ACL would allow the relayer to accept requests for a single machine, 192.168.100.30, whereas a filter of 192.168.100.0/24 would allow the relayer to accept requests from 192.168.100.0 through 192.168.100.255 inclusive.

6. "Error: <c t/o> must be between 1 and 3 inclusive."

The program cannot continue because you supplied a TCP connect timeout that was not within the specified range. The TCP connection timeout pertains to the maximum time in seconds that the relayer will wait to establish a connection to the target host once a valid requester has made the attempt to do so. If this timeout is exceeded then the connection attempt is aborted.

7. "Error: <i t/o> must be between 1 and 1800 inclusive."

   The program cannot continue because you supplied a session idle timeout that was not within the specified range. The idle timeout pertains to the maximum number of seconds that an established relay thread may be idle (not sending or receiving data) before it is dropped.

8. "Error: <buf> must be between 100 and 4000 inclusive."

   The program cannot continue because you supplied a packet transfer buffer size (in bytes) that was not within the specified range.

9. "Error: WinSock initialization failure."

   The program cannot continue because it encountered a problem initializing the WinSock subsystem.

10. "Error: WinSock version must be at least 2.0."

    The program cannot continue because it encountered a problem with the version of the WinSock subsystem.

11. "Error: Listener socket creation failure."

    The program cannot continue because it could not allocate a listening socket.

12. "Error: Could not bind listener port to socket."

    The program cannot continue because it could not assign the port you specified in <fwd port> to the listening socket.

13. "Error: Listener failed to initialize."

    The program cannot continue because it could not begin the process of receiving incoming connections from requesters.

14. "Error: Unable to start the maintenance thread."

    The program cannot continue because it could not launch the primary worker thread that manages the session pool, relay threads, and process lifetime.

15. "The relayer is now active. Data received on TCP port <fwd port> will be forwarded to <trg host> port <trg port>..."

    All initialization has completed successfully and the relayer is ready to marshal sessions.

16. "Connection attempt made from <rqst IP addr>."

A connection attempt was received from source IP address <rqst IP addr> and will subsequently be checked against the ACL.

17. "Connection from <rqst IP addr> rejected due to ACL mismatch."

    A request from <rqst IP addr> was discarded because it did not pass the filter specified in the ACL.

18. "Connection from <rqst IP addr> was accepted. Allocating a session handler."

    The request from <rqst IP addr> passed the ACL filter rule and will be processed further.

19. "Could not allocate a socket to target <trg host>. Request from <rqst IP addr> aborted."

    The valid request from <rqst IP addr> could not be processed further because the relayer could not allocate a socket to use in connecting to the target server.

20. "Connection from <rqst IP addr> was dropped because the session handler thread failed to start."

    The thread that initiates the connection to the target in behalf of a valid requester failed to initialize.

21. "Session pool exhausted. Connection from <rqst IP addr> was dropped."

    The relayer pool limit of 500 has been reached and no further sessions may be established until some are freed.

22. "Error number <err num> was returned during requester connection attempt."

    The WinSock specific error number is returned in <err num> for a failed attempt to accept a connection from a requester.

23. "Absolute timeout expired. This process will now terminate."

    The time in seconds specified by <a t/o> has elapsed and the relayer will immediately terminate itself.

24. "Session handler serial number <ser num> connecting <rqst IP addr> to <trg host> exceeded the idle timeout and has been terminated."

    An established session has been inactive past the <i t/o> parameter and was discarded. Serial numbers referenced by <ser num> are non-zero and can be used to correlate session events across log messages.

25. "Session handler serial number <ser num> connecting <rqst IP addr> to <trg host> has been terminated due to an expired target connection attempt."

    An unestablished connection to the target has been inactive past the <c t/o> parameter and was discarded. Serial numbers referenced by <ser num> are non-zero and can be used to correlate session events across log messages.

26. “Session handler serial number <ser num> was allocated for request from <rqst IP addr>. Connecting to target <trg host>.”

    A session handler was successfully created to handle a new request. A subsequent connection attempt to the target server will be made. Serial numbers referenced by <ser num> are non-zero and can be used to correlate session events across log messages.

27. “The target refused the connection attempt for session handler serial number <ser num>.”

    The relayer tried to connect to the target but it declined to accept. Serial numbers referenced by <ser num> are non-zero and can be used to correlate session events across log messages.

28. “Session handler serial number <ser num> made a successful connection to the target. Starting the relay threads.”

    The target accepted the connection from the relayer for a new request and data forwarding threads will subsequently be launched. Serial numbers referenced by <ser num> are non-zero and can be used to correlate session events across log messages.

29. “Session handler serial number <ser num> could not start the requester relay thread.”

    The relayer encountered a problem launching the thread that forwards data from the requester to the target. Serial numbers referenced by <ser num> are non-zero and can be used to correlate session events across log messages.

30. “Session handler serial number <ser num> could not start the target relay thread.”

    The relayer encountered a problem launching the thread that forwards data from the target to the requester. Serial numbers referenced by <ser num> are non-zero and can be used to correlate session events across log messages.

31. “Session handler serial number <ser num> successfully started both relay threads.”

    Both unidirectional data relay threads associated with the stated session handler have been successfully created. Serial numbers referenced by <ser num> are non-zero and can be used to correlate session events across log messages.

32. “Session handler serial number <ser num> failed to allocate a requester transfer buffer.”

    A data transfer buffer of size <buf> bytes could not be allocated for the thread that sends data from the requester to the target. Serial numbers referenced by <ser num> are non-zero and can be used to correlate session events across log messages.

33. “Session handler serial number <ser num> requester-to-target channel is now active.”

    The thread that marshals data from the requester to the target has started successfully. Serial numbers referenced by <ser num> are non-zero and can be used to correlate session events across log messages.

34. “Session handler serial number <ser num> terminated due to requester read closure.”

The requester closed the connection referenced by <ser num> and the relayer has freed the corresponding item from the session pool. Serial numbers referenced by <ser num> are non-zero and can be used to correlate session events across log messages.

35. “Session handler serial number <ser num> terminated due to target write closure.”

    The target closed the connection referenced by <ser num> and the relayer has freed the corresponding item from the session pool. Serial numbers referenced by <ser num> are non-zero and can be used to correlate session events across log messages.

36. “Session handler serial number <ser num> failed to allocate a target transfer buffer.”

    A data transfer buffer of size <buf> bytes could not be allocated for the thread that sends data from the target to requester. Serial numbers referenced by <ser num> are non-zero and can be used to correlate session events across log messages.

37. “Session handler serial number <ser num> target-to-requester channel is now active.”

    The thread that marshals data from the target to the requester has started successfully. Serial numbers referenced by <ser num> are non-zero and can be used to correlate session events across log messages.

38. “Session handler serial number <ser num> terminated due to target read closure.”

    The target closed the connection referenced by <ser num> and the relayer has freed the corresponding item from the session pool. Serial numbers referenced by <ser num> are non-zero and can be used to correlate session events across log messages.

39. “Session handler serial number <ser num> terminated due to requester write closure.”

    The requester closed the connection referenced by <ser num> and the relayer has freed the corresponding item from the session pool. Serial numbers referenced by <ser num> are non-zero and can be used to correlate session events across log messages.