AFT Working Group
draft-ietf-aft-socks-v6-req-00.txt
Expires six months from -->

Marc VanHeyningen
Aventail Corp.
September 1, 1999

SOCKS successor requirements

Abstract

The SOCKS protocol version 5 has been deployed and seen use as a mechanism for authenticated firewall traversal. Experience with the use of this protocol and its limitations has led to the desire for a new firewall traversal protocol; we tentatively name this new protocol SOCKS version 6.

1. Introduction

SOCKS5 has enjoyed wide use in a variety of network environments as a protocol for traversing trust boundaries in heterogenous network environments. Its support for various authentication methods, specification of destinations by name rather than by IP address, and UDP proxy support have provided much benefit; however, there are new

features required, and the existing protocol was not designed to be sufficiently extensible such that an easy retrofit is possible.

Proposed here is a set of new top-level requirements for the protocol as a whole, along with a set of specific new functionality desired in this new version. As a base requirement, SOCKS v6 should be able to do everything currently done by compliant SOCKS v5 implementations.

2. General protocol features

Experience with SOCKS5 has indicated some fundamental aspects of the protocol which do not provide the level of flexibilty desired for wide use and enhancement. Thus, the successor should minimally include:

- Major and minor version numbers, to allow for revisions which do not break backward compatibility
- A general mechanism for negotiating the support of new protocol features.
- Authentication methods (and, if possible, the authentication framework) should leverage existing standards rather than re-invent them.
- A "control channel" may exist which allows multiple proxy operations to be conducted without incurring the overhead of re-authentication. This control flow should persist throughout the lifetime of the connection(s).

3. TCP-BIND features

The BIND command as defined in SOCKS5 is designed primarily for cases in which a server must make a "back-connect" to a client, as is the case in FTP. For this purpose the command as defined is sufficient; however, there are protocols which require multiple back-connects to a single listening address/port, and some require a specific port be used when accepting this connection.

The TCP BIND functionality shall include:

- The ability to support multiple connections, not just one, to the proxy's listening port
- The ability of the client to request a specific port be used by the server when listening on its behalf

4. UDP-BIND features

UDP was a new feature for SOCKS v5, and the initial support was very

limited in its capabilities. The model envisioned was that of applications like archie; a client sending data and a response being received. Many UDP applications have different requirements, such as receiving UDP data without sending any, using a specific port, and requiring IP address information.

The UDP BIND functionality shall include:

- The ability to establish the connection and receive address information about the proxy via a reliable channel
- The ability to send or receive UDP first
- The ability for the client to control the port used on its behalf
- Support for sending and receiving multicast UDP traffic, in a multicast or non-multicast environment.
- Support for tunneling UDP inside a reliable channel, at a performance penalty, if needed.

5. References

[RFC 1928] Leech, M., Ganis, M., Lee, Y., Kuris, R. Koblas, D., & Jones, L., "SOCKS Protocol V5," April 1996.

Author's Address

Marc VanHeyningen
Aventail Corporation
808 Howell Streeet; Suite 200
Seattle, WA 98101

Phone: +1 (206) 215-1111
Email: marcvh@aventail.com