[JET-PROTO]:
JET Relay Protocol

Revision History

Revision summary			
Author	**Date**	**Revision history**	**Comments**
Marc-André Moreau	10/25/2018	0.1	Initial draft

Contents

1 Introduction

This document specifies the JET Relay Protocol.

1.1 Glossary

The following terms are specific to this document:

MAY, SHOULD, MUST, SHOULD NOT, MUST NOT: These terms (in all caps) are used as described in [RFC2119]. All statements of optional behavior use either MAY, SHOULD, or SHOULD NOT.

1.2 References

1.2.1 Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997, http://www.ietf.org/rfc/rfc2119.txt.

1.3 Overview

The JET protocol bears some similarities with the SOCKS proxy protocol and the routing token packets often used in remote desktop connections for load balancing and session selection. However, all of these protocols make connections in a forward manner: the client connects to the proxy, then the proxy connects to the server and then relays the traffic. The JET protocol is designed to relay TCP traffic between a TCP client and server using only outgoing TCP connections.

1.4 Prerequisites/Preconditions

The JET protocol requires a TCP transport.

1.5 Applicability Statement

The JET protocol is suitable for simple, efficient relaying of TCP protocols.

2 Messages

2.1 Transport

The JET protocol is designed to provide a simple, efficient way to relay TCP traffic between two nodes that can only perform outgoing TCP connections to the same server, using a rendezvous connection style.

2.2 Message Syntax

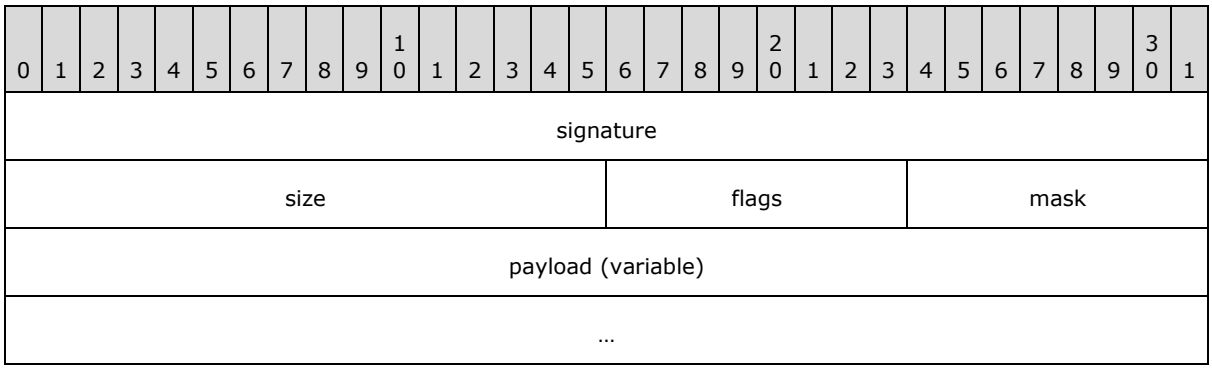
The following sections specify the JET protocol message syntax. All fields defined in this document use big endian byte ordering and are byte-aligned to their sizes (a field of 4 bytes starts at an offset that is a multiple of 4).

2.2.1 Protocol Messages

All JET protocol messages are contained within a JET_PACKET structure.

2.2.1.1 JET_PACKET

The JET_PACKET structure contains a routing token used to associate two indirect TCP connections.



signature (4 bytes): The packet signature. MUST be set to { 0x4A, 0x45, 0x54, 0x00 } ("JET").

size (2 bytes): The total size of the packet, including headers. The minimum size is 8.

flags (1 byte): This field is reserved for future use and MUST be set to zero.

mask (1 byte): This field contains a one-byte mask that MUST be applied to the payload.

payload (variable): This field contains a masked HTTP request or response using the value from the mask field.

2.2.2 Protocol Examples

Here is a sample TCP connection established using the JET protocol. To avoid interference from proxies, an 8-byte binary header is used to encapsulate HTTP requests and responses. The payload is masked using the value from the mask field.

2.2.2.1 TCP Server Accept

>> Request:

GET / HTTP/1.1

Host: jet.wayk.net

Connection: Keep-Alive

Jet-Method: Accept

Jet-Version: 1

<< Response:

HTTP/1.1 200 OK

Jet-Association: e6ec698c-5793-4c63-af79-bd644ccf022f

Jet-Instance: 101.jet.wayk.net:443

Jet-Version: 1

2.2.2.2 TCP Client Connect

>> Request:

GET / HTTP/1.1

Host: 101.jet.wayk.net

Connection: Keep-Alive

Jet-Method: Connect

Jet-Association: e6ec698c-5793-4c63-af79-bd644ccf022f

Jet-Version: 1

<< Response:

HTTP/1.1 200 OK

Jet-Version: 1

2.2.2.3 Connection Sequence