

Spider keylogger network protocol

Christopher Steel
Software Engineer

<https://github.com/Christopher-Steel>

Tuesday, October 14th 2014

1 Handshake

Client: protocol version

Client: identifier

Server: response

protocol version 4 bytes, specifies protocol version to check for compatibility with the server. Each byte represents a version level (Release Module Feature Bugfix)

identifier byte array, terminated by a null byte. Used by the server to group log by client.

response a Short response, defined below

Upon connection to the server, the client must follow this handshake until receipt of the server's response. Failure to do so will result in ignored commands or rejection of the client's connection.

2 Keystroke

Client: 1

Client: timestamp

Client: size

Client: text

1 1 byte, this is the command identifier that defines what the rest of the packet will be

timestamp 8 bytes, this is used to represent the time at which the Keystroke happened. For a Keystroke packet containing multiple keystrokes, the time of the last keystroke is used.

size 2 bytes, the size of text

text byte array, utf-8 encoded string

The client can send keystrokes separately whenever it catches them or it can send text blocks. It may construct text blocks based on the time that passes between keystrokes to decide if they belong to the same block. Blocks would be sent when sufficient time passes after the insertion of the last keystroke. Each block received by the server is stored separately.

3 Mouse click

Client: 2

Client: timestamp

Client: button

Client: position

2 1 byte, this is the packet identifier that defines what the rest of the packet will be

timestamp 8 bytes, this is used to represent the time at which the Keystroke happened. For a Keystroke packet containing multiple keystrokes, the time of the last keystroke is used.

button 1 byte, 0 = left click, 1 = middle click, 2 = right click

position 4 bytes, 2 for position X, and the other 2 for position Y

Mouse clicks are sent whenever they are caught.

4 Command

Server: 10

Server: length

Server: command

Client: data

10 1 byte, this is the packet identifier that defines what the rest of the packet will be

length 2 bytes, length of command, in bytes

command byte array, encoded ASCII string interpreted by client

data a Command response, defined below

5 Short response

Server: 11

Server: success

11 1 byte, this is the packet identifier that defines what the rest of the packet will be

success 1 byte, 1 = success, 0 = failure

6 Command response

Client: 3

Client: success

Client: size

Client: data

3 1 byte, this is the packet identifier that defines what the rest of the packet will be

success 1 byte, 1 = success, 0 = failure

size 4 bytes, size of data

data byte array, various possibilities of data depending on the command that caused the response