Spider Inter Communication RFC

BRUEL Jonathan
BONGOLO-BETO Berdrigue
LIANI Marwan
IACONA Alexandre
FRERE Lucas
LEGENDRE Adrien
HYRONDELLE Mathias
KONOVODOFF Kostas

Septembre 2017

Inter Communication Client - Server for Spider Project

1. Introduction

This document describes the protocol between the Keyloger client and the Spider server communication. The clients and servers use a TCP network.

2. Data Definition

The package is composed by two elements. The first one is the header (8 bytes). The header is composed by the request id (4 bytes) and the data size of the package (4 bytes).

The second one is the data.

The data is defined with a JSON norm, encrypted (base64) and start at the 9th byte.

This is the package's template.

```
{
        FROM:"",
        TO:"",
        TIMESTAMP:"",
        [...]
}
```

The is an example with the CLICK_ACTIVITY_REQ.

The encrypted one:

RyA+EuWGchBkKmuNEkgIJg5AwDqphqeKSklYf/SHoGDgkLTLG7BT6pwZo+V35ackbFpVP1X7GdrceyNDDyf9Bw

The decrypted one:

```
{
        FROM:"client_01",
        TO:"spider_server01",
        TIMESTAMP:"1506346693",
        CLICK_TYPE:”0”,
        CLICK_COORDINATES_X:”23”,
        CLICK_COORDINATES_Y:”87”,
        PROCESS_INFO:”firefox”
}
```

3. Request Definition

This is the list of all the requests between the client and the server

AUTH_REQ

Request which authenticates the client to the server.
The client sends his public key to the server to be allowed to communicate with it.
This will secure all the future communications.

Variables:

- PUB_KEY, Type: RSA

AUTH_RESP

Response from the sever to the client which informs if the authentication is done with success or failure.

If the connexion is done with success the server registers the client’s public key and send his own public key too.

Variables:

- IS_AUTH, Type: bool
- PUB_KEY, Type: RSA

CLICK_ACTIVITY_REQ

The request sends all information about the click of a client to a server.
This request sends the click type (Right, left or middle) and the position.

Variables:

- CLICK_TYPE, Type: enum (int)
- CLICK_X, Type: int
- CLICK_Y, Type: int
- PROCESS_INFO, Type: string

KEY_INFO_REQ

The KEY_INFO_REQ is a request that sends all the keys get by the client.

Variables:

- KEYS, Type: string
- PROCESS_INFO, Type: string

PING_REQ

Test the availability of the receiver.

PING_RESP

Give a positive answer to the sender ping test.

NOTICE_RECEIPT_REQ

Tells the sender that the receiver receives with success his request.

Variables:

- REQ_TIMESTAMP: unsigned int
- REQ_ID: enum (int)

ERROR_REQ

Tells the sender that an error has been detected with his request.
The sender must resend his request.

Variable:

1. ERROR_TYPE, Type: enum (int)
2. MESSAGE, Type: string

DISCONNECT_REQ

Tells the receiver that the sender is disconnect.

4. Enum definition

REQ_ID:

1. AUTH_REQ
2. AUTH_RESP
3. CLICK_ACTIVITY_REQ
4. KEY_INFO_REQ
5. PING_REQ
6. PING_RESP
7. NOTICE_RECEIPT_REQ
8. ERROR_REQ
9. DISCONNECT_REQ

ERROR_TYPE:

1. PACKAGE_BROKEN
2. READ_ERROR

CLICK_TYPE:

1. RIGHT

2. MIDDLE
3. LEFT

5. Security implementation

All the content of the data in the package defined in *2. Data Definition* is encrypted with the public key and decrypt with the private key of the sender. Only one exception is done with the AUTH_REQ.

The system use is RSA algorithm by the OpenSSL library.