**RFC SPIDER**

Crédits : Brendan BOBIS

**1. INTRODUCTION**

Le Protocole Spider est conçu pour supporter les communications entre un logiciel serveur et plusieurs clients appelés Keylogger.

[**2. VUE D'ENSEMBLE**](http://www.frameip.com/rfc/rfc791-fr.php#2)

Le protocole utilisé lors de l’envoi de données binaires entre un client et le serveur est le même que celui utilisé entre le serveur et un client. On appellera commande la valeur de l’attribut cmd contenu dans le type **t\_cmd**, cette valeur correspondra à une instruction donnée selon si elle est reçue sur le client ou le serveur. Toutes les valeurs possibles sont trouvable dans l’**enum commandType**.

**2. LISTE DES COMMANDES CLIENT**

Le client peut recevoir les commandes suivantes:

**BAD**: Utilisé pour signaler une commande invalide ou inexistante, sur certains client, cette commande ne donne pas obligatoirement lieu à une action sur le client.

**CLOSE**: Utilisé pour demander au client son arrêt immédiat.

**DOWNLOAD\_LOG**: Utilisé pour demander au client d’envoyer son fichier de log au serveur à des fins d’enregistrement. Pour cela, le client doit envoyer en retour au serveur une ou plusieurs structures contenant la commande DOWNLOAD\_LOG, un flux de donnée ainsi que la taille des donnés envoyées pour chaqu’une de ses structures, le client enverra ensuite **DOWNLOAD\_LOG\_END** pour signaler la fin de l’envoi des données.

**DISPLAY\_LOG**: Utilisé pour demander au client d’envoyer son fichier de log au serveur à des fins d’affichage. Pour cela, le client doit envoyer en retour au serveur une ou plusieurs structures contenant la commande DOWNLOAD\_LOG, un flux de donnée ainsi que la taille des donnés envoyées pour chaqu’une de ses structures, le client enverra ensuite **DISPLAY\_LOG\_END** pour signaler la fin de l’envoi des données.

**PURGE\_LOG**: Utilisé pour demander l'effacement du log du client

**3. LISTE DES COMMANDES SERVEUR**

Le serveur peut recevoir les commandes suivantes:

**BAD**: Utilisé pour signaler une commande invalide ou inexistante, sur certains client, cette commande ne donne pas obligatoirement lieu à une action sur le serveur ou le client.

**DOWNLOAD\_LOG**: Utilisé pour demander au serveur d’enregistrer des données dans un fichier préalablement ouvert sur le serveur.

**DOWNLOAD\_LOG\_END**: Utilisé pour signaler la fin de l’enregistrement d’un fichier ouvert sur le serveur et donc ordonner la fermeture de ce fichier.

**DISPLAY\_LOG**: Utilisé pour demander au serveur d’afficher des données dans un fichier préalablement ouvert sur le serveur.

**DISPLAY\_LOG\_END:** Utilisé pour signaler la fin de l’affichage des données

**4.** [**SPÉCIFICATION**](http://www.frameip.com/rfc/rfc791-fr.php#3) **ET STRUCTURE DE DONNÉES**

Le protocole prend la forme d’une structure de données qui a pour typedef **t\_cmd** et est définie de la sorte:

**enum commandType** { BAD = 0, CLOSE, DOWNLOAD\_LOG, DOWNLOAD\_LOG\_END, DISPLAY\_LOG, DISPLAY\_LOG\_END, PURGE\_LOG };

**typedef struct s\_cmd**

{

commandType cmd;

unsigned char data\_len;

char buffer[buffer\_size];

} **t\_cmd**;

