

# Dialoguer avec un serveur Counter-Strike : Source en PHP

par Adrien Pellegrini ([Page d'accueil](#))



Date de publication : 4/09/2007

Dernière mise à jour :

Comment récupérer des informations à propos d'un serveur Counter Strike : Source ? Comment lancer des commandes RCON depuis un site Web ? Comment faire une demande au "master server" de Steam pour récupérer la liste des serveurs disponibles ? Tout ceci vous est expliqué dans cet article.

- I - Introduction
  - I.1 - À lire
- II - Requête sur un serveur
  - II.1 - Challenge number
    - II.1.1 - Format de la requête
    - II.1.2 - Format de la réponse
    - II.1.3 - Script
  - II.2 - Informations sur le serveur
    - II.2.1 - Format de la requête
    - II.2.2 - Format de la réponse
    - II.2.3 - Script
  - II.3 - Informations sur les règles du serveur
    - II.3.1 - Format de la requête
    - II.3.2 - Format de la réponse
    - II.3.3 - Script
  - II.4 - Informations sur les joueurs
    - II.4.1 - Format de la requête
    - II.4.2 - Format de la réponse
    - II.4.3 - Script
- III - Commande RCON
  - III.1 - Format de la requête
  - III.2 - Format de la réponse
  - III.3 - Script
- IV - Lister les serveurs
  - IV.1 - Format de la requête
  - IV.2 - Format de la réponse
  - IV.3 - Script
- V - Scripts divers
  - V.1 - Voir l'état du serveur
  - V.2 - Fonctions utilisées
  - V.3 - Classe
- VI - Liens & remerciements

## I - Introduction

On peut facilement récupérer toutes sortes d'informations en envoyant des paquets  **TCP** ou  **UDP** prédéfinis à des servers. Ceux-ci nous renverront un ou plusieurs paquets, appelé réponse, qui contiendront l'information désirée.

Les paquets utilisés par Steam ont une taille de 1400 bytes plus les entêtes IP/UDP (ou de 4096 bytes pour les commandes RCON).

Les réponses obtenues par le serveur sont généralement peu lisibles. Pour les rendre plus lisible, j'ai écrit à la fin de cet article une série de fonctions qui permettent de convertir les réponses.

### I.1 - À lire

Chaque sous partie va être séparée comme suit :

- *Format de la requête* correspond à la commande qu'il faut envoyer au serveur pour récupérer la réponse appropriée.
- *Format de la réponse* explique les différentes parties qui composent la réponse.
- *Script* nous montre le code PHP utilisé.

## II - Requête sur un serveur

### Un serveur répond à quatre requêtes qui sont les suivantes :

- 1 récupérer un numéro de défi (challenge number)
- 2 récupérer des informations à propos du serveur
- 3 récupérer des informations sur les joueurs présents sur le serveur
- 4 récupérer des informations sur les règles du serveur

Les requêtes doivent se faire via le protocole  UDP.

### II.1 - Challenge number

Le challenge number est requis pour les requêtes concernant la récupération des règles du serveur ainsi que des joueurs présents dessus.

#### II.1.1 - Format de la requête

```
0020 ff ff ff ff 57          ....W
```

Donnée	Type
0xFFFFFFFF	int
W ou 0x57	byte

#### II.1.2 - Format de la réponse

```
0020 ff ff ff ff 41 fa 05 a4 00      ....A....
```

Donnée	Type	Explication
0xFFFFFFFF	int	-
Challenge	long	Le "challenge number" à utiliser

#### II.1.3 - Script

```
// Constant
define('PACKET_SIZE', '1400');
define('SERVERQUERY_GETCHALLENGE', "\xFF\xFF\xFF\xFF\x57");
define('REPLY_GETCHALLENGE', "\x41");

// Ip address and port
$_ip = '82.149.249.243';
$_port = '27017';

// Open a connection with server
$socket = stream_socket_client('udp://'.$_ip.':'.$_port, $errno, $errstr, 30);

// Send command to server
$cmd = SERVERQUERY_GETCHALLENGE;
$length = strlen($cmd);
fwrite($socket, $cmd, $length);
```

```
// Get response from server
$response = fread($socket, PACKET_SIZE);

// Clean response
$pattern = "#\xFF\xFF\xFF\xFF".REPLY_GETCHALLENGE."#";
$response = preg_replace($pattern, '', $response);
```

## II.2 - Informations sur le serveur

Cette partie consiste à récupérer toutes les informations disponibles sur un serveur tel que son nom, nombre de joueurs maximum, nombre de joueurs présents, carte en cours ...

### II.2.1 - Format de la requête

#### Requête au format hexadécimal

```
0020 FF FF FF FF 54 53 6F 75 72 63 65 20 45 6E 67 69  ȳȳȳȳTSource Engi
0030 6E 65 20 51 75 65 72 79 00                      ne Query
```

Donnée	Type
0xFFFFFFFF	int
T	byte
Source Engine Query	string

### II.2.2 - Format de la réponse

#### Réponse au format hexadécimal

```
0020 ff ff ff ff 49 07  ....I.
0030 77 77 77 2e 63 6c 61 6e 6b 69 6c 6c 65 72 7a 2e  www.clankillerz.
0040 64 65 20 2e 2e 3a 3a 50 75 62 6c 69 63 20 6f 6e  de ...:Public on
0050 6c 79 20 48 69 67 68 73 6b 69 6c 6c 3a 3a 2e 2e  ly Highskill:...
0060 00 64 65 5f 64 75 73 74 32 00 63 73 74 72 69 6b  .de_dust2.cstrik
0070 65 00 43 6f 75 6e 74 65 72 2d 53 74 72 69 6b 65  e.Counter-Strike
0080 3a 20 53 6f 75 72 63 65 00 f0 00 00 0f 00 64 77  : Source.....dw
0090 00 01 31 2e 30 2e 30 2e 33 34 00  ..1.0.0.34.
```

Donnée	Type	Explication
0xFFFFFFFF	int	-
Type	byte	Type est toujours égal à 'I' (0x49)
Version	byte	Version du réseau Steam, actuellement 0x07
Nom du server	string	-
Carte	string	-
Répertoire du jeu	string	Le nom du répertoire qui contient les fichiers du jeu (ex: "cstrike")
Description du jeu	string	Le nom complet du jeu (ex: "Counter-Strike : Source")
AppID	short	Steam Application ID, pour CSS c'est 240

Nombre de joueurs	byte	-
Nombre maximum de joueur	byte	-
Nombre de bots	byte	-
Serveur dédié	byte	'l' pour listen, 'd' pour dedicated et 'p' pour Source TV
OS	byte	Système d'exploitation; 'l' pour Linux et 'w' pour Windows
Mot de passe	byte	Indique si le serveur est verrouillé par mot de passe; si 0x01 il y a un mot de passe
Sécurisé	byte	Indique si VAC est activé; si 0x01 VAC est activé
Version du jeu	string	Version du jeu (ex: "1.0.0.34")

## II.2.3 - Script

### Envoi de la requête et récupération de la réponse

```
// Constant
define('PACKET_SIZE', '1400');
define('SERVERQUERY_INFO', "\xFF\xFF\xFF\xFFSource Engine Query");
define ('REPLY_INFO', "\x49");

// Ip address and port
$_ip = '82.149.249.243';
$_port = '27017';

// Open connection with server
$socket = stream_socket_client('udp://'.$_ip.':'.$_port, $errno, $errstr, 30);

// Send command to server
$cmd = SERVERQUERY_INFO;
$length = strlen($cmd);
fwrite($socket, $cmd, $length);

// Get response from server
$response = fread($socket, PACKET_SIZE);

// Clean response
$pattern = "#\xFF\xFF\xFF\xFF.REPLY_INFO.#";
$response = preg_replace($pattern, '', $response);
```

...:Public only Highskill::...de\_dust2cstrikeCounter-Strike: Source

*La réponse obtenue après la requête n'est pas très lisible.*

### Formatage de la réponse pour la rendre plus lisible

```
//Version - byte (Network version. 0x07 is the current Steam version.)
$server['version'] = getByte($response);

// Ip and port
$server['ip'] = $_ip;
$server['port'] = $_port;

//Server Name - string (The Source server's name, eg: "Recoil NZ CS Server #1")
```

**Formatage de la réponse pour la rendre plus lisible**

```
$server['name'] = trim(getString($response));

//Map - string (The current map being played, eg: "de_dust")
$server['map'] = getString($response);

//Game Directory - string (The name of the folder containing the game files, eg: "cstrike")
$server['gamedir'] = getString($response);

//Game Description- string (A friendly string name for the game type, eg: "Counter-Strike: Source")
$server['gamedesc'] = getString($response);

//AppID - short (Steam Application ID)
$server['appid'] = getShortSigned($response);

//Number of players - byte (The number of players currently on the server)
$server['numplayers'] = getByte($response);

//Maximum players - byte (Maximum allowed players for the server)
$server['maxplayers'] = getByte($response);

//Number of bots - byte (Number of bot players currently on the server)
$server['bot'] = getByte($response);

//Dedicated - byte ('l' for listen, 'd' for dedicated, 'p' for SourceTV)
$data = chr(getByte($response));

$server['dedicated'] = 0;
$server['sourcetv'] = 0;
$server['listen'] = 0;

if ($data == 'd') $server['dedicated'] = 1;
if ($data == 'p') $server['sourcetv'] = 1;
if ($data == 'l') $server['listen'] = 1;

//OS - byte (Host operating system. 'l' for Linux, 'w' for Windows)
$data = chr(getByte($response));

$server['os'] = 'undefined';

if ($data == 'l')
{
    $server['os'] = 'linux';
}
elseif ($data == 'w')
{
    $server['os'] = 'windows';
}

//Password - byte (If set to 0x01, a password is required to join this server)
$data = getByte($response);

$server['password'] = 0;

if ($data == 1)
{
    $server['password'] = 1;
}

//Secure - byte (if set to 0x01, this server is VAC secured)
$data = getByte($response);

$server['secure'] = 1;

if ($data == 1)
{
    $server['secure'] = 1;
}
```

### Formatage de la réponse pour la rendre plus lisible

```
//Game Version - string (The version of the game, eg: "1.0.0.22")
$server['gameversion'] = getString($response);
```

L'affichage du tableau \$server nous donne :

```
Array
(
    [version] => 7
    [ip] => 82.149.249.243
    [port] => 27017
    [name] => www.clankillerz.de :::Public only Highskill:...
    [map] => de_dust2
    [gamedir] => cstrike
    [gamedesc] => Counter-Strike: Source
    [appid] => 240
    [numplayers] => 0
    [maxplayers] => 15
    [bot] => 0
    [dedicated] => 1
    [sourcetv] => 0
    [listen] => 0
    [os] => windows
    [password] => 0
    [secure] => 1
    [gameversion] => 1.0.0.34
)
```

## II.3 - Informations sur les règles du serveur

Cette partie consiste à récupérer toutes les règles sur un serveur tel que mp\_timelimit, sv\_alltalk, ...

### II.3.1 - Format de la requête

```
FF FF FF FF 56 <4 byte challenge number> ....V
```

Donnée	Type
0xFFFFFFFF	int
V (ou 0x56)	byte
Challenge Number	long

### II.3.2 - Format de la réponse

#### Réponse au format hexadécimal

```
0020                ff ff ff ff 45 41                ....EA
0030  00 6d 61 74 74 69 65 5f 6d 75 67 6d 6f 64 00 31  .mattie_mugmod.1
0040  00 6d 61 74 74 69 65 5f 65 76 65 6e 74 73 63 72  .mattie_eventscr
0050  69 70 74 73 00 31 00 65 76 65 6e 74 73 63 72 69  ipt.1.eventscri
0060  70 74 73 5f 76 65 72 00 31 2e 33 2e 30 2e 30 30  pts_ver.1.3.0.00
0070  36 00 6d 61 6e 69 5f 72 65 73 65 72 76 65 5f 73  6.mani_reserve_s
0080  6c 6f 74 73 00 30 00 6d 61 6e 69 5f 61 64 6d 69  lots.0.mani_admi
0090  6e 5f 70 6c 75 67 69 6e 5f 76 65 72 73 69 6f 6e  n_plugin_version
00a0  00 31 2e 32 42 65 74 61 52 20 56 53 50 00 6d 61  .1.2BetaR VSP.ma
00b0  6e 69 5f 74 69 63 6b 72 61 74 65 00 31 30 30 00  ni_tickrate.100.
```

Réponse au format hexadécimal

```

00c0 6d 61 6e 69 5f 6e 65 78 74 6d 61 70 00 63 73 5f mani_nextmap.cs_
00d0 6f 66 66 69 63 65 00 65 73 74 5f 76 65 72 73 69 office.est_versi
00e0 6f 6e 00 30 2e 34 31 37 61 00 6d 70 5f 74 65 61 on.0.417a.mp_tea
00f0 6d 70 6c 61 79 00 30 00 6d 70 5f 66 72 61 67 6c mplay.0.mp_fragl
0100 69 6d 69 74 00 30 00 6d 70 5f 66 61 6c 6c 64 61 imit.0.mp_fallda
0110 6d 61 67 65 00 31 00 6d 70 5f 77 65 61 70 6f 6e mage.1.mp_weapon
0120 73 74 61 79 00 30 00 6d 70 5f 66 6f 72 63 65 72 stay.0.mp_forcer
0130 65 73 70 61 77 6e 00 31 00 6d 70 5f 66 6f 6f 74 espawn.1.mp_foot
0140 73 74 65 70 73 00 31 00 6d 70 5f 66 6c 61 73 68 steps.1.mp_flash
0150 6c 69 67 68 74 00 31 00 6d 70 5f 61 75 74 6f 63 light.1.mp_autoc
0160 72 6f 73 73 68 61 69 72 00 31 00 64 65 63 61 6c rosshair.1.decal
0170 66 72 65 71 75 65 6e 63 79 00 34 35 00 6d 70 5f frequency.45.mp_
0180 74 65 61 6d 6c 69 73 74 00 68 67 72 75 6e 74 3b teamlist.hgrunt;
0190 73 63 69 65 6e 74 69 73 74 00 6d 70 5f 61 6c 6c scientist.mp_all
01a0 6f 77 4e 50 43 73 00 31 00 6d 70 5f 66 72 69 65 owNPCs.1.mp_frie
01b0 6e 64 6c 79 66 69 72 65 00 31 00 73 76 5f 67 72 ndlyfire.1.sv_gr
01c0 61 76 69 74 79 00 38 30 30 00 73 76 5f 73 74 6f avity.800.sv_sto
01d0 70 73 70 65 65 64 00 37 35 00 73 76 5f 6e 6f 63 pspeed.75.sv_noc
01e0 6c 69 70 61 63 63 65 6c 65 72 61 74 65 00 35 00 lipaccelerate.5.
01f0 73 76 5f 6e 6f 63 6c 69 70 73 70 65 65 64 00 35 sv_noclipspeed.5
0200 00 73 76 5f 73 70 65 63 61 63 63 65 6c 65 72 61 .sv_specaccelera
0210 74 65 00 35 00 73 76 5f 73 70 65 63 73 70 65 65 te.5.sv_specspee
0220 64 00 31 00 73 76 5f 73 70 65 63 6e 6f 63 6c 69 d.1.sv_specnocli
0230 70 00 30 00 73 76 5f 6d 61 78 73 70 65 65 64 00 p.0.sv_maxspeed.
0240 33 32 30 00 73 76 5f 61 63 63 65 6c 65 72 61 74 320.sv_accelerat
0250 65 00 35 00 73 76 5f 61 69 72 61 63 63 65 6c 65 e.5.sv_airaccele
0260 72 61 74 65 00 31 30 00 73 76 5f 77 61 74 65 72 rate.10.sv_water
0270 61 63 63 65 6c 65 72 61 74 65 00 31 30 00 73 76 accelerate.10.sv
0280 5f 77 61 74 65 72 66 72 69 63 74 69 6f 6e 00 31 _waterfriction.1
0290 00 73 76 5f 66 6f 6f 74 73 74 65 70 73 00 31 00 .sv_footsteps.1.
02a0 73 76 5f 72 6f 6c 6c 73 70 65 65 64 00 32 30 30 sv_rollspeed.200
02b0 00 73 76 5f 72 6f 6c 6c 61 6e 67 6c 65 00 30 00 .sv_rollangle.0.
02c0 73 76 5f 66 72 69 63 74 69 6f 6e 00 34 00 73 76 sv_friction.4.sv
02d0 5f 62 6f 75 6e 63 65 00 30 00 73 76 5f 73 74 65 _bounce.0.sv_ste
02e0 70 73 69 7a 65 60 31 38 00 72 5f 56 65 68 69 63 psize.18.r_Vehic
02f0 6c 65 56 69 65 77 44 61 6d 70 65 6e 00 31 00 72 leViewDampen.1.r
0300 5f 4a 65 65 70 56 69 65 77 44 61 6d 70 65 6e 46 _JeepViewDampenF
0310 72 65 71 00 37 2e 30 00 72 5f 4a 65 65 70 56 69 req.7.0.r_JeepVi
0320 65 77 44 61 6d 70 65 6e 44 61 6d 70 00 31 2e 30 ewDampenDamp.1.0
0330 00 72 5f 4a 65 65 70 56 69 65 77 5a 48 65 69 67 .r_JeepViewZHeig
0340 68 74 00 31 30 2e 30 00 72 5f 41 69 72 62 6f 61 ht.10.0.r_Airboa
0350 74 56 69 65 77 44 61 6d 70 65 6e 46 72 65 71 00 tViewDampenFreq.
0360 37 2e 30 00 72 5f 41 69 72 62 6f 61 74 56 69 65 7.0.r_AirboatVie
0370 77 44 61 6d 70 65 6e 44 61 6d 70 00 31 2e 30 00 wDampenDamp.1.0.
0380 72 5f 41 69 72 62 6f 61 74 56 69 65 77 5a 48 65 r_AirboatViewZHe
0390 69 67 68 74 00 30 2e 30 00 6d 70 5f 74 69 6d 65 ight.0.0.mp_time
03a0 6c 69 6d 69 74 00 35 30 30 00 73 76 5f 61 6c 6c limit.500.sv_all
03b0 74 61 6c 6b 00 30 00 6d 70 5f 64 79 6e 61 6d 69 talk.0.mp_dynami
03c0 63 70 72 69 63 69 6e 67 00 30 00 6e 65 78 74 6c cpricing.0.nextl
03d0 65 76 65 6c 00 6d 70 5f 61 75 74 6f 74 65 61 evel..mp_autotea
03e0 6d 62 61 6c 61 6e 63 65 00 31 00 6d 70 5f 6d 61 mbalance.1.mp_ma
03f0 78 72 6f 75 6e 64 73 00 30 00 6d 70 5f 72 6f 75 xrounds.0.mp_rou
0400 6e 64 74 69 6d 65 00 34 00 6d 70 5f 66 72 65 65 ndtime.4.mp_free
0410 7a 65 74 69 6d 65 00 35 00 6d 70 5f 63 34 74 69 zetime.5.mp_c4ti
0420 6d 65 72 00 33 35 00 6d 70 5f 6c 69 6d 69 74 74 mer.35.mp_limitt
0430 65 61 6d 73 00 30 00 6d 70 5f 68 6f 73 74 61 67 eams.0.mp_hostag
0440 65 70 65 6e 61 6c 74 79 00 33 00 73 76 5f 76 6f epenalty.3.sv_vo
0450 69 63 65 65 6e 61 62 6c 65 00 31 00 73 76 5f 63 iceenable.1.sv_c
0460 6f 6e 74 61 63 74 00 77 77 77 2e 63 6c 61 6e 6b ontact.www.clank
0470 69 6c 6c 65 72 7a 2e 64 65 00 73 76 5f 70 61 75 illerz.de.sv_pau
0480 73 61 62 6c 65 00 30 00 73 76 5f 63 68 65 61 74 sable.0.sv_cheat
0490 73 00 30 00 63 6f 6f 70 00 30 00 64 65 61 74 68 s.0.coop.0.death
04a0 6d 61 74 63 68 00 31 00 74 76 5f 70 61 73 73 77 match.1.tv_passw
04b0 6f 72 64 00 30 00 74 76 5f 72 65 6c 61 79 70 61 ord.0.tv_relaypa
04c0 73 73 77 6f 72 64 00 30 00 73 76 5f 70 61 73 73 ssword.0.sv_pass
04d0 77 6f 72 64 00 30 00 word.0.

```

Donnée	Type	Explication
0xFFFFFFFF	int	-
Type	byte	Type est toujours égal à 'E' (0x45)
Nombre de règle	short	-
Nom de la règle	string	-
Valeur de la règle	string	-

### II.3.3 - Script

#### Envoi de la requête et récupération de la réponse

```
// Constant
define('PACKET_SIZE', '1400');
define('SERVERQUERY_GETCHALLENGE', "\xFF\xFF\xFF\xFF\x57");
define('SERVERQUERY_RULES', "\xFF\xFF\xFF\xFF\x56");
define('REPLY_GETCHALLENGE', "\x41");
define('REPLY_RULES', "\x45");

// Ip address and port
$_ip = '82.149.249.243';
$_port = '27017';

// Open connection with server
$socket = stream_socket_client('udp://'.$_ip.':'.$_port, $errno, $errstr, 30);

// Get the challenge number
// Send command to server
$cmd = SERVERQUERY_GETCHALLENGE;
$length = strlen($cmd);
fwrite($socket, $cmd, $length);

// Get response from server
$response = fread($socket, PACKET_SIZE);

// Filter the response
$pattern = "\xFF\xFF\xFF\xFF".REPLY_GETCHALLENGE."^";
$challengeNumber = preg_replace($pattern, '', $response);

// Get the server rules info
// Send command to server
$cmd = SERVERQUERY_RULES.$challengeNumber;
$length = strlen($cmd);
fwrite($socket, $cmd, $length);

// Get response from server
$response = fread($socket, PACKET_SIZE);

// Clean response
$pattern = "#\xFF\xFF\xFF\xFF".REPLY_RULES."#";
$response = preg_replace($pattern, '', $response);
```

#### Formatage de la réponse pour la rendre plus lisible

```
// Number of rules
$ruleNumber = getShortSigned($response);

while ($response !== false)
{
    //Rule Name - string (The name of the rule)
    $name = getString($response);

    //Rule Value - string (The rule's value)
    $rules[$name] = getString($response);
}
```

**Formatage de la réponse pour la rendre plus lisible**

}

L'affichage du tableau `$rules` nous donne :

**Array**

```
(
  [mattie_mugmod] => 1
  [mattie_eventscripts] => 1
  [eventscripts_ver] => 1.3.0.006
  [mani_reserve_slots] => 0
  [mani_admin_plugin_version] => 1.2BetaR VSP
  [mani_tickrate] => 100
  [mani_nextmap] => cs_office
  [est_version] => 0.417a
  [mp_teamplay] => 0
  [mp_fraglimit] => 0
  [mp_falldamage] => 1
  [mp_weaponstay] => 0
  [mp_forcerespawn] => 1
  [mp_footsteps] => 1
  [mp_flashlight] => 1
  [mp_autocrosshair] => 1
  [decalfrequency] => 45
  [mp_teamlist] => hgrunt;scientist
  [mp_allowNPCs] => 1
  [mp_friendlyfire] => 1
  [sv_gravity] => 800
  [sv_stopspeed] => 75
  [sv_noclipaccelerate] => 5
  [sv_noclip speed] => 5
  [sv_specaccelerate] => 5
  [sv_specs speed] => 1
  [sv_specnoclip] => 0
  [sv_maxspeed] => 320
  [sv_accelerate] => 5
  [sv_airaccelerate] => 10
  [sv_wateraccelerate] => 10
  [sv_waterfriction] => 1
  [sv_footsteps] => 1
  [sv_rolls speed] => 200
  [sv_rollangle] => 0
  [sv_friction] => 4
  [sv_bounce] => 0
  [sv_stepsize] => 18
  [r_VehicleViewDampen] => 1
  [r_JeepViewDampenFreq] => 7.0
  [r_JeepViewDampenDamp] => 1.0
  [r_JeepViewZHeight] => 10.0
  [r_AirboatViewDampenFreq] => 7.0
  [r_AirboatViewDampenDamp] => 1.0
  [r_AirboatViewZHeight] => 0.0
  [mp_timelimit] => 500
  [sv_alltalk] => 0
  [mp_dynamicpricing] => 0
  [nextlevel] =>
  [mp_autoteambalance] => 1
  [mp_maxrounds] => 0
  [mp_roundtime] => 4
  [mp_freezetime] => 5
  [mp_c4timer] => 35
  [mp_limitteams] => 0
  [mp_hostagepenalty] => 3
  [sv_voiceenable] => 1
  [sv_contact] => www.clankillerz.de
)
```

```
[sv_pausable] => 0
[sv_cheats] => 0
[coop] => 0
[deathmatch] => 1
[tv_password] => 0
[tv_relaypassword] => 0
[sv_password] => 0
)
```

## II.4 - Informations sur les joueurs

Cette partie consiste à récupérer quelques informations à propos des joueurs présents sur le serveur tel que leur pseudo, nombre de skills et temps présents sur le serveur.

### II.4.1 - Format de la requête

```
0020 FF FF FF FF 55 <4 byte challenge number> ....U
```

Donnée	Type
0xFFFFFFFF	int
U (ou 0x55)	byte
Challenge Number	long

### II.4.2 - Format de la réponse

```
0020 ff ff ff ff 44 08 .....D.
0030 00 c2 bb 7d c3 87 4b c5 bd 7b c2 ab 30 30 37 7c ...}.K..{..007|
0040 6c 6f 57 62 49 72 64 00 33 00 00 00 39 97 6d 45 loWbIrd.3...9.mE
0050 01 42 69 6c 6c 79 74 68 65 4b 69 64 00 05 00 00 .BillytheKid....
0060 00 83 01 82 44 02 c5 a0 c4 ac c5 98 c5 8e e2 84 ....D.....
0070 a2 c2 aa c2 b9 7c 20 4c 75 6b 40 73 31 39 38 39 .....| Luk@s1989
0080 28 63 68 29 00 01 00 00 00 3d 5a 49 43 03 44 72 (ch).....=ZIC.Dr
0090 2e 45 76 69 6c 20 5b 47 45 52 5d 00 01 00 00 00 .Evil [GER].....
00a0 cf 29 a3 42 04 46 61 74 74 69 58 78 5e 00 0e 00 .).B.FattiXx^...
00b0 00 00 dc 5f 4d 44 05 5e 53 6f 4f 6e 59 20 3a 3e ..._MD.^SoOnY :>
00c0 20 3a 3e 20 3a 3e 00 01 00 00 00 2b 23 8e 42 07 :> :>.....+#.B.
00d0 49 73 4e 23 50 77 4e 62 00 01 00 00 00 b4 bd 21 IsN#PwNb.....!
00e0 43 0d 21 62 45 6a 6b 32 31 4f 2a 00 1e 00 00 00 C.!bEjk210*.....
00f0 66 a9 0b 45 f..E
```

Donnée	Type	Explication
0xFFFFFFFF	int	-
Type	byte	Type est toujours égal à 'D' (0x44)
Nombre de joueurs	byte	-
Index	byte	Index compris entre 0 et le nombre de joueurs
Nom du joueur	string	-
Kills	long	-
Temps connecté	float	Temps en seconde

### II.4.3 - Script

### Envoi de la requête et récupération de la réponse

```
// Constant
define('PACKET_SIZE', '1400');
define('SERVERQUERY_GETCHALLENGE', "\xFF\xFF\xFF\xFF\x57");
define('SERVERQUERY_PLAYER', "\xFF\xFF\xFF\xFF\x55");
define('REPLY_GETCHALLENGE', "\x41");
define('REPLY_PLAYER', "\x44");

// Ip adress and port
$_ip = '82.149.249.243';
$_port = '27017';

// Open connection with server
$socket = stream_socket_client('udp://'.$_ip.':'.$_port, $errno, $errstr, 30);

// Get the challenge number
// Send command to server
$cmd = SERVERQUERY_GETCHALLENGE;
$length = strlen($cmd);
fwrite($socket, $cmd, $length);

// Get response from server
$response = fread($socket, PACKET_SIZE);

// Filter the response
$pattern = "\xFF\xFF\xFF\xFF.REPLY_GETCHALLENGE.#";
$challengeNumber = preg_replace($pattern, '', $response);

// Get the server rules info
// Send command to server
$cmd = SERVERQUERY_PLAYER.$challengeNumber;
$length = strlen($cmd);
fwrite($socket, $cmd, $length);

// Get response from server
$response = fread($socket, PACKET_SIZE);

// Clean response
$pattern = "#\xFF\xFF\xFF\xFF.REPLY_PLAYER.#";
$response = preg_replace($pattern, '', $response);
```

### Formatage de la réponse pour la rendre plus lisible

```
//Num Players - byte (The number of players reported in this response)
$num = getByte($response);

$i = 0;
while ($response !== false)
{
//Index - byte (The index into [0.. Num Players] for this entry)
$index = getByte($response);

//Player Name - string (Player's name)
$players[$index]['name'] = trim(getString($response));

//Kills - long (Number of kills this player has)
$players[$index]['kills'] = getLong($response);

//Time connected - float (The time in seconds this player has been connected)
$players[$index]['time'] = getFloat($response);

$i++;
}
```

L'affichage du tableau `$players` nous donne :

```
Array
(
    [0] => Array
        (
            [name] => Å»}Ã#KÃ¼{Ã«007|lowbIrd
            [kills] => 51
            [time] => 3801.45141602
        )

    [1] => Array
        (
            [name] => BillytheKid
            [kills] => 5
            [time] => 1040.04724121
        )

    [2] => Array
        (
            [name] => Å Ä-Å#Å#â#çÃªÃ¹ | Luk@s1989(ch)
            [kills] => 1
            [time] => 201.352493286
        )

    [3] => Array
        (
            [name] => Dr.Evil [GER]
            [kills] => 1
            [time] => 81.5816574097
        )


    [4] => Array
        (
            [name] => FattiXx^
            [kills] => 14
            [time] => 821.497802734
        )

    [5] => Array
        (
            [name] => ^SoOnY :> :> :>
            [kills] => 1
            [time] => 71.068687439
        )

    [7] => Array
        (
            [name] => IsN#PwNb
            [kills] => 1
            [time] => 161.741027832
        )

    [13] => Array
        (
            [name] => !bEjk210*
            [kills] => 30
            [time] => 2234.58740234
        )
)
```

### III - Commande RCON

Les commandes RCON doivent être envoyées via le protocole  TCP.

#### III.1 - Format de la requête

Donnée	Type	Explication
Taille du paquet	int	-
Request ID	int	Nombre quelconque qui doit être incrémenté entre deux commandes RCON
Type de la requête	int	Deux types de requête existent :  - "SERVERDATA_AUTH", il faut envoyer l'entier 3  - "SERVERDATA_EXECCOMMAND", il faut envoyer l'entier 2
Chaine 1	string	La commande RCON à exécuter ("cvarlist", "changelevel de_dust2", ...)
Chaine 2	string	Chaine vide

#### III.2 - Format de la réponse

Si la commande RCON : "cvarlist" a été envoyée au serveur, la réponse contient bien plus de caractères que peut contenir un paquet. Il faudra alors gérer les "multi-paquets".

La taille d'un paquet ne dépassera jamais les 4096 bytes.

Steam ne nous permet pas de savoir quand nous sommes en train de lire le dernier paquet. Il faudra donc faire une boucle qui lit les paquets et jouer avec le timeout.

Donnée	Type	Explication
Taille du paquet	int	-
Request ID	int	Le Request ID sera égal à celui envoyé.
Type de la réponse	int	Deux types de réponse existent :  pour "SERVERDATA_AUTH", si le bon mot de passe a été fournis, alors la réponse vaudra 1 sinon elle sera égale à -1.
Chaine 1	string	La réponse à la commande soit un message d'erreur.
Chaine 2	string	Chaine vide

#### III.3 - Script

La première chose à faire avant d'envoyer la commande au serveur est de s'authentifier grâce à la commande "SERVERDATA\_AUTH".

Si la réponse à cette commande vaut -1, un mauvais mot de passe a été fourni. Si la réponse vaut 2, l'authentification a réussi. Tout autre nombre résulte en un échec de l'authentification.

Si l'authentification est réussie, on peut alors envoyer notre commande.

Si la commande est correct, vous aurez le résultat sinon un message d'erreur sera fourni.

Il n'y a pas besoin de formater les réponses car elles sont soit vides, soit en texte normal non formaté.

#### Demande d'authentification au serveur

```
// Constant
define('PACKET_SIZE', '1400');
define('SERVERDATA_AUTH', 3);
define('SERVERDATA_EXECCOMMAND', 2);

// Ip adress and port
$_ip = '192.168.1.2';
$_port = '27015';
$_password = '123456789';
$command = 'cvarlist';
$s2 = '';
$requestId = 1;

// -- Open connection with server
$socket = stream_socket_client('tcp://'.$_ip.':'.$_port, $errno, $errstr, 30);
stream_set_timeout($socket, 1, 0);

// -- Send auth packet
// Construct packet
$data = pack("VV", $requestId, SERVERDATA_AUTH).$_password.chr(0).$s2.chr(0);

// Prefix the packet by it's size
$data = pack("V", strlen($data)).$data;

// Send packet
fwrite($socket, $data, strlen($data));

$requestId++;

// Check if auth is successful
$junk = fread($socket, PACKET_SIZE);

$string = fread($socket, PACKET_SIZE);
$size = getLong($string);
$id = getLong($string);

if ($id == -1)
{
    // Error
    die('Auth failed : bad password !');
}
```

#### Envoi de la commande 'cvarlist' et récupération de la réponse

```
// -- Send command
// Construct packet
$data = pack("VV", $requestId, SERVERDATA_EXECCOMMAND).$command.chr(0).$s2.chr(0);

// Prefix the packet by it's size
$data = pack("V", strlen($data)).$data;
```

## Envoi de la commande 'cvarlist' et récupération de la réponse

```
// Send packet
fwrite($socket, $data, strlen($data));

$requestId++;

// -- Read response
$i = 0;
$text = '';

while ($string = fread($socket, 4))
{
    $info[$i]['size'] = getLong($string);

    $string = fread($socket, $info[$i]['size']);

    $info[$i]['id'] = getLong($string);
    $info[$i]['type'] = getLong($string);
    $info[$i]['s1'] = getString($string);
    $info[$i]['s2'] = getString($string);

    $text .= $info[$i]['s1'];
}
```

## IV - Lister les serveurs

Pour récupérer la liste des serveurs, il faut envoyer une requête spécial au "master server" de Steam. Notez que cette liste n'est pas a 100% fiable.

Le serveur à l'adresse : hl2master.steampowered.com:27011. Son IP pourra changer, donc il faudra appliquer la fonction **gethostbyname()** avant de passer l'IP à la fonction de connexion aux serveurs.

Cette requête doit se faire via le protocole  **UDP**.

### IV.1 - Format de la requête

```

0020                               31 03 30 2e 30 2e                1.0.0.
0030   30 2e 30 3a 30 00 5c 74 79 70 65 5c 64 5c 73 65   0.0:0.\type\d\se
0040   63 75 72 65 5c 31 5c 67 61 6d 65 64 69 72 5c 63   cure\1\gamedir\c
0050   73 74 72 69 6b 65 5c 6d 61 70 5c 64 65 5f 64 75   strike\map\de_du
0060   73 74 32 00                                         st2.
    
```

Donnée	Type	Explication
1 (ou 0x31)	byte	-
Code de la région	byte	Voir la tableau ci dessous pour la liste complète des région.
Adresse IP : port	string	L'adresse IP de départ est 0.0.0.0:  Comme vous pourrez en douter, la liste des serveurs se récupèrera sur plusieurs paquets.  Il faudra donc incrémenter l'adresse IP avec le dernier serveur reçu dans la réponse  pour pouvoir récupérer le reste afin de repartir du bon endroit.
0x00	byte	-
Filtre	string	Un filtre permet de spécifier des critères de recherche tels que le nom du jeu, la carte, l'os, ...
0x00	byte	-

Voici la liste des différentes régions :

Byte	Description
0x00	Côte Est des États-Unis
0x01	Côte Ouest des États-Unis
0x02	Amérique du Sud
0x03	Europe

0x04	Asie
0x05	Australie
0x06	Moyen-Orient
0x07	Afrique
0xFF	Toute - Autre

Voici la liste des différents filtres :

String	Description
\type\d	Serveur dédié
\secure\1	Serveur sécurisé
\gamedir\[mod]	Jeu sur le serveur (ex: cstrike)
\map\[map]	Carte sur le serveur
\linux\1	Serveur linux
\empty\1	Serveur non vide
\full\1	Serveur non rempli
\proxy\1	Serveur qui a des spectateurs

## IV.2 - Format de la réponse

### Exemple d'un paquet récupéré

```

0020                ff ff ff ff 66 0a                ....f.
0030 50 c6 7c 56 69 90 50 be fd dc 69 aa 50 dd 1e ac  P.|Vi.P...i.P...
0040 69 88 50 be fb c2 69 c8 50 be fb c2 69 8d 50 be  i.P...i.P...i.P.
0050 f8 33 69 88 50 ed cb df 69 87 50 be f6 2f 69 87  .3i.P...i.P../i.
      ...
0570 51 13 db 21 69 87 50 be 48 1f 69 87 51 13 db 21  Q..!i.P.H.i.Q..!
0580 69 91 51 13 db 24 69 87 51 13 db 26 69 87 50 be  i.Q..$i.Q..&i.P.
0590 48 0e 69 87 51 13 db 27 69 87                H.i.Q..'i.

```

Donnée	Type	Explication
0xFFFFFFFF	int	-
Type	byte	Type est toujours égal à '0x660A'
1 er partie de l'adresse IP	byte	-
2 ième partie de l'adresse IP	byte	-
3 ième partie de l'adresse IP	byte	-
4 ième partie de l'adresse IP	byte	-
Port	unsigned short	-

## IV.3 - Script

Pour récupérer la liste des serveurs il faut faire un boucle qui s'exécutera tant que la dernière IP trouvée n'est pas 0.0.0.0:

```

// Constant
define('PACKET_SIZE', 1400);

// Master Server info
define('MASTER_SERVER_HOST', 'hl2master.steampowered.com');
define('MASTER_SERVER_PORT', 27011);
define('MASTER_SERVER_QUERY', "\x31");

```

```
define('MASTER_SERVER_REPLY', "\x66\x0A");
define('MASTER_SERVER_IP', "0.0.0.0:0");

// Areas
define('REGION_US_EAST', "\x00");
define('REGION_US_WEST', "\x01");
define('REGION_US_SOUTH', "\x02");
define('REGION_EUROPE', "\x03");
define('REGION_ASIA', "\x04");
define('REGION_AUSTRALIA', "\x05");
define('REGION_MIDDLE_EAST', "\x06");
define('REGION_AFRICA', "\x07");
define('REGION_OTHER', "\xFF");

// Filters
define('FILTER_ALL', '');
define('FILTER_DEDICATED', "\x5C\x74\x79\x70\x65\x5C\x64"); //"\type\d
define('FILTER_SECURE', "\x5C\x73\x65\x63\x75\x72\x65\x5C\x31"); //"\secure\1
define('FILTER_GAME', "\x5C\x67\x61\x6D\x65\x64\x69\x72\x5C\x63\x73\x74\x72\x69\x6B\x65");
//"\gamedir\cstrike
define('FILTER_MAP_DE_DUST2', "\x5C\x6D\x61\x70\x5C\x64\x65\x5F\x64\x75\x73\x74\x32");
//"\map\de_dust2
define('FILTER_LINUX', "\x5C\x6C\x69\x6E\x75\x78\x5C\x31"); //"\linux\1
define('FILTER_NOT_EMPTY', "\x5C\x65\x6D\x70\x74\x79\x5C\x31"); //"\empty\1
define('FILTER_NOT_FULL', "\x5C\x66\x75\x6C\x6C\x5C\x31"); //"\full\1
define('FILTER_PROXY', "\x5C\x70\x72\x6F\x78\x79\x5C\x31"); //"\proxy\1

// Open connection with server
$socket = stream_socket_client(
    'udp://'.gethostbyname(MASTER_SERVER_HOST).':'.MASTER_SERVER_PORT,
    $errno,
    $errstr,
    30
);

$list = array();

// Start IP
$currentIP = MASTER_SERVER_IP;

do
{
    // Construct request packet, min must contain :
    // Byte - Message Type (0x31 - the character "1")
    // Byte - Region Code
    // String Zero - IP:Port
    // String Zero - Filter
    $packet =
        MASTER_SERVER_QUERY
        .REGION_OTHER
        . $currentIP
        . "\x00"
        . "\x00";

    // Send packet
    fwrite($socket, $packet);

    // Retrieve data
    $string = fread($socket, PACKET_SIZE);

    // Clean response
    $pattern = "#\xFF\xFF\xFF\xFF".MASTER_SERVER_REPLY."#";
    $response = preg_replace($pattern, '', $string);

    while ($string)
    {
        // Parse data
        // IP adress - 4 Byte
```

```
$ip = getByte($string);
$ip .= '.'.getByte($string);
$ip .= '.'.getByte($string);
$ip .= '.'.getByte($string);

// Port number - unsigned short
$port = getShortUnsigned($string);

$list[] = array(
    'ip' => $ip,
    'port' => $port
);

$currentIP = $ip.':'.$port;
}
} while($currentIP != MASTER_SERVER_IP);

var_dump($list);
```

Ce bout de code a pour résultat une erreur bien sûr. Le nombre de serveur récupérer de cette façon (sans aucun filtre) dépasse de loin la taille de mémoire autorisée par défaut par PHP.

Voici la variable `$packet` avec des filtres.

```
$packet =
MASTER_SERVER_QUERY
.REGION_EUROPE
.$currentIP
."\x00"
.FILTER_DEDICATED
.FILTER_SECURE
.FILTER_GAME
.FILTER_MAP_DE_DUST2
."\x00";
```

## V - Scripts divers

### V.1 - Voir l'état du serveur

Ce script permet de voir si le serveur est disponible ou s'il est déconnecté.

Pour se faire, on peut tout simplement utiliser le premier script de cet article (celui qui récupère le "challenge number") et tester si la réponse obtenue est vide ou pas.

Ceci étant la manière de faire la plus sûre car elle vérifie en même temps si le serveur accepte le protocole de Steam.

Il ne faut par contre pas oublier de spécifier un timeout, car si le serveur est hors-ligne, il va bloquer la page pendant le nombre de secondes défini par le timeout par défaut (30).

```
// Ip address and port
$_ip = '82.149.249.243';
$_port = '2707';

// Open a connection with server
$socket = stream_socket_client('udp://'.$_ip.':'.$_port, $errno, $errstr, 30);
stream_set_timeout($socket, 1, 0);

// Send command to server
$cmd = SERVERQUERY_GETCHALLENGE;
$length = strlen($cmd);
fwrite($socket, $cmd, $length);

// Get response from server
$response = fread($socket, PACKET_SIZE);

if (empty($response))
{
    echo 'Server Offline';
}
else
{
    echo 'Server Online';
}
```

### V.2 - Fonctions utilisées

```
/**
 * Return a byte and split it out of the string
 * - unsigned char
 *
 * @param string $string String
 */
function getByte(&$string)
{
    $data = substr($string, 0, 1);

    $string = substr($string, 1);

    $data = unpack('Cvalue', $data);

    return $data['value'];
}
```

```
/**
 * Return an unsigned short and split it out of the string
 * - unsigned short (16 bit, big endian byte order)
 *
 * @param string $string String
 */
function getShortUnsigned(&$string)
{
    $data = substr($string, 0, 2);

    $string = substr($string, 2);

    $data = unpack('nvalue', $data);

    return $data['value'];
}

/**
 * Return a signed short and split it out of the string
 * - signed short (16 bit, machine byte order)
 *
 * @param string $string String
 */
function getShortSigned(&$string)
{
    $data = substr($string, 0, 2);

    $string = substr($string, 2);

    $data = unpack('svalue', $data);

    return $data['value'];
}

/**
 * Return a long and split it out of the string
 * - unsigned long (32 bit, little endian byte order)
 *
 * @param string $string String
 */
function getLong(&$string)
{
    $data = substr($string, 0, 4);

    $string = substr($string, 4);

    $data = unpack('Vvalue', $data);

    return $data['value'];
}

/**
 * Return a float and split it out of the string
 *
 * @param string $string String
 */
function getFloat(&$string)
{
    $data = substr($string, 0, 4);

    $string = substr($string, 4);

    $array = unpack("fvalue", $data);

    return $array['value'];
}

/**
```

```
* Return a string and split it out of the string
*
* @param string $string String
*/
function getString(&$string)
{
    $data = "";

    $byte = substr($string, 0, 1);

    $string = substr($string, 1);

    while (ord($byte) != "0")
    {
        $data .= $byte;
        $byte = substr($string, 0, 1);
        $string = substr($string, 1);
    }

    return $data;
}
```

## V.3 - Classe

Voici une classe qui regroupe toutes les fonctionnalités vues précédemment.

**Source** [Classe à télécharger ici.](#)

## VI - Liens & remerciements

Tous mes remerciements à **Yogui** et **loka** pour leur relecture.

