A Closer Look at the Low Orbit Ion Cannon
and
Distributed Denial of Service

While I largely believe in what 4Chan stands for, especially when it put a stop to ACS:Law sending threat letters to pensioners who were in fact innocent of copyright theft, the methods are still questionable. Only a small minority of Anonymous are real hackers who know how LOIC works, can write their own scripts, and are familiar enough with the law to get themselves out of trouble if they get caught. The rest are putting themselves at risk, at least until the hackers there start employing the more effective ways of slowing down servers and covering their tracks, unless they're already doing that and using LOIC as cover.

Readers should also be aware this is an entirely different game from previous efforts that took down the copyright law firms. The current targets are larger companies who employ real hackers and professionals to manage their servers, which is why relatively little disruption has been caused.
Hopefully this article will give enough information to deter those who don't know the risks from joining Anonymous and ending up with a 10 year prison sentence.

Overview

Sophos Security, who have taken a less tolerant attitude to LOIC since 2008, has posted a more general introduction to the code on its site. This post goes into a little more detail. The LOIC DDoS attacks work like this:

1. Download the LOIC client
2. Configure the client to connect to an IRC server
3. The target gets flooded with requests from the LOIC clients operating in 'Hive' mode

This is a classic Distributed Denial of Service (DDoS) using a botnet, except in this case people volunteer to join it. It's important to note the LOIC client is a legitimate security testing application, apparently developed by Praetox Technologies. It does not include code for masking the originator's IP address, which will show up somewhere on the target server's logs and can easily be traced back to the user's ISP account, and eventually the local router. A couple of teenagers have already been arrested and police are now investigating the latest round of DDoS attacks.

Source Code

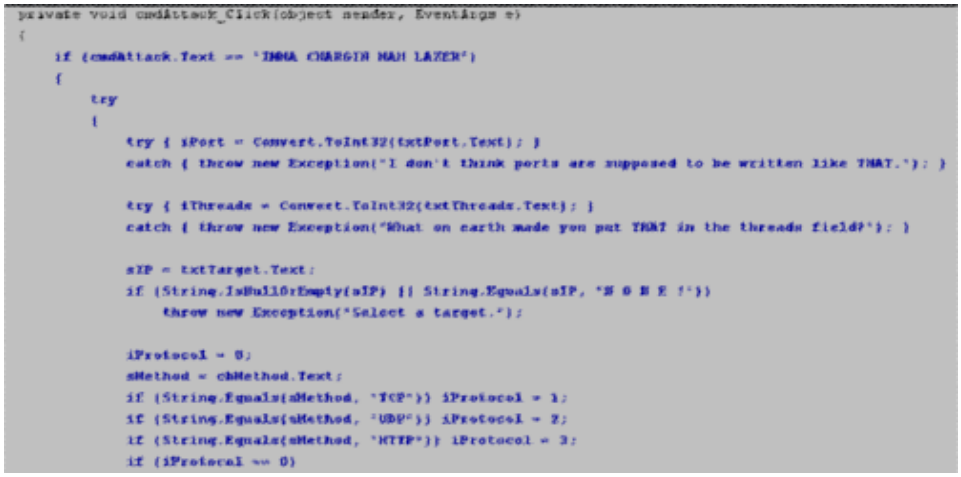
The C# source code for the LOIC client is available at GitHub for anyone who wants to look at it, and the executable should be found in the /bin directory. Readers might want to test the client on their own servers to see what shows up on the logs.
Most the files are for creating the interface, but three of them are of interest:

- *frmMain.cs*
- *HTTPFlooder.cs*
- *Program.cs*

Main Form/GUI Code

The file *frmMain.cs* generates the main part of the user interface, and where the user specifies the URL or IP address of the target server. When the command *IMMA CHARGIN MAH LAZER* is

received, the program does a series of checks for valid addresses, port numbers, payload, etc. before running the DDoS code for whichever of the three methods (TCP, UDP or XXP) is selected, until the command *Stop Flooding* is entered.
The rest of the code in that file's for displaying the current status of the attack.



```
private void cmdAttack_Click(object sender, EventArgs e)
{
    if (cmdAttack.Text == "IMMA CHARGIN MAH LAZER")
    {
        try
        {
            try { iPort = Convert.ToInt32(txtPort.Text); }
            catch { throw new Exception("I don't think ports are supposed to be written like THAT."); }

            try { iThreads = Convert.ToInt32(txtThreads.Text); }
            catch { throw new Exception("What on earth made you put THAT in the threads field?"); }

            sIP = txtTarget.Text;
            if (String.IsNullOrEmpty(sIP) || String.Equals(sIP, "S O R E ?"))
                throw new Exception("Select a target.");

            iProtocol = 0;
            sMethod = cbMethod.Text;
            if (String.Equals(sMethod, "TCP")) iProtocol = 1;
            if (String.Equals(sMethod, "UDP")) iProtocol = 2;
            if (String.Equals(sMethod, "HTTP")) iProtocol = 3;
            if (iProtocol == 0)
```

IRC And Hive Mode

In the 'Hive' mode, which is enabled with */hivemind* entered, commands are sent to the LOIC client through IRC. The IRC server, channel and port are set through on of the Windows forms and defined in *Program.cs*, which uses the C# *SmartIRC4NET* library.

```
/* IRC */
string ircserver = "";
string ircport = "";
string ircchannel = "";
/* Lets try this! */
 int count = 0;
foreach (string s in cmdLine)
    {
        /* IRC */
        if (s.ToLower() == "/hivemind")
    {
                hive = true;
                ircserver = cmdLine[count + 1]; //if no server entered let it crash
                try {ircport = cmdLine[count + 2];}
                catch (Exception) {ircport = "6667";} //default
                try {ircchannel = cmdLine[count + 3];}
                catch (Exception) {ircchannel = "#loic";} //default
     }
       /* Lets try this! */
     if (s.ToLower() == "/hidden") {hide = true;}
     count++;
```

As you can see in the code, the default is channel *#loic* at *port 6667*. In this mode, the user has volunteered to join the botnet which collectively sends requests to whatever Anonymous decides the target is.

A typical command recieved by the client through IRC sets the parameters:

default targethost=http://server.com subsite=/ speed=3
threads=15 method=tcp message=Enjoy_the_DDoS port=80 start