

# **Imaging Over NetCat (GUI)**

4050 - 581

By Alex O'Connor

## **Table of Contents**

- Executive Summary – Page 3
- Current Status – Page 4
- Tool Selection – Page 4
- Tool Development – Page 5
- Development Issues –Page 5
- Conclusion – Page 6

## Executive Summary

When working with something you've never used before or maybe something you don't like to use in its current state, it's always nice to have options. We've worked a lot with DD and this has been done at the command line, something many people like, but not all. If you don't like the command line but there isn't a GUI you are stuck, unless you want to make your own, which is what I did here. Although I do prefer the command line for its speed and feel. This Tool was developed to make using DD over a network connection with netcat easy to do. It may not be fully functional in the current release, but it should still suffice for basic needs and hopefully found to be useful.

The tool can be found at <https://sourceforge.net/projects/imageovernet/>

## Current Status

As of right now both the client and server are working at the most basic level. They are able to image with minimal options for DD and netcat. Some of the additional features of DD do work with the two programs, while others do need to be fixed. In particular the options in the programs with drop down boxes are not functional when piecing the DD command together; I need to figure out how to pull the selected choice. I also want to modify the way those options work so that user may select more than one option in some cases, like the conversion option. I also need to figure out how to kill the thread that calls the copy method without killing the whole program, until this is done the server must be terminated if one would like to change the options they made before starting it. There's a few small things that need to be fixed as well, like the exit button on the client, otherwise I would like to go through the code and clean it up a bit and maybe improve it. Also strange problem I've yet to figure out, trying to send large files results in DD not sending anything at all, the client reads 0 bytes. I think this may have something to do with the size over TCP and may just need to send small chunks.

## Tool Selection

When trying to come up with something that needed to be improved upon I was having trouble thinking of what I could do. All of the labs we complete required imaging or an image, the imaging was usually done via the command line and a much larger application. Instead of having to use a big application I figure a small easy to use program would be a good choice, not everyone likes the command line. Since I figured some may want to image over a network connect I decided to make a client and server imaging "package". My original idea was to have a server that multiple machines can connect to and get an image from via DD over netcat. This didn't work out so I made a server and client. The server, once started, will listen for a connection and once a client connects the image will be sent. I did manage to get the server to stay up it will handle multiple transfers.

## Tool Development

To begin making this tool I first set out to find a Perl GUI builder, something to aid with the GUI design as it has been a while since I've worked with perl Tk. I found a perl program call VisualCamel that did what I needed. So I used the program to make some labels and buttons and text boxes, but once those were created it was easier to just modify the values and base new stuff off of what I now had. Once I had the GUI setup a way that seemed nice I began work on the actual connection part of the client and server. This is something I had zero experience with, so I looked up a few examples of both and played around with them and testing them with each other and with netcat. This part took a long time to get working how I wanted it to, but eventually I got the hang of it.

So the GUI was setup and so was the networking portion, but the GUI really needed to be fixed. When first making the GUI I make labels and text fields for every option, which was useless if the server is only using input options, if any. At this point I pretty much deleted half of the GUI elements that were already created and organized so that only the input options were left. I then had to rearrange the elements again. That was just the server, up until this point I've been using a modified copy of the server for the client, mean I needed to redo most of the client GUI now. Finally got all of that figured out and now had a server and client with nice GUIs.

I spent a lot of time on the networking part of this project, a big part of that had to do with how to issue the DD commands when both the server and client were ready. I decided to send strings to verify when the server wanted to send and when the client wanted to receive. After these strings were sent/received on both sides the DD commands were executed via a system call. This also took a while because of small errors I was overlooking; it's been a while since I've done this much with Perl. The DD/NC command was eventually fixed as well.

## Development Issues

While developing the tool I ran in to multiple issues, the biggest is before submitting my project idea I looked at the netcat man page and saw the `-k` option; I planned on using this to make the whole server thing work, which doesn't make sense when thinking about it, or maybe it does and I'm more confused than I thought.

Pulling the value from drop down boxes has been kind of annoying to figure out, but as stating before I would like to change this later on. Either way it does not get the value as of this point.

When the server start button it clicked it calls a method that will start a thread that then calls a second method. This was done so that starting the server doesn't hang up the program, this way hitting stop can kill the server, which is where the problems lies. I cannot figure out how to kill the thread via the stop button without killing the program. I've never been the best with threading, so it may be easier than I'm making it. I've also never worked with threading in Perl.

## Conclusion

Working on this tool was kind of a small nightmare, but then so are a lot of the programming projects I've worked on. I would like to get some more stuff working on it as I believe it can be a useful tool. I happen to like using the command line more often than not, but a GUI is always nice, especially when it is something I have used or just haven't used in a long time. If I had to pin point something that was the biggest problem during development I'd have to go with lack of experience. I chose to do something that was more work than I thought it would be, especially at such a busy time in the quarter, but also included a lot of things I haven't worked with. But being that I got to learn some new things and the tool is functional I think this was a good experience in general.