8005 Asn3 - Testing & Documentation

Man in the middle your data for fun and proffit

Isaac Morneau; A00958405

README

Replicated below as shown on https://github.com/isaacmorneau/8005-asn3

README.md

8005-asn3

Building

the project uses a submodule for wrappers so it can be fetched at clone with

```
git clone --recurse-submodules https://github.com/isaacmorneau/8005-asn3.git
```

to keep the root folder orderly it is recomended to make from within bit as follows

```
mkdir 8005-asn3/bin
cd bin
cmake ../
make
```

Running

After it is built running the program with no arguments will give the following output. This page includes valid options as well as a couple examples

NOTE: `-p example.com@80:80` is functionally identical to ommiting the duplicate port such as `-p example.com@80`

```
usage options:
        -[-p]airs <address@inport[:outport]> - multiple flags for each forwarding pair
                -p example.com@3000:80 - redirects traffic locally from port 3000 to port 80 on example.com
        -[-t]cp - switch to tcp pairs(default)
        -[-u]dp - switch to udp pairs
        -[-h]elp - this message


example usage:
        -Forward ssh traffic from port 2000 to local port 22
        ./8005-asn3 -p localhost@2000:22
        -Forward http traffic to a different site
        ./8005-asn3 -p example.com@80
```

Figure 1

Build & Run

Steps

To build the project, run `cmake ../` inside of ./bin
After cmake generates the makefile run `make`

Expected Result

```
15:31:47(master)isaac@isaacbox:bin$ cmake ../
-- Configuring done
-- Generating done
-- Build files have been written to: /home/isaac/code/8005-asn3/bin
15:31:50(master)isaac@isaacbox:bin$ make
Scanning dependencies of target 8005-asn3
[ 20%] Building C object CMakeFiles/8005-asn3.dir/src/main.c.o
[ 40%] Building C object CMakeFiles/8005-asn3.dir/src/tcp_port_server.c.o
[ 60%] Building C object CMakeFiles/8005-asn3.dir/src/udp_port_server.c.o
[ 80%] Building C object CMakeFiles/8005-asn3.dir/src/wrappers/wrapper.c.o
[100%] Linking C executable 8005-asn3
[100%] Built target 8005-asn3
15:31:54(master)isaac@isaacbox:bin$
```

Figure 2

Run

To verify that it was all successful run `./8005-asn3` to print the usage message

Expected Result

```
15:31:54(master)isaac@isaacbox:bin$ ./8005-asn3
usage options:
        -[-p]airs <address@inport[:outport]> - multiple flags for each forwarding pair
                -p example.com@3000:80 - redirects traffic locally from port 3000 to port 80 on example.com
        -[-t]cp - switch to tcp pairs(default)
        -[-u]dp - switch to udp pairs
        -[-h]elp - this message


example usage:
        -Forward ssh traffic from port 2000 to local port 22
        ./8005-asn3 -p localhost@2000:22
        -Forward http traffic to a different site
        ./8005-asn3 -p example.com@80
```

Figure 3

Test Cases

Test	Steps	Result
Print Help	Run `./8005-asn3`	A help message about program usage is printed (Figure 3)
Forward SSH	• Run `./8005-asn3 -p <host>@8000:22` where <host> is the ip to connect to • Run `ssh localhost -p 8000` to connect to the forwarder	Ssh connects to the server via the forward (Figure 4)
Forward HTTP	• Run `./8005-asn3 -p example.com@8000:80` • Run `curl localhost:8000` to request the page	The server returns a document verifying the request went through (Figure 5)
Forward UDP hping	• Run `./8005-asn3 -u -p <host>@8000` where <host> is the ip to connect to • Run `sudo tcpdump -n udp dst port 8000` to display packets that are sent • Run `sudo hping3 --udp -V localhost -p 8000` to send the UDP packets	Tcpdump shows that the messages were sent (Figure 6)
Performance Test	• Run `./8005-asn3 -u -p <host>@8000` where <host> is the ip of the server • Run `iperf -s -p 8000` on the server • Run `iperf -c localhost -p 8000` on the forwarder	Iperf reports the bandwidth as 8.58 Mbits/sec (Figure 7)

Additional Figures

```
15:34:15(master)isaac@isaacbox:bin$ ./8005-asn3 -p ellybox@8000:22
link 8000 -> 22 at ellybox


15:44:22(master)isaac@isaacbox:~$ ssh elly@localhost -p 8000
The authenticity of host '[localhost]:8000 ([127.0.0.1]:8000)' can't be established.
ECDSA key fingerprint is SHA256:6o+DlDA4aq64I75trgNCghXjP4ufYpAQfbBAsMbql4E.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '[localhost]:8000' (ECDSA) to the list of known hosts.
Last login: Sun Apr  8 13:48:47 2018 from 192.168.0.5
15:44:46(master)elly@ellybox:~$ echo it works
it works
```

Figure 4

```
15:45:35(master)isaac@isaacbox:bin$ ./8005-asn3 -p example.com@8000:80
link 8000 -> 80 at example.com


15:45:32(master)isaac@isaacbox:~$ curl localhost:8000
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
         "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
        <head>
                <title>404 - Not Found</title>
        </head>
        <body>
                <h1>404 - Not Found</h1>
        </body>
</html>
```

Figure 5

```
15:43:08(master)isaac@isaacbox:~$ sudo tcpdump -n udp dst port 8000
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on enp3s0, link-type EN10MB (Ethernet), capture size 262144 bytes
15:51:30.852900 IP 192.168.0.5.8000 > 192.168.0.6.8000: UDP, length 0
15:51:31.853004 IP 192.168.0.5.8000 > 192.168.0.6.8000: UDP, length 0
```

Figure 6

```
15:51:32(master)isaac@isaacbox:~$ iperf -c localhost -p 8000
------------------------------------------------------------
Client connecting to localhost, TCP port 8000
TCP window size: 2.50 MByte (default)
------------------------------------------------------------
[  3] local 127.0.0.1 port 57420 connected with 127.0.0.1 port 8000
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0-10.4 sec  10.7 MBytes  8.58 Mbits/sec
```

Figure 7