pnetcat - a netcat(1) workalike in python

- What is netcat?
- How is pnetcat similar?
- How is pnetcat different?
- How does pnetcat get the performance it does?

What is netcat?

- Allows writing simple network programs from a shell like bash, tcsh or cmd.exe
- Can do TCP (reliable) or UDP (unreliable) sockets
- Can act as a client or a server
- Not great for back-and-forth, but nice for unidirectional communication

How is pnetcat similar?

- Allows writing simple network programs from a shell like bash, tcsh or cmd.exe
- Can do TCP (reliable) or UDP (unreliable) sockets
- Can act as a client or a server
- Not great for back-and-forth, but nice for unidirectional communication

How is pnetcat different?

- Different implementation language; netcat is in C while pnetcat is in Python
- The command line options are pretty different
- pnetcat is easier to modify because of its implementation language
- netcat is more security-oriented
- pnetcat is more performance oriented

netcat's security features

- Source routing
- Port randomization
- Source address spoofing

pnetcat's performance features

- Allows specifying a block size
- Allows specifying a TCP window size
- Allows turning off the Nagel algorithm

Performance comparison

- netcat – 499 seconds (without tuning)

```
•    netcat -v -l -p 5555 -q 300 > /dev/null
•    dd if=/dev/zero bs=1024k count=$((2**16)) | \
•      time netcat -v -q 300 localhost 5555 > /dev/null
```

- pnetcat – 326 seconds (without tuning)

```
•    pnetcat -l 5556 -o -w $((2**17)) -b $((2**17)) -N 0 > /dev/null
•    dd bs=1024k count=$((2**16)) < /dev/zero | \
•       time pnetcat -i -O localhost 5556 -w $((2**17)) -b $((2**17)) -N 0 > /dev/null
```

Adjusting the block size

- Just a matter of using appropriately sized read(), write(), send() or receive()
- Moderately large sizes allow CPU time to be used more effectively, because you aren't traversing the network stack as much
- Huge sizes slow down again, likely because of CPU caches

- input_socket.setsockopt(socket.SOL_SOCKET, socket.SO_RCVBUF, tcp_window_size)
- Allows more out of order packets to be received before packets must be dropped and retransmitted

Setting the Nagel algorithm

- input_socket.setsockopt(socket.SOL_TCP,socket.TCP_NODELAY, tcp_nodelay_flag)
- Turning off Nagel disables pausing a while to see if more data is coming before transmitting