Socket Intro

Bottom Line Up Front

Please have a **terminal** available. linux / osx is what I use and it works well. If you use windows, `cmd` wors for me on my virtual machine.

Install python package `pwntools`:
    `pip install pwntools`.
(in your interpreter you `import pwn`)
If your machine has trouble installing this, let me know and I'll provide a prime on using the `socket` package instead.
But pwntools is simpler to use.

Install python package `json`.
    `pip install json`
(in your interpretor you `import json`)
We will only use the dump string and load string functions `json.dumps()` and `json.loads()`.

Install tools: `nslookup`, `curl`

The Lab Intro

In this Lab we are going to cover some basics about how communication with a server takes place.

This week's lab will setup future labs where we will be able to work with more realistic protocols and attacks. The intent here is to see in a 'hands-on' way how what we have been studying is used.

This is the first socket lab and I want to make the transition as easy as possible. Therefore, I wanted to lay out how these challenges will work, discuss the tools you will use, and explain the communications procedures.

What is a Socket connection

A **socket** is specifically an IP address combined with a port. directly talk to some simple servers I've created.

An **IP address** is used to identify a computer. If I want to communicate with another computer I need to be able to reach out and identify it. An IP address does just that.

A **port number** is a value from 0-65535, which is appended to an IP address. The port is simply an agreed upon path for communication.

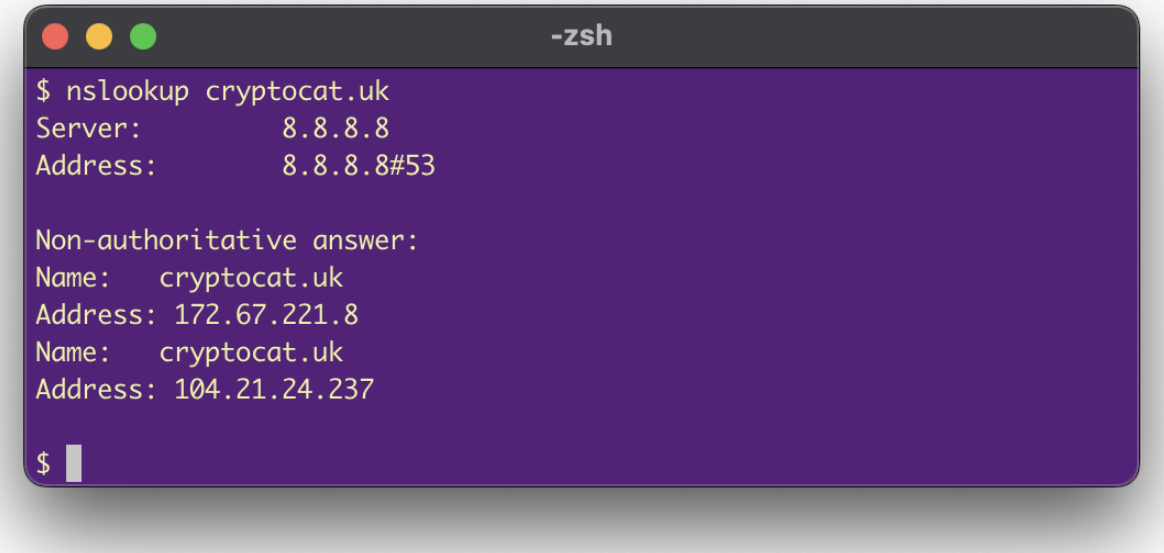
Let's look at IP Addresses more closely

IP address is the routing information which allow one computer to find another computer. I will focus on webservers and the websites they host for now.

A **DNS** server is a translation service which translates between human readable text and IP addresses. If you want to navigate to `https://www.cryptocat.uk`, you type that into the navigation bar.

Your webbrowser then contacts a *DNS* server... commonly `8.8.8.8` the google DNS. The DNS server returns an IP address, which tells your computer how to address its communication.

This is what it looks like.
I'm using a command called name service `nslookup`, to see where our class websites reside.



```
$ nslookup cryptocat.uk
Server:         8.8.8.8
Address:        8.8.8.8#53

Non-authoritative answer:
Name:   cryptocat.uk
Address: 172.67.221.8
Name:   cryptocat.uk
Address: 104.21.24.237

$ 
```

This should work on most of your computers, this command is available in linux, osx, and windows cmd. You can see the name lookup went to 8.8.8.8, google dns, and translated `cryptocat.uk` into IP addresses.

TRY THIS:

See what you get when you run nslookup in your own terminal. Often it doesn't go to 8.8.8.8. There are different configurations. If you are on a router or a virtual machine you may not be able to access this service so directly.

TRY THIS:

The the IP address, 104.21.24.237 into your browser navigation bar. Often times you can go directly to a website. I have high security on `cryptocat.uk` so cloudflare should block the immature header.

If you want you can run nslookup on my socket connections link, `socket.cryptocat.uk`. This should provide you a direct connection to my webserver (not running through Cloudflare security). It may be instructive.

Let's look at port numbers more closely

Now that we have found the computer, via IP address, we wish to talk to. We need to communicate in an understandable way. Ports allow computers to agree on a specif method of talking. Most protocols use specific, pre-arranged, ports. Hopefully you have heard of a few of them.

port 22 --> ssh
port 80 --> http
port 443 --> https
port 3389 --> remote desktop

But while these are the standard ports for these protocols, we can use anyport we want, we just have to know what we are looking for. We will do some of this in the lab.

put it all together in a socket

A socket is an IP:port combination.
Identically a socket is a url:port combination, since we know we use DNS to translate a url to an IP.
For instance a websocket can be https://www.cryptocat.uk:443 which is standard secure webtraffic.

Say you have a raspberry pi in your home network located at 192.168.1.5.
If you SSH into your pi you would point your ssh client to

Some tools

We have used a terminal to view `nslookup` already.
I want to add a few more tools to think about.

`curl` is a command, client url.
This allows us to get an html document via text.
It is the command your browser is using on the backend.
We will be using

`json` is a python package which takes strings and bundles them into json formatted objects. The huge advantage of the json package is that it will create json formatted objects from variables, so there is no need to hard-code variable values, which can be extra work when dealing with these challenges.

`socket` and `pwntools` provide functions for creating input / output connections with socket servers.
Please read the full getting started tutorial.