Capport Hackathon Results

What we did

- Implemented most components of the architecture:
  - ICMP
  - Captive Portal Enforcement
  - CAPPORT API
  - Part of the DHCP Server (RFC 7710)

ICMP

- Implemented in Client
- Implemented at Captive Portal Enforcement
- Started trying out new version of ICMP draft

API

- Ran on Captive Portal Enforcement Server
- Accepting logins/logouts with a crude password schema.
- Three steps:
  - Get info from capport api
  - create session
  - authenticate using info resulting from created session

DHCP/RA

- Implemented the server side of RFC 7710 for DHCP
  - didn't get around to client side ☹

Tcpdump and wireshark

- Added support for new ICMP messages

What did we learn?

- Need to update architecture to make use of new ICMP message.
- Fairly simple to implement ICMP handling in Linux userspace
  - Not sure about other operating systems
- We were able to break the project into parallel tasks nicely.
- REST API easy to program to

Lots of Good Discussion

- Some discussion about why the API would be necessary:
  - IoT
  - People unable to use a standard browser UI
- Discussion about making portions of the architecture optional:
  - you get something useful with only some of the components?
  - If so, could be nice for building up support for the architecture
  - What happens if a DHCP server isn't available, but using IPv4? (e.g. an LTE network)
- Discussions about simplicity of the components
- Captive Portal Enforcement device policy needs to identify based on something it has access to in the packets.
  - What if this differs from the user equipment knows about?
  - E.g. NAT in the way
- Talked about putting the identity provided by RFC 7710 into the URI -- simplifies the client. Less state/it doesn't need to know what information is important
- Important for the solution to support IPv4 *and* IPv6
- An option for first logging in: send a message and have it fail, then get access, or log in preemptively when detecting API server. Up to implementation?
- David Bird had an idea to use the CAPPORT ICMP message to provide feedback to users (e.g. colour their wireless connectivity icon) to indicate when they are reaching the limits of their service.

Next Steps

- Use the code to continue to prototype the drafts as they progress
- Any other ideas?

Questions?