ECE 435 Final Project (Sample)
Apple II Webserver

Vince Weaver

http://www.deater.net/weave/vmwprod/apple2_eth/

vincent.weaver@maine.edu

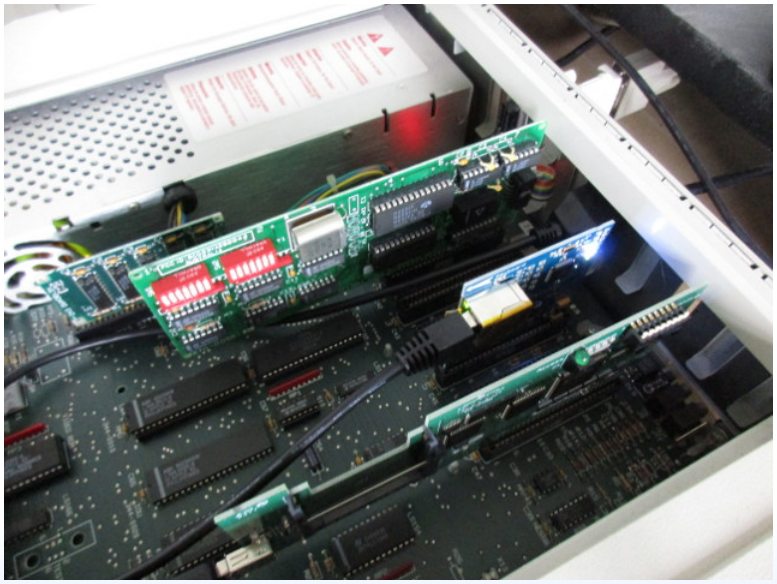
30 November 2017

Background

- Apple II hardware, originally from 1977 (this IIe 1987)

System Setup

- 1MHz 6502 processor, 64kB RAM (well, 128kB, it's complicated)
- Uthernet ethernet Card



Related Work

- Popular to retro-compute, try modern things on old machines
- Many just dump raw HTTP over serial port, use helper machine to handle Ethernet
- Often webpages are hard-coded rather than read arbitrary files off disk
- Unusual to use BASIC

Features

- Serve html, txt, jpg or png files. Also supports favico.ico
- Properly returns 400 and 404 status messages
- RFC 2324 compliant
- Tested to work with Firefox and wget, in theory should work with any browser.

Uthernet Ethernet Card

- Wiznet 5100 ethernet chip
  - standard ethernet card: program with MAC address, it watches for frames with it, copies it to memory, sends an interrupt
  - You can run in this manner, but only so much memory on Apple II, also interrupt support horrible.
  - Would have to program ARP, TCP/IP yourself. This is possible (people have done it) in assembly language
  - Chip also supports hardware stack. You program MAC,

IP, and then you get up to 4 sockets. Handle all of TCP/IP for you. Setup IP parameters,

- ◦ You can get interrupt (or poll) when packet comes in and the data is in a circular buffer

Software Setup

- BASIC – common on 1980s micros, interpreted language
- Wrote simple web server in C first, few hundred lines of code
- In BASIC use PEEKs and POKEs to initialize card (MAC address, socket listening on port80)
- Waits for connection, gets HTTP packet
- BASIC parses strings, reads files from disk
- Writes resulting file back out
- Prints debug to screen

Sample Code

```
1200 REM *** LOAD FILE
1202 X$=RIGHT$(N$,3):M$="text/html"
1203 IF X$="txt" THEN M$="text/plain"
1204 IF X$="png" THEN M$="image/png"
1205 IF X$="jpg" THEN M$="image/jpg"
1206 IF X$="ico" THEN M$="image/x-icon"
1207 IF N$="teapot.html" GOTO 9000
1208 ONERR GOTO 8000
1209 PRINT "LOADING ";N$
1210 PRINT CHR$(4)+"BLOAD ";N$
1215 POKE 216,0: REM CANCEL ONERR
1220 FS=PEEK(43616)+256*PEEK(43617): REM FILESIZE
1225 PRINT "DONE LOADING"
```

Network Layer being Used

- Mostly socket programming, so Primarily at the Application / Transport Layer

Security Concerns

- Easy to DDOS
- Behind firewall
- Limited utility in getting ”root” in an Apple II running BASIC

Challenges

- BASIC is slow. Memcopying by PEEK/POKE
- https or HTTP/2 not really possible

Future Work

- Use inline assembly routine for PEEK/POKE
- Also try to do minimal ARP and TCP/IP stack
  That might require assembly language

Demo/Questions

- Was a featured project on Hack a Day:

  `https://hackaday.com/2016/12/17/apple-ii-web-server-written-in-basic/`
- See also a video:

  `https://www.youtube.com/watch?v=qmIC1rImhU4`