ECE 435 – Network Engineering
Lecture 19

Vince Weaver

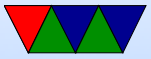
https://web.eece.maine.edu/~vweaver

vincent.weaver@maine.edu

9 March 2026

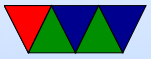
Announcements

- HW#6 was posted
- Midterm on Wednesday March 11th

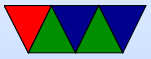


HW#2 – Programming Notes

- Watch warnings, though I might be running newer version of gcc
- Don't use string operations on binary files
- If no file specified, index.html If no index.html send a 404 error
- ctime prints own linefeed
- If you report HTTP 1.1, don't close connection after file, there might be more requests and you might get "connection reset"

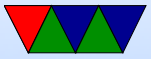


- Be sure to check for unexpected errors – what if huge URL is sent?
- Many crashed if I requested the `README` file. Have to handle unexpected input from user. (in this case, no file extension)
- Traditionally the biggest problem (if the browser refuses to display) is the wrong Content-length:
  If you send less data than you say you will, it will wait forever for it, or else give a "connection reset" if you close the connection.
- Be sure you read everything the browser is sending



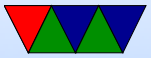
(Either big enough buffer, or repeat in loop reading it all). If you send a response before it is done sending it can confuse things. How can you hold an arbitrary size header? malloc()? Do you want to?

- Be sure to drop the leading / in the file part of a URL
- If you use firefox you'll see it might also request favico.ico? Why? What should you return (assuming the file doesn't exist?) 404.



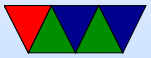
HW#2 – Why write web server in C?

- A pain to write in C.
- But... what language are most webservers written in? Apache=C, nginx =C, lighttpd = C, litespeed = C

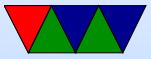


HW#2 – Questions

- browser
  - Error 404 – not found
  - Error 418 – RFC 2324 coffee protocol (I'm a teapot)
  - Error 451 – Unavailable For Legal Reasons / Ray Bradbury
- http header from www.maine.edu
  - nginx/1.20.1
  - Isn't actually a website, just redirect to the https site
  - Old days they ran Apache 2.2.2.

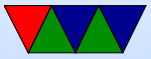


- Re-ran things this year and more complex so left last year's, for some reason now there's a big chunk of javascript



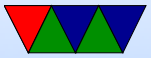
HW#2 – Something Cool

- I do appreciate the pages you made, even if I didn't comment specifically in the grades.

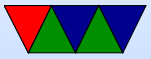


HW#3 Review

- md5sum/encryption, some people had odd md5s?
- How to validate PGP key is indeed for who it says?
  You have to trust someone...
  - Do you trust a website you find? What if it's fake? What if it's `https`?
  - Certificate Authority (costs money)
  - Distributed Web of Trust (key signing party).
  - Compare in person/phone, key fingerprint if not want to send whole thing

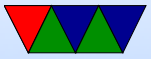


  - Can you trust phone/video calls anymore?
- Encrypted message went fine
- Why not use SHA-1 for git anymore? It's been "broken" which means possible to generate a collision
- umaine website certificate
  - aside: it stores 107k of private data on you?
  - Internet2 certificate, Public key 4096bit RSA, valid for a year, possibly SHA-256 ECDSA
  - That was signed by Incommon RSA, certificate valid for 10 years
  - That was then signed by Usertrust, good until Jan



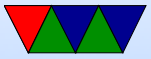
2038 (!?)

- md5sum extra credit, people did get some collisions. Though birthday and/or chosen-prefix of course

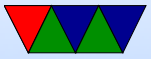


HW#4 Review – E-Mail headers

- First warning sign – says its from a bank, but the return address is from a Florida dental school
  Also not a bank of mine
- encrypted and verified from UFL, but sent from videotron.ca cablemodem
- Virus scanned and SPAM scanned, just sort of barely passed
- pop from deater.net via fetchmail (this isn't suspicious, it's the sender not receiver you have to look at)

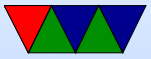


- LMTP – local mail transport. LHLO. No mail queue, says right away whether deliver mail is possible.



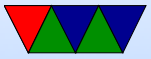
HW#4 Review – E-Mail headers / PDF attachment

- pdf attached probably had some sort of exploit or phishing document. Didn't open.
- Can a PDF compromise your system? Modern browsers will open in-browser
- Note, the attachment being listed as "Application" does not mean it's an executable
  - Just the first part of the mime type
  - If it was an executable, is that an issue?



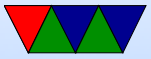
HW#4 Review – E-mail Phishing/Ransomware

- Should you trust e-mail?
- IT constantly trying to warn about Phishing attacks
- "Rick Astley has done more to raise awareness of clicking random links than any anti-phishing initiative"



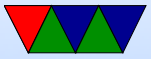
HW#4 Review – E-Mail jpeg attachment

- was looking for MIME as what's going on
- Also was looking for base64 as the encoding



HW#4 Review – DKIM/SPF headers

- Trying to avoid spam
- These here from my personal e-mail to umaine
- My hosting provider has working SPF but DKIM gives "permerror"

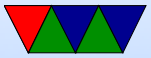


HW#4 Review – Domains

- maine.edu created
  - 2 December 1988
  - Fairly typical. Early schools 1985 or so
- Registrar is EDUCAUSE

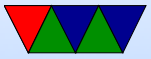
  What is a Registrar?

  Note: registrar, not registrant



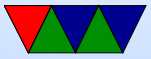
HW#4 Review – DNS

- A 130.111.218.23
- AAAA 2607:f8b0:4006:802::2004
- NS nameo.unet.maine.edu / namep
- MX ALT4.ASPMX.L.GOOGLE.COM
  - Also for MX have a priority value
  - Why google? Running mailserver difficult, at some point all universities let google/microsoft take over. For various reasons you gave, but also likely hoping to lock you into gmail.



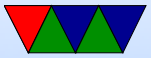
HW#4 Review – DNS Security

- Still various issues



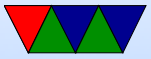
Brief HW#5 Review

- source/destination/size/checksum
  - src: a9a0 = 43424 (note, hex dumps are naturally big endian)
  - dest: 35 = 53 (DNS)
  - size: 2a = 42 bytes
  - yes checksum (note: 0000 means no checksum. ffff is a valid one)
  - protocol is DNS (how can you tell?)
- Why use UDP vs TCP



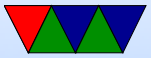
lower latency, lower overhead (no need to handshake), simpler

Be careful just saying “faster”, need to explain more what you mean by that.



HW#5 Coding Notes

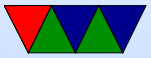
- Remember to comment your code!
- Getting source port from incoming connection
- Note this is not the IP address
- Getting it from the struct is sort of hard
- Also remember it's in network endian, need to convert with `ntohs()` In general would be an ephemeral port above 40000



IPv4 Fragmentation

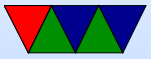
- Complex solution to problem where varying routers might support different maximum packet sizes
- Useful IP Fragmentation article:

  `https://lwn.net/Articles/960913/`



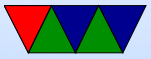
IPv4 Packet Fragmentation

- Ethernet MTU (maximum transmission unit) 1500 bytes but IP MTU is 64k, so must break up larger packets
- Can be further broken up depending on MTU along way
- Final destination is responsible for reassembling
- Can mark packet "do not fragment". What happens then if too big? (dropped, ICMP error sent)
- All fragments have same ID/sequence number. Last fragment marked with 0 for "more fragments" flag. Position from fragmentation offset field



IPv4 Packet Fragmentation – Example

- Example: original, 3200 bytes of data
  remember, offset is multiplied by 8
  Unclear how you pick the id value (random?)
  - header id=x, more=1, offset=0, 1480 bytes
  - header id=x, more=1, offset=185 1480 bytes
  - header id=x, more=0, offset=370 240 bytes
- Each fragment is a valid IP packet

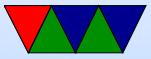


Fragmentation Limits

- RFC 791 (1981)
- IPv4 Receivers must be able to handle fragmented packets with total re-assembled size of up to 576 bytes (modern OSes can generally handle up to 64k)
- IPv4 packets under 68 bytes can't be fragmented
- Picking the id/sequence number is complex see

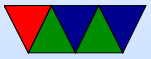
  `https://crnetpackets.com/2015/08/29/a-short-story-about-the-ip-id-field/`

  (people wanted to re-use ID field for de-duplication but RFC 6864 says if DNF set you must ignore ID)



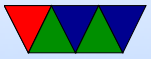
Problems with Fragments

- no way to notify other side of missing fragments
- last fragment is usually short (wasting resources)
- receiver must hold in RAM fragments to be reassembled.
- can DoS by sending lots of fragments but none complete
- fragments have no TCP/UDP header, firewall can't easily filter
- Most modern implementations set DNF on TCP connections and instead rely on path-mtu-discovery
- `https://blog.cloudflare.com/ip-fragmentation-is-broken/`



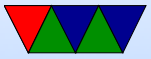
Path MTU Discovery

- Automatically determine the MTU (max transmission unit) between hosts
- Originally for routers, now also for endpoints
- Process
  - Set DNF bit on packets
  - Any router where packet size too big drops packet and sends back error via ICMP
  - Source reduces MTU and tries again until it gets through

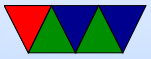


Path MTU Issues

- If MTU gets smaller, will get noticed and can adjust. No way to easily find if MTU gets bigger
- When packets encapsulated/tunneled inside of another protocol an extra header is added, which can kick things above the MTU threshold.
- Complete 3-way handshake can happen (small packets) but then drop all actual traffic. "black hole connection"
- Why would ICMP be blocked?
  - Over-zealous sysadmin



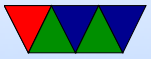
- Traffic load balancers have to keep all TCP packets on same machine, but when ICMP comes in it's not always clear who it belonds to



Handling if MTU discovery blocked

- Various workarounds for this. Force MTU to be Ethernet everywhere? Use TCP to probe size, treat packet drops as MTU issue not congestion?
- Interesting article

  `https://blog.cloudflare.com/path-mtu-discovery-in-practice/`



Security Issues with Fragments

- ICMP/UDP larger than MTU, cannot be reassembled
- TCP "Teardrop" attack, send fragments with overlapping offsets, confuse/crash machines
- Fragments can be constructed to obscure malicious text