

ECE 435 – Network Engineering

Lecture 8

Vince Weaver

<http://web.eece.maine.edu/~vweaver>

vincent.weaver@maine.edu

10 February 2022

Announcements

- HW#3 due Friday
- HW#4 will be Posted
- HW#2 mostly graded, will be sent out soon

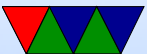


HW#2 – Programming

- Be sure to check for unexpected errors – what if huge URL is sent?
- Time Wait is always a pain
- Don't ignore compiler warnings!
- 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
- Many crashed if I requested the `README` file. Have to handle unexpected input from user. (in this case, no file extension)
- A pain to write in C. But... what language are most webservers written in? Apache=C, nginx =C, lighttpd



= 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
 - nginx/1.16.1
 - Isn't actually a website, just redirect to the https site
- Can you connect with telnet? No, way more complex,



even discounting encryption



HW#2 – Something Cool

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



POP/IMAP

- POP (post office protocol) RFC 1939
 - download mail to local machine which handles
 - port 110
- IMAP (internet message access protocol) RFC 2060
 - manipulate mail on server
 - gmail can present as IMAP. tags are really imap “folders”. Can actually download local (I do).



e-mail body

- What happens if try to start line with “From”? Try it.
- Useful to check headers for things like SPAM, phishing attacks
- Signatures (4 lines/80col?)



SPAM/other

- In early days, “open relays” if an e-mail came in the server would take mail from anyone and try to deliver it to anyone. Not a good idea (spammers)
- Origin of term SPAM?
First commercial spam March 5, 1994 Law Firm, Green Card Spam
- Spam/Virus filtering (joke of getting viruses via e-mail)
- procmail sorting
- mail spoofing (What’s to stop you from putting someone



else's address at FROM? how can you catch this?)



SPAM countermeasures

- On the sysadmin side, make sure systems are secure. Many ISPs block outgoing port 25
- SPF records in DNS, say which machines in your network are allowed to send e-mail. Downside, if user has bought a domain and uses it but the ISP doesn't support SPF.
- Not posting your e-mail, intentionally mixing up your e-mail so address harvesters have trouble getting it. Downside? Things like + in e-mail address?
- Challenge/response. Need to ACK before e-mail goes



through. No one likes this.

- DNS black lists, lists of known spamming sources
Some people block whole countries or all cable-modem connections
- Strict SMTP implementations. Spammers don't always implement their mail senders well.
- Greylisting – delay delivery of the mail by a few minutes (with a 400 response). Most legitimate servers will retry, a lot of spam software doesn't bother.
- Filtering, blocking keywords/all-caps
False positives?



- e-mails with chunks of books in them, crazy characters
- Bayesian filtering – auto learning. Sometimes can see this in headers



e-mail security

- SSL encrypted connection to SMTP server (usually plain text) SSMTP
- SMTP end to end still unencrypted
- Can use PGP (pretty good privacy) to encrypt e-mails, practically no one does this



Can you run your own e-mail server

- Used to be possible/common
- Common e-mail servers: configuring them, specifically SendMail



Other common protocols we won't cover

- Legacy (inetd): echo, chargen, discard, time, finger (.profile, .plan), qotd, systat, write, talk
why no longer supported? security? lack of interest?
- Messaging:
 - IRC – internet relay chat
 - AIM/ICQ/MSN etc
 - unix talk/write
 - MUDs, talkers
- IPP – printer protocol (CUPS, lpd, jetdirect)



- backup software
- syslog
- telephony
 - skype
 - facetime
 - VOIP
 - ASTERISK
- ntp – network time protocol
- LDAP/Authentication
- Network Attached Storage/Fileservers
 - NFS



- Samba/CIFS
- andrewfs (afs)
- Databases: mysql
- Distributed/Torrent sites
- Distributed computing (SETI@Home)



Domain Name System (DNS)

- Why do we need it? Send e-mails to vince@192.168.8.1?
What if server moves?
- Hierarchical distributed database
- RFC 1034, 1035 (1987)
Supersedes RFC 882, 883 (1983)
- Maps hostnames to IP addresses



Ancient History

- In early days NIC.arpa has a “HOSTS.TXT” file you downloaded occasionally with all known machines. Didn't really scale.
- `/etc/hosts` is a relic of this, usually checked first
- On Linux this is configured via `/etc/nsswitch.conf`



Domain Names

- Which ones can you name? .com/.org/.gov/.edu/.net/.mil
- Country codes (.us/.uk/.ie etc)
- Huge expansion in the last few years (.horse)
- Owner of a domain can subdivide, i.e. eece.maine.edu
- How do you buy them? Used to be fairly expensive and only for two years at a time from a single registrar. Not so much anymore.
- whois will show you info on who owns



Name Rules

- Can have 127 levels, each 1-63 chars.
- Usually total name cannot exceed 253 chars.
- LDH (letters,digits,hyphens, cannot start with hyphen, not all numbers)
- Case-insensitive
- International names: “punycode”. Trouble, why?
Foreign letters that look like ASCII ones.
- punycode – snowman example <http://xn--n3h.net/>



- First commercial name 15 March 1985 symbolics.com
example.com set aside (why be careful with your example names?)
- Shortest? g.cn. Various one-letter domains (like x.org) but they were later reserved.
- Typosquatting, domain squatting, copyrighted names, etc.



DNS Server

- Listens on port 53, usually UDP
(Special case if > 512 bytes: use TCP)
- Zone records
- 5-tuple, NAME TTL CLASS TYPE VALUE
 - TTL (how long to cache)
 - Class (usually IN for internet)
 - Type and RDATA (resource data)
 - Common types
 - SOA – start of authority (parameters) primary source,



e-mail of admin, etc

- A – IPv4 address of host (32bit int)
linux.deater.net 86400 IN A 1.2.3.4 can have multiple and be cycled through round-robin
- AAAA – IPv6
- MX – Mail exchange (can have multiple, specify priority)
- NS – name sever (name server for this domain)
- CNAME – Canonical name, allows aliases
- PTR – alias for IP, for reverse lookup
4.3.2.1.in-addr.arpa



- HINFO – cpu and OS type (text) (uncommon)
- TXT – raw ASCII text
- SRV – new – sort of generic version of MX
- SPF – which machines can send e-mails (avoid spam)
- Can you store other things in records? Text adventure?
File transfer? Tunneling (iodine?)



DNS client

- Name resolver, translate from ASCII (still?) name to IP addr



DNS Lookup Example

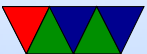
- Basically: application calls a library (resolver) with the hostname.
gethostbyname() in socket examples
This sends UDP to local DNS server, which figures out the address (possibly recursively) and returns the address to caller.
- Details
 - First check `/etc/nsswitch.conf` which might say to check `/etc/hosts` and maybe NIS/LDAP first



- Query via UDP local nameserver (`/etc/resolv.conf`)
- If the local is the official nameserver, get *authoritative response* (from responsible zone)
the alternative is a cached response
- If local DNS server doesn't know about it, it has to ask up the chain.
- If not known, query “root” server. So if looking up `weaver-lab.eece.maine.edu` will ask root, which will direct to `.edu` DNS server
- 13 root servers, mostly in US
Single-letter server names, limitation of number that



- can fit in single 512B UDP packet
- Recursive query It will not likely know but will know about maine.edu, so ask that one, which will ask eece.maine.edu, etc, and then passed back
 - Result is then cached along the way (TTL) caching up to 68 years (or none at all). Why low values? Why can that be bad?
 - Caching also means usually the root server does not have to respond to each request
 - As the response is passed back through it will be cached along way.



DNS Query with dig – dnsutils

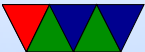
```
dig weaver-lab.eece.maine.edu
```

```
; <<>> DiG 9.11.3-2-Debian <<>> weaver-lab.eece.maine.edu
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 1268
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 512
;; QUESTION SECTION:
;weaver-lab.eece.maine.edu. IN A

;; ANSWER SECTION:
weaver-lab.eece.maine.edu. 3599 IN A 130.111.218.24

;; Query time: 79 msec
;; SERVER: 8.8.8.8#53(8.8.8.8)
;; WHEN: Tue Sep 25 14:12:42 EDT 2018
;; MSG SIZE rcvd: 70
```



Reverse DNS request

- Given IP address, how can you find the name?
- Linux can use the “host” command.
- For IPv4, there is special in-addr.arpa domain
- To look up 1.2.3.4, lookup 4.3.2.1.in-addr.arpa
- It will iterate down. This gets trickier now with non-contiguous IP allocations.
- Similar thing for IPv6 using ip6.arpa



DNS Packet format

- Packet format

16-bits	16-bits
ID	Control Flags
Query Count	Answer Count
Authority Count	Additional Count
...	
...	

- Flags

- QR – request (0) or response (1)
- OpCode – QUERY, IQUERY, STATUS, NOTIFY,



UPDATE

- AA – Authoritative Answer (1) or cache (0)
- Truncated – (1) means too big for UDP
- RD – Recursion Desired
- RA – Recursion Available
- Z – zeros (reserved)
- AD – Authenticated Data (DNSSEC)
- CD – Checking Disabled (DNSSEC)
- RCODE – Error Code
- Counts say how many of each included
- Then the actual requests



Other DNS Notes

- Zone transfers – copying zone list between machines
- Can also do non-recursive (iterative)
- breaking out of circular queries
- BIND/named
- dig / nslookup tools

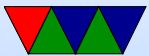


DNS Security

- cache poisoning attack – pretend to be authority, poison with wrong results
- DNSSEC
 - RFC 3833
 - Digitally sign response
 - Can provide things like public keys
 - Backwards compat
 - Slow uptake
- 0x20 encoding, toggle case in QNAME for extra bits in



ID



DNS Privacy

- Can people spy on your web-browsing through DNS?
- Can a web-browser tunnel DNS over https?
- 1.1.1.1 and 8.8.8.8 name servers?

