ECE 435 – Network Engineering
Lecture 36

Vince Weaver

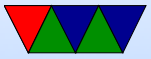
https://web.eece.maine.edu/~vweaver

vincent.weaver@maine.edu

25 April 2025

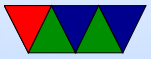
Announcements

- Final is Monday, May 5th 10:30am, here
  Will review for it on Monday
- Will go over HW#11 Monday
- Some presentations on Monday
  If you are ready to present on Monday let me know, there is room (sorry for the confusion on that)

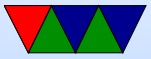


Sample Project Presentation

Applesoft BASIC Webserver on 8-bit Apple II

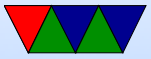


Network Security Continued



General Security Issues

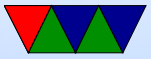
- Malware
- Virus
- Worms (spread over network)
  Famously the Morris Worm (1988)
- Trojan Horses



Application Layer

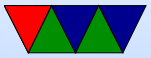
- Web-browser
  - cross-site scripting/XSS

    Inject javascript from other sites into that being displayed, runs with permissions of document
- e-mail (can you get a virus over e-mail?)
  - Phishing
  - Ransomware



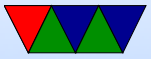
ssh security

- Fail2ban
- Nonstandard port
- Port knocking
- Call asterisk for one-time pin?
- No-password (key only)
- LCD device



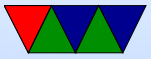
Fuzzing

- Searching for issues by sending random (or almost random) inputs and see what happens



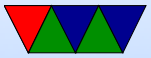
Mitigations

- DoS
  - blackholing/sinkholing. Send all traffic to non-existent server
  - firewalls
- Weak passwords
  - Password rotations, strength (downsides of this?)
  - Using public key instead of password
  - Two-factor Authentication



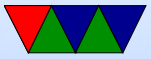
VPN/Tunnel

- Create a tunnel, TCP/IP inside of TCP/IP directly from your machine into remote network (past firewall) or network-network.
- Link layer tunnel – all Ethernet packets go through as if were local
- IPSEC – IP level tunnel, IP in certain range (or all) go through the secure IP tunnel to other side



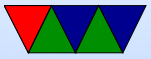
Firewalls

- Runs on machine, intercepts all incoming packets before allowing them through.
- packet-filter based – looks at layer3/layer4
  fast because addr/port fixed locations
- application-gateway – looks into protocol
  may be a proxy server (so can do things like filter http requests to certain websites)



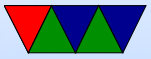
Firewalls

- 1st generation – packet filtering. Check for port number or IP destination and drop if not OK
- 2nd generation – stateful firewall. Keep a packet history so it can make decisions based on state of connection (new connection, existing connection, etc)
- 3rd generation – application level. Can understand protocols like ftp, http, etc, and make decisions

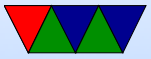


Deep Packet Inspection

- Can be used to block viruses and such, but also censorship
- WAF (Web-application Firewall) at CDNs that blocks certain keywords found in attacks, but can block legitimate users who are maybe just trying to talk about it (for example, block any html with “virus” in it)
  some insurers force companies to use them, can lead to hard-to-debug networking issues when traffic dropped because of it

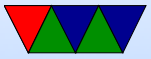


- Encryption can help against this (for better or worse)
- Organizations can MitM you by forcing CA authority that lets them decrypt your connections at the network border
-



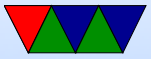
Firewall – Configuring

- eBPF – extended Berkeley Packet Filter
  - Filter can be written in high level language and compiled and inserted into kernel, faster than scripted
  - Special language, deterministic finish, limited looping, etc
  - Linux used for things besides filters these days



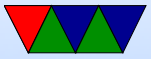
Organizational Firewall Setup

- firewall to outside, extra DMZ layer where any servers might be, then an additional more restrictive firewall to internal network.
- DMZ – Demilitarized Zone, part of network infrastructure separate from the internal parts, used for untrusted machines or machines needing to talk to outside world (webservers? guest wi-fi?)
- why? if servers compromised don't want free reign over rest of network.

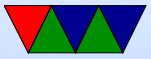


Over-Reliance on NAT

- To do NAT you use firewall technology. Monitor incoming packets. Only allow those from existing connections. Port forward to inside.
- Many people have a router implementing NAT and so get a fairly strict IPv4 firewall "for free"
- A firewall doesn't have to do NAT
- IPv6 doesn't need a NAT, and so you need an IPv6 firewall if you want to be as safe
- People forget this and so IPv6 networks can be less

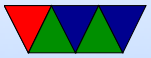


protected



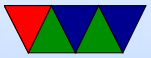
iptables

- Linux changes up firewall interface all the time
- ipfwadm (linux 1.2 - 2.2)
- ipchains (linux 2.2 - 2.4) stateless
- netfilter/iptables (2.4) – stateful firewall
  can filter on lots of things. BPF filters
  NAT is done via this
  port forwarding
  had 4 separate engines (ipv4, ipv6, ethernet, arp)
- nftables (linux 3.13) – merges things, virtual machine



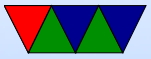
(but not BPF) to speed things up

- Separate ip6tables utility for setting IPv6 rules
- Also arptables/ebtables for filtering ethernet



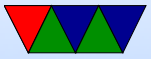
iptables example

```
# Flush all rules

iptables -F
iptables -t nat -F
iptables -t mangle -F

iptables -A FORWARD -i eth1 -o eth0 -m state --state ESTABLISHED,RELATED -j ACCEPT
iptables -A FORWARD -i eth0 -o eth1 -j ACCEPT
iptables -t nat -A POSTROUTING -o eth1 -j MASQUERADE

echo 1 > /proc/sys/net/ipv4/ip_forward

iptables -t nat -A PREROUTING -p tcp -i eth1 --dport 2131 -j DNAT --to-destination 1
iptables -A FORWARD -p tcp -d 192.168.8.18 --dport 22 -m state --state NEW,ESTABLISH
```



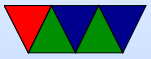
encryption problems

- Keys leaked (DVD/game console issues)
- poor random numbers used (Debian problem)
- differential cryptanalysis (start with similar plaintexts and see what patterns occur in output) [DES IBM/NSA story]
- Power/Timing analysis – note power usage or timing/cache/cycles when encryption going on, can leak info on key or algorithm



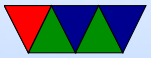
XZ Vulnerability Case Study (2024)

- Was in the news. Computer security but relate to stuff in this class.
- Problem with open source software and trust. Can you trust random contributors? Can you trust random tools? Reflections on Trusting Trust.
- ssh vulnerable. Maybe only thing you let through incoming to your firewall
- Because of this ssh much scrutinized for bugs
- What if some other code unrelated can take over ssh?

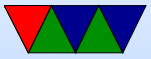


XZ Case Study (bg)

- Supply chain attack?
- What if someone pretened they had a bug fix but instead it introduced evil code?
- Changeset that is like if (!strncmp(passwd,"mypassword"))
- In theory easy to spot, lots of reviewers, benefit of open source everyone can see in the open, run tools
- What if try to be sneaky about it?
- if (uid=0) printf("Error!\n");
- Underhanded C contest

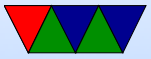


- U of Minnesota Linux kernel incident. IRB. No justice.

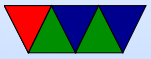


XZ Case Study (how it happened)

- XZ issue. Problem not with Linux, or ssh. Was with xz compression library.
- Maintainer volunteer, overworked, someone showed up offered to help, gradually gained trust. Eventually given commit privileges.
- They started making seemingly innocent changes
- Broke some of the hardening tests in autuoconf, including adding a hard to see "." so the test for it would always fail

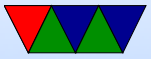


- No actual code added to C code, but part of build process it would take some of the files from the test suite and patch the binary



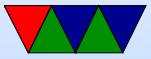
XZ Case Study (details)

- original calls `__get_cpuid` at library start to see if can use CLMUL instruction
- adds a `_get_cpuid` with one underscore to do sneaky stuff
- systemd on Linux links against this. The library loaded by systemd before launching sshd. Would override some symbols (complicated linker stuff) but would override RSA key checking
- When the certificate came in with the connection, if it



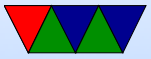
decompressed with the key from attacker then treat it as a binary and run it

- This is bypassing everything, would be really hard to detect in audit.



XZ Case Study (Detection)

- How was it noticed? postgres guy was benchmarking using perf and noticed ssh connections taking 10x as long, tracked it down. Luck.
- Attacker playing long game. Years to do things. Apparently had other accounts that were doing things like patching fuzzer/security tools to try to ignore this, also bugging stable distributions to update to newest version
- Only really got to the point of testing in Linux distros



(debian unstable)

- Who responsible? They were careful to make it look like from China, some analysis of timezones maybe it was eastern-europe/middle east timezone
- Cuckoo's Egg by Cliff Stoll