Foundations of Network and Computer Security

John Black

Lecture #15
Oct 20th 2005

CSCI 6268/TLEN 5831, Fall 2005

Announcements

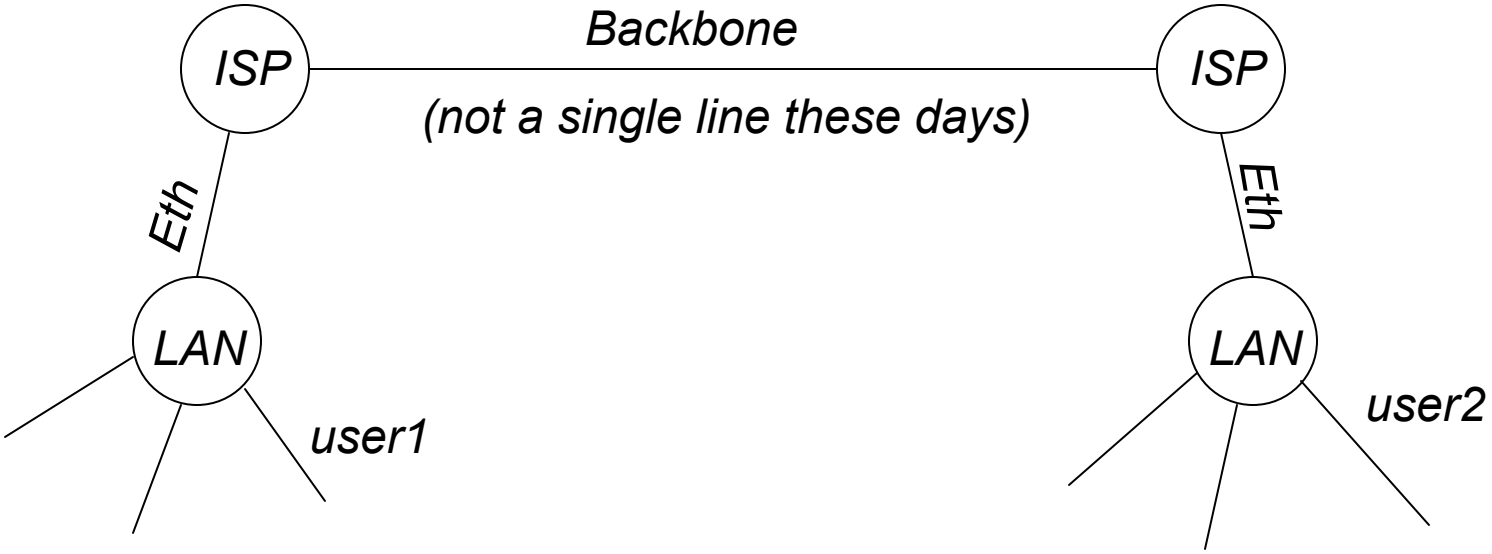
- Reading: How to 0wn the Internet
  - See schedule page

- Project #1 is assigned
  - See web page for description and cacert.pem
  - Due Thurs, Nov 3rd (distance students too!)
  - Note: Martin is out, Tues thru Sunday next week

- Midterm #2 is Nov 8th (2.5 weeks from now)

Where were we?

- The basic model:



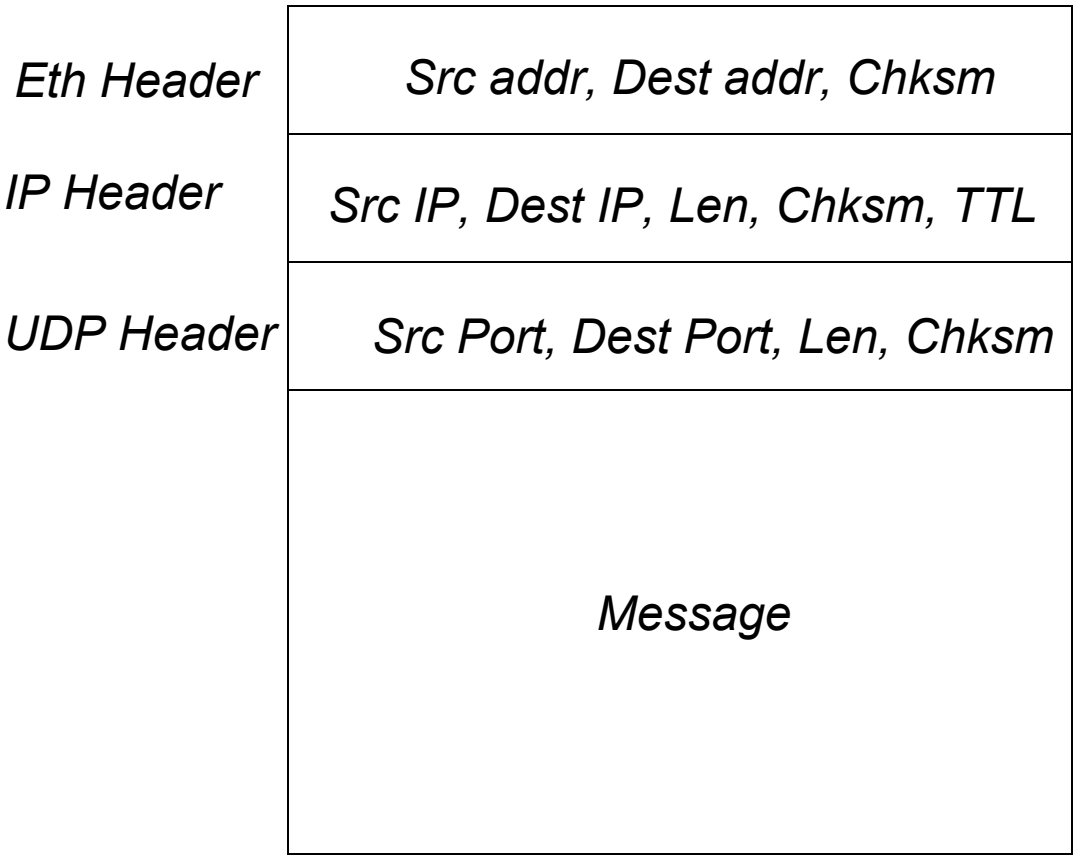
Basic Networking

- Suppose user1 sends a UDP packet to user2, what happens?
  - What's UDP?
    - User Datagram Protocol
    - Just like IP but with ports
  - Well, first we need an IP address!
    - What's an IP address
    - For IPv4, it's a "dotted quad" of bytes
      - Ex, 128.138.242.21
      - 32 bits
    - For IPv6, it's 128 bits
      - 16 bytes in hex separated by colons

Sending a UDP packet

- Assume IPv4
  - Get IP address via DNS
    - Domain Name Service
    - Distributed database mapping textual names to IP addresses
    - Insecure
      - DNS spoofing
      - More on this later
  - Ok, so we have an IP address
  - And we presumably have a port #

Pack it Up!



Ethernet addresses are called "MAC addresses"

Ethernet checksum is actually appended to end of packet

Ethernet MTU is 1500 bytes

Routing on a Network

- Usually done via OSPF or LSP for LANs
  - Open Shortest Path First, Link-State Protocol
  - These protocols assume “modest sized” networks
  - A routing protocol decides how to forward packets based on routing tables
- BGP is used on backbone
  - Border Gateway Protocol
  - Routes using incomplete information

Local Routing Table

- Our local routing table (on host of user1) is not going to have a route to IP of user2
  - Routing table will therefore send our packet to the gateway
  - Gateway is the machine/router on the “edge” of the network responsible for processing all incoming/outgoing traffic from/to the LAN
    - NAT boxing, firewalling, and other stuff is usually done here as well

Getting to the Gateway

- How do we route to the IP address of the gateway on our local Ethernet?
  - ARP (Address Resolution Protocol)
    - Translates IP addresses into MAC addresses
    - Caches old lookups, so we probably already have the MAC address of the gateway
    - If not, we send an ARP Request to the LAN, including the IP address whose MAC we seek
    - Owner (ie, the gateway) sends ARP Reply with his MAC address and we cache it
      - Usually, all other machines who hear the ARP Reply cache it as well
      - Leads to attacks… more later

Sending to the Gateway

- Now we have the MAC address of the gateway
  - Send our packet to the gateway via the Ethernet protocol
  - This is usually done with a hardware device (network card) which often puts the Eth header on your packet for you, computes checksums, etc.
    - Broadcasts packet, detects collisions
    - Exponential backoff
    - Promiscuous mode – Sniffers use this
      - Works through hubs, but doesn't work through switches on a switched Ethernet
      - You can often fool switches

Gateway Receives Eth Packet

- Strips Eth header and again tries to route the resulting IP packet
  - Looks in routing table, sends to ISP
  - ISP probably routes using BGP
  - Reaches other ISP
    - Note that we're using other Ethernets and similar physical-layer protocols for each hop!
  - Other ISP routes to other LAN's gateway
    - Gateway sees IP is in its range and does ARP to route to user2

User2 Receives Packet

- User2 receives the IP packet
  - Removes IP header
    - No one else (is supposed to) look inside packet until user2 receives it
    - NAT boxes break this rule
    - Firewalls break this rule
  - See it's a UDP packet and "sends" to proper port
  - Ports are mapped to applications via listento()
    - Application receives message and processes it

Other Protocols

- We didn't even talk about SLIP or PPP
- ATM, FDDI, Wireless
- What about DHCP?
  - Dynamic IP addresses
- There is also ICMP
  - Internet Control Message Protocol
  - Echo (ping), traceroute
- Application Layer Protocols
  - HTTP – Hypertext Protocol
  - SNMP – Network Management
  - SMTP – Sendmail
  - POP/IMAP – Mail protocols

MTU – Maximum Transmission Unit

- MTU for Ethernet is 1500 bytes
  - If MTU is exceeded, packet is “fragmented”
  - IP has support for packet fragmentation and reassembly
  - A packet is broken into as many pieces as necessary to comply with MTU
  - Fragments routed as regular IP datagrams, independent of each other
  - Reassembly done at host only

IP – Best Effort Datagrams

- IP is “best effort”
  - There is no tracking of packets
  - If something is dropped… oh well
    - ICMP message is sometimes generated and received
  - If one fragment is dropped, many transport layer protocols (like TCP) will consider the whole thing lost and not ACK
  - This seems bad, but it’s one of the biggest successes of IP
  - UDP is IP with ports, so it too is “best effort”

TCP – Transmission Control Protocol

- Stateful connections
  - Runs over IP just like UDP, but adds more than just ports
  - Establish a connection with listen() and connect()
    - IP and UDP were “stateless” protocols
  - Reliable delivery
    - Unlike best-effort, this protocol guarantees delivery of packets, in proper order
    - Uses sequence numbers, sliding windows, ACKs every transmission

Crypto on a Network

- How do we do crypto on a network?
  - We've seen application-layer examples
    - SSL/TLS, SSH
    - This is called "end-to-end" cryptography, meaning between hosts
    - The routers don't care if the innermost part of each packet (the "payload") is ciphertext or plaintext
  - IPSec
    - IPSec does crypto at the network layer (the IP layer)
    - Extremely well-engineered; hardly used
    - We won't study IPSec in this course

Network Security: The Biggest Challenges

- What are the biggest problems now, today, on the Internet
  - What are the most common types of attacks?
    - Viruses, worms
    - Break-ins via software vulnerabilities
    - Denial of Service attacks (DoS)
      - And Distributes Denial of Service (DDoS)
  - What about keyloggers, spyware, rootkits?
    - Not as relevant to network security
    - More likely to be end-results of other break-ins
      - Many viruses will install a keylogger

Viruses (Worms)

- Today, most everyone just calls them viruses
  - Technically most are “worms”
  - Worm is a self-contained propagating program
  - Viruses embed in other programs and self-replicate
    - Kind of like viruses in biology

Viruses: History

- Morris Worm, Nov 2nd, 1988
  - The first worm (I know of) was the Morris worm
  - Robert T. Morris, Jr.
    - 23 years old
    - Cornell grad student
    - Father worked at the NSA (whoops!)
  - Wrote a self-propagating program as a “test concept”
    - Exploited Unix vulnerabilities in sendmail and fingerd
    - Released at MIT
    - Bug in the worm caused it to go wild
      - Probably wouldn’t have caused much damage otherwise!

Morris Worm (cont)

- Shut down thousands of Unix hosts
  - But this was 1988…
- Reactions
  - People didn't know what to do, so they panicked
    - Disconnected from net
    - Unable to receive patches!
  - Morris fined $10k, 3 yrs probation, 400 hrs community service
  - CERT was created

Modern Viruses

- Almost all look for Windows hosts
  - Windows runs on more than 90% of desktops these days
  - A lot of hosts on cable modems
    - Fast, always on
  - Destructive payloads
    - Wipe hard disk, eg
    - Some install backdoors for later use
  - All kinds of weird behaviors though
    - Some innocuous

Viruses: Why?

- Who writes these things?
  - Typical profile: male, teenager, geeky, smart
  - Script Kiddies
    - Don’t really write them, but launch them
    - Sometimes make small mods and call them their own
    - Scariest hackers: beyond the reach of the law
- Why?
  - Intellectual challenge (sigh…)
  - Peer recognition
  - Bot building (Zombie armies)
  - Because it’s there?

Brief History

- Would take weeks to look at all the viruses we've seen
  - Also, wouldn't be *that* instructive
- We'll look at the ones I think were most instructive, important, and which have interesting lessons
  - So it's a *selective* brief history of viruses

AIDS Trojan (1989)

- Often called a “virus”
  - A trojan is a program with a “surprise” payload
  - The AIDS trojan was distributed as a way to enable graphics on TTL monitors
    - Duh
  - Payload: erase harddisk

- Interesting note: first virus scanners appear around this time (1990)

Tequila (1990)

- First polymorphic virus
  - Polymorphic means “changing form”
  - This was done to defeat virus checkers

- Current status (2005) of polymorphic viruses
  - Well, the current virus toolkits (MPC, VCS, VCL) create code which is still caught by scanners
    - VCL – Virus Creation Laboratory (1992); pull-down menus, selectable payload
  - But it’s possible to make a toolkit which will defeat the scanners – hasn’t been done yet (to my knowledge)

Michelangelo (1992)

- First virus to get lots of headlines
  - Lives in MBR (master boot record)
  - Targets MS-DOS machines
  - Transfers to floppies/hard-disks when intermixed
    - Note this predates widespread use of the Internet
  - Payload: destroy boot and FAT on March 6th
    - Michelangelo's birthday

DMV (1995)

- Word Macro virus
  - Macros are sets of executable instructions specific to an application
  - Back in 1995, MS Word was configured out-of-the-box to execute immediately any macros in a Word document
  - This meant that simply opening a document in an email or from the Web was dangerous
- DMV
  - Distributed with the paper “Document Macro Viruses”
  - Harmless (even had dialog boxes)
  - Trying to prove a point
- Other macro viruses possible with Excel, Access, Adobe Acrobat, and more

Back Orifice Trojan (1998)

- Pun on MS Back Office
  - Allows remote access via the Internet of Win 95/98 boxes (BO-2000 runs on Win 2k and NT)
  - Waits for commands starting with "*!*QWTY?"
    - US version used encryption; international could not! ☺
  - Doesn't show up in the task list
  - Written by cDc (Cult of the Dead Cow) and advertised as a legitimate tool
    - Used by network managers, in fact
  - But has been abused of course
    - Has plug-ins to 0wn your box (view remote screen, download registry, etc)

Melissa (1999)

- Just when you thought it was safe
  - Melissa was a major virus
  - Combination Word Macro virus and email virus
    - Sent as an attached doc file
    - Scanned Outlook address book and sent itself to first 50 addresses
    - Subject: “Important message from <you>”
    - Body: Here is the document you asked for; don’t show anyone
      - Then attached the most recent doc you had been working on, infected with Melissa
    - Spread VERY rapidly all over the world
      - Tons of variants

ILoveYou (2000)

- Clever technology, great social engineering
  - Subject: I love you
  - Body: Kindly check attached love letter from me
    - And message was from sender you know!
  - Attachment: LOVE-LETTER-FOR-YOU.TXT.vbs
    - Note the double-extension – VBS script
    - If you didn't have your OS set to show extensions, you'd just see **LOVE-LETTER-FOR-YOU.TXT**

ILoveYou (cont)

- Complex payload
  - The worm copies itself into two places where it will be executed on each computer restart.
  - It will try to send itself to every entry in your Outlook address book.
  - The worm searches all drives (local and networked) for files ending in VBS, VBE, JS, JSE, CSS, WSH, SCT or HTA. If found, they are overwritten with the virus and their extension renamed to .VBS.
  - Graphics file with JPG or JPEG extensions are also overwritten with the virus and .VBS added to their name (so they will end up with a double extension).
  - Multimedia files with MP2 and MP3 extensions are marked as hidden and then copied to a new file with the same name and .VBS added. (Note that of all the files attacked, these are the only ones that can be recovered directly; all others have to be recovered from backups.)

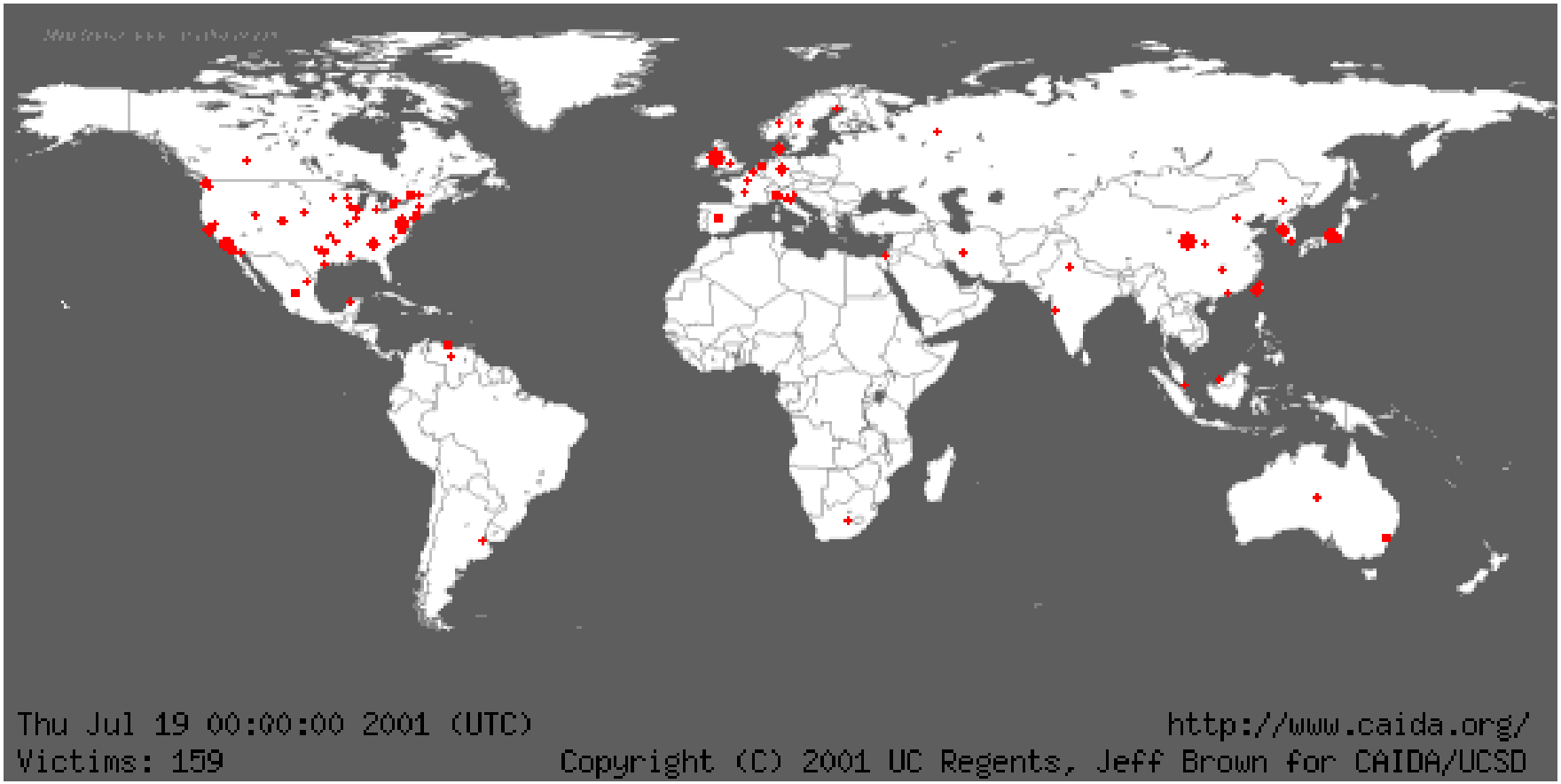
ILoveYou (cont)

- Was wildly successful
  - Mostly due to human nature: someone loves me
- Has countless variants
  - Joke attached
  - Mother's Day Gift confirmation
    - Now that's just wrong
  - How to stop the ILoveYou virus

It Gets Worse

- SirCam, Nimda, CodeRed, BadTrans
  - Nimda: very complex
    - Mostly spread via unpatched IIS servers, but also
      - Via email (attached EXE)
      - Browsing dubious web sites with unsecured browser
      - Using backdoors from other viruses (CodeRed II, eg)
      - Payload: back door access
  - Code Red: still around today!

Code Red Spread
(14 hrs, 350,000 hosts)



Code Red Payload

- Coordinated attack against www1.whitehouse.gov
  - Used hardcoded IP address
  - Checked to ensure port 80 was active first
  - Easy to stop this, and indeed the IP was moved before Code Red launched its payload, so no direct damage done
- windowsupdate.microsoft.com was infected too
  - Users got infected while trying to patch!
- First version used static seed for random()
  - Limited the number of IPs it generated
- Five days later this was fixed

Code Red Details

- Spreads as a bad HTTP request.
- The IIS system mishandles the request, and instead executes the included packet with full permissions.
- The infected server then creates 99 threads which each attack random IP addresses
  - Random number generator works properly now
- This continues for the 1-19 of the month. On the 20-27 of the month, all the threads attack a specific IP at www.whitehouse.gov
  - Still see network traffic surges *today* from this worm
  - People don't patch!
- Defaces current pages on the server
  - Welcome to http://www.worm.com!
    Hacked by Chinese!

SQL/Slammer (2003)

- Exploits buffer overflow in MS SQL server
  - UDP traffic to port 1434
- Side-effect was DoS
  - Worm propagated so fast that it shut down many sites
  - Launched 12:30am EST victim numbers doubled every 8.5 seconds
  - By 12:45am, large pieces of the Internet were basically gone
    - 300,000 cable modem users in Portugal down
    - South Korea off the map (no cell phones or computer access)
  - Seattle 911 resorted to paper
  - Continental cancelled flights from Newark hub

Witty Worm (March 2004)

- Attacked a *security* product!
  - Internet Security Systems (ISS)
    - ISS RealSecure Network, RealSecure Server Sensor, RealSecure Desktop, and BlackICE
  - You can't even trust your security systems?!
- Vulnerability revealed by eEye Digital Security
  - Witty released *10 hours* after vulnerability was released
  - Destructive payload (deletes pieces of hard drive)

Flash Viruses

- Viruses can spread very fast
  - SQL/Slammer had only a 376 byte code size
  - No pause between propagation attempts
- Reading assignment
  - Read “How to 0wn the Internet in your Spare Time”
- A real problem
  - If you reinstall an old OS and attempt to download patches, you may be infected before you can patch!

Prevention

- Stay patched
  - windowsupdate.com
  - Linux patches (yum)
- Reduce network services to those needed
  - “Best block is not be there” – Mr. Miagi
  - Windows still comes with a ton of stuff turned on
    - Getting better though!
  - SQL Slammer victims didn’t even know they were running an SQL server!
  - netstat –a
    - Might surprise you

Prevention (cont)

- Don’t open attachments unless you’re sure
  - Always run a virus scanner
  - Even Word docs are dangerous
- Don’t visit questionable web sites
  - Esp if your browser is set to low security levels
  - Javascript is evil
- Felton’s Javascript attack

Trojans

- Malicious code hidden within another object
  - Email attachments can contain trojans
  - This is how many viruses spread
- Backdoor is usually considered as a synonym
  - Putting a backdoor into login.c qualifies

Thompson's Turing Award Lecture (1995)

- Thompson and Ritchie won the Turing award for creating Unix
- Thompson's is my favorite Turing award lecture
  - "Reflections on Trusting Trust"
  - Please read it (it's short)
- His lecture has three stages
  - Stage I: a "Quine"
  - A Quine is a program which outputs its own source code

A Quine in C

```
char*f="char*f=%c%s%c;main()
  {printf(f,34,f,34,10);}%c";
  main(){printf(f,34,f,34,10);}
```

- We printf the string f, inserting f into itself as a parameter
  - Yow!
- We could attach any extra code we like here
- File this away in your head for now: we can write a program which outputs its own source code

Thompson, Stage II

- Note that a C compiler is often written in C
  - Kind of strange chicken-and-egg problem
  - How to bootstrap
- Interesting “learning behavior”
  - You add a feature, compile compiler with itself, then it “knows” the feature
- Once you get a rudimentary compiler written, it can be arbitrarily extended

Thompson, Stage III

- Add a backdoor to login.c
  - Allow valid passwords *plus* some “master” password
  - Note that this would be caught soon enough because it exists in the login.c source code
- Ok, so be sneakier
  - Add code in cc.c (the C compiler) to add the backdoor to login.c whenever compiling login.c
  - Add self-replicating code to the C compiler to reproduce itself plus the login.c backdoor!

Implementing the Trojan

- Now compile login.c
  - Compiler adds the backdoor
- Compile cc.c
  - Compiler sees that it's compiling itself and self-replicating code runs to ensure login.c trojan and cc.c trojan are compiled into cc binary
- Now remove all this new code from cc.c
  - Back door exists only in binary!
  - login.c and cc.c will continue to have trojan even after infinite recompiles

Moral of the Story

- The amount of cleverness we haven't even thought of yet is scary
  - We're probably never going to have completely secure computers and networks
  - The most we can hope for is "best effort" from those we trust and from ourselves
  - It's going to be an eternal battle between us and the criminals