Foundations of Network and Computer Security

John Black

Lecture #10
Sep 23rd 2004

CSCI 6268/TLEN 5831, Fall 2004

Announcements

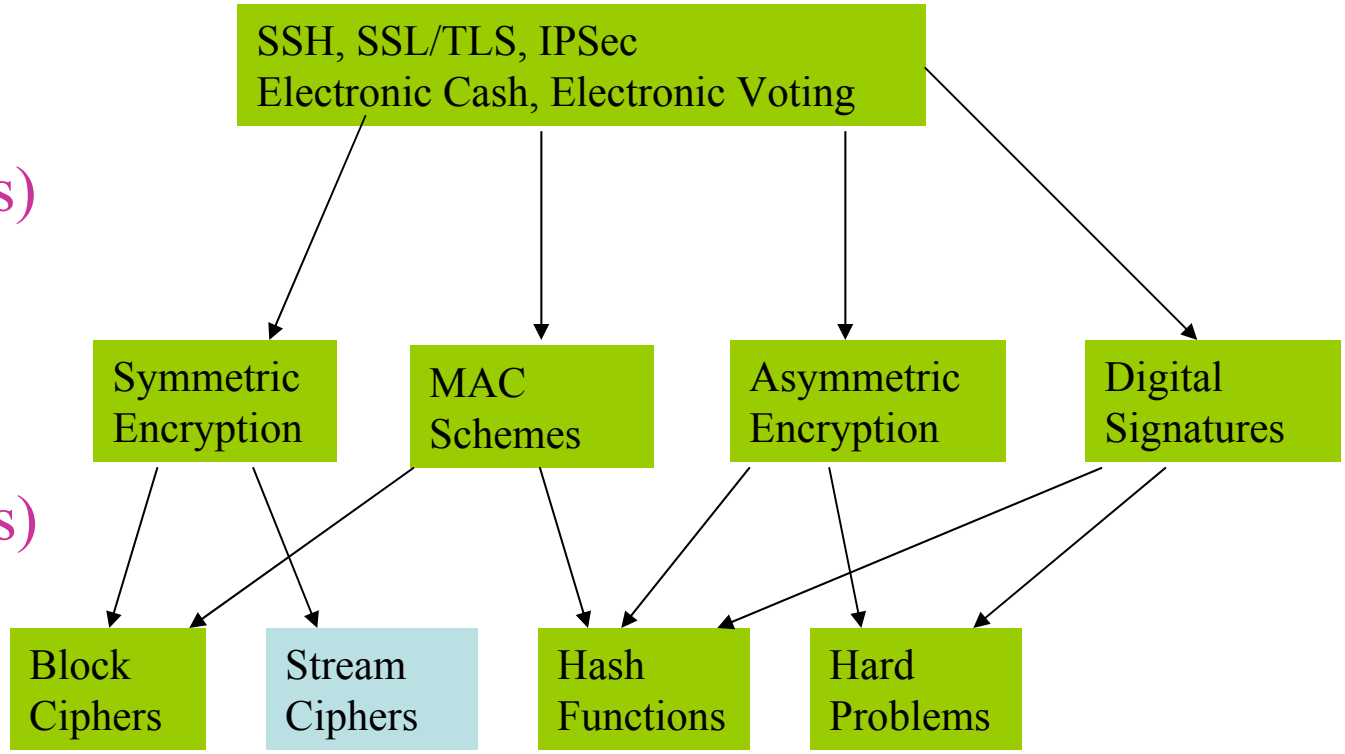
- Quiz #2 next meeting
  - Will cover material up to **last** class (ie, Tuesday's lecture)
    - Up to OpenSSL stuff
    - Nothing from today or next Tuesday's lectures
    - No, you don't have to memorize OpenSSL commands, flags, and oddities
    - Yes, you should know *why* things are done the way they are
    - Review of the RSA stuff on our web site
      - On the course schedule for last lecture
      - There's more than we talked about in class

The Big (Partial) Picture

Second-Level Protocols
(Can do proofs)

First-Level Protocols
(Can do proofs)

Primitives



(No one knows how to prove security; make assumptions)

Network Security

- Haven't we already been talking about network security?!
  - Kind of… cryptography is a central part of it
  - Cryptography is nice because it's a neatly packaged science; but we're done for now
  - Network security itself is a vast area with fuzzy borders
  - Research tends to be more ad hoc
    - How do we stop attack A, how do we prevent bug B, how do we detect or tolerate intrusions, etc.

Crypto …. Good

- The easiest way to break into a computer is usually not by breaking the crypto
  - We've said this a number of times in this class before; there are usually easier ways
- Let's suppose we want to break into a friend's account on CSEL
  - What kind of friend are you??
  - Ok, give me methods… *simple* methods

Breaking into a “Friend’s” Account

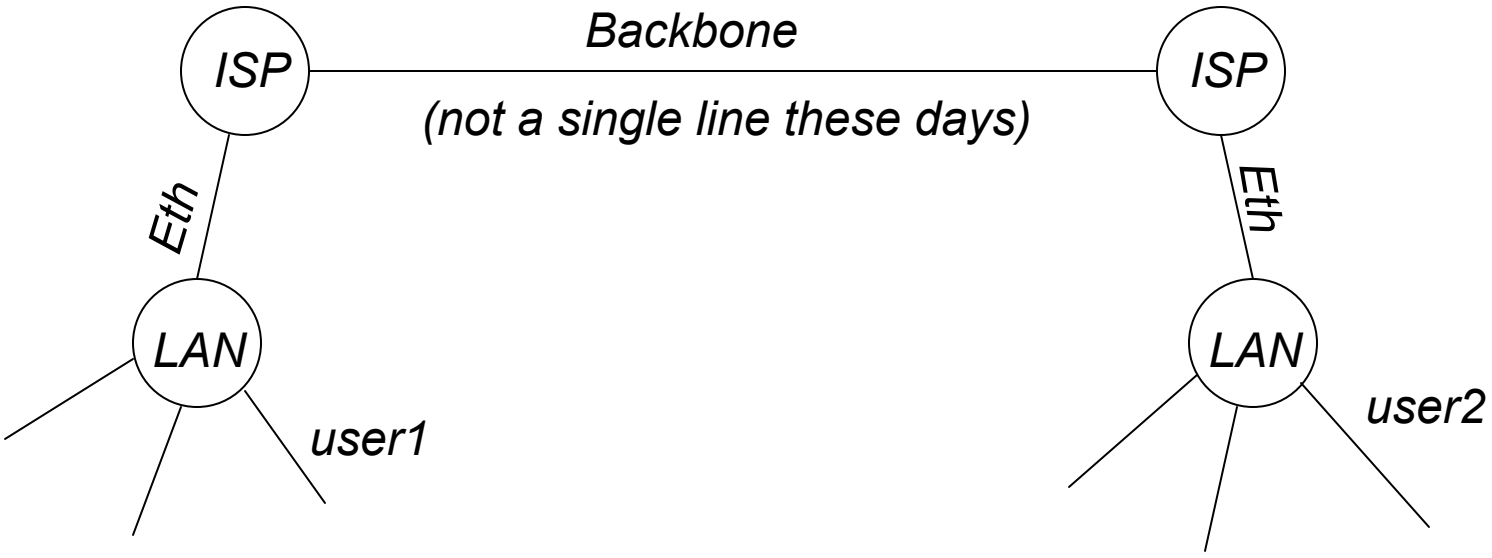
- Digression
  - Before we talk about this, let me introduce the “John Disclaimer”
  - I would like each of you to sign a statement “promising not to be evil”
  - I will hand this out at the end of lecture
    - Please remind me
  - It’s also on our web site…
    - Distance students, please print this out and send it in

Ok, Breaking into a “Friend’s” Acct

- Fake Login Screen
- Shoulder Surfing
- Password Cracker
  - MD5 hashes publicly available on web
- Social engineering
  - Hard to trick CSOps though
  - Might be easy to impersonate CSOps! ☺
- Key loggers
  - Software and hardware versions
- Keystroke analysis
  - Ok, getting obscure

Networking Refresher

- For some of you this will be boring… sorry
- 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

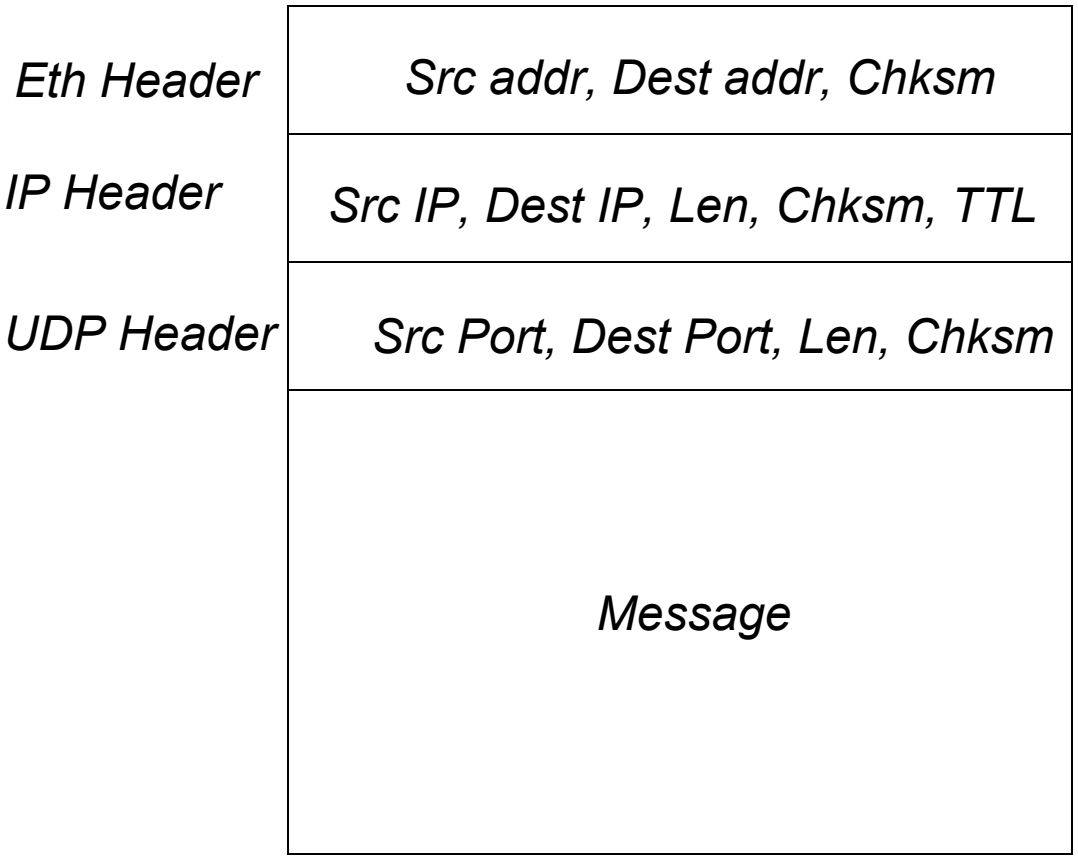
Running out of IP addresses

- 2^{32} is a lot, but we're having problems
  - A lot of hosts out there
  - The class A, B, C scheme is wasteful
    - Though subnetting helps
    - A lot of NAT Boxing "helps"
    - Since we're getting by, it means a slower migration to IPv6

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 to 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
  - 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
  - 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
      - A recent virus was found to 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

CERT -- They were first

- Carnegie mellon Ermergency Response Team
  - But don’t expand it into an acronym
- Provide technical advice and coordinate responses to security compromises
- Identify trends in intruder activity
- Work with other security experts to identify solutions to security problems
- Disseminate information to the broad community
- Analyze product vulnerabilities
- Publishes technical documents
- Presents training courses

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?