CIS 4930: Secure IoT

Prof. Kaushal Kafle

Lecture 15

Class Notes

Quiz time!

- Project report submission was ***yesterday***.
- Next class
  - Outline of the next project – **Security analysis of IoT apps**
  - Similar to before, you will submit a project proposal.

Smart Home Privacy

Smart Homes

Transmit *device and environment data* to remote servers!

↓

Vendors may **process**
privacy-sensitive information
about home usage!

Behavior Profiling

Affecting Insurance Claims

Inferring Sensitive Information

Smart Homes

Transmit *device and environment data* to remote servers!

↓

Vendors may process **privacy-sensitive information** about home usage!

↓

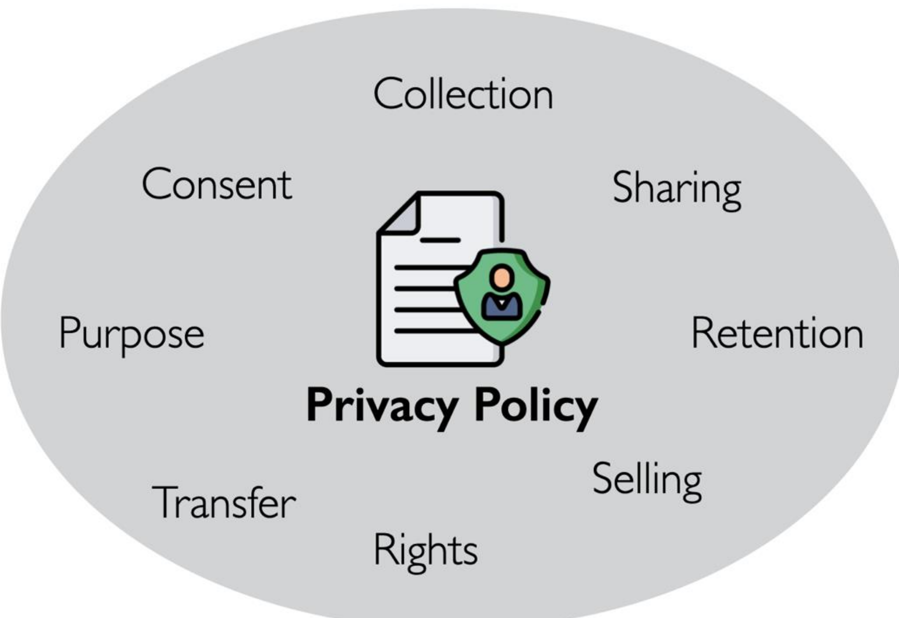
Consumers should be informed about the privacy practices with regard to device data.

Behavior Profiling

Affecting Insurance Claims

Inferring Sensitive Information

Privacy Policies



- **Legally Binding**
- **Conveys Data Handling Practice**
- **Informed Decision Making**

Privacy Policies

Inferring Sensitive Information

"XYZ may be required to process data that are deemed by applicable legislation to be sensitive, since they may incidentally reveal Users' religious beliefs or sexual orientation.

This may be the case if electricity and the Application are not recorded as being used between Friday night and Saturday night (***which could suggest that users belong to the Jewish faith***) or if only one room (such as a bedroom) is registered on the Application for a home shared by two people of the same sex (***which could suggest the occupants' homosexual or bisexual orientation***)."

Understanding Smart Home Privacy

How difficult is it for consumers to *obtain* privacy policies that apply to their smart home devices?	**Availability**
How *precisely* is the collection and sharing of device data described in smart home product privacy policies?	**Content**
How *comprehensive* are smart home product privacy policies in describing the collection/ sharing of device-data?	**Coverage**

Availability Findings

Finding 1: 10.57%, i.e., 63/596 of smart home vendors *do not provide privacy policies*, i.e., not even for their websites.

Finding 2: 43.52% do not provide policies for *smart home products.*

Finding 3: Only 64.38% made policies available from their website.

Source	Number of device policies
Vendor websites	188 (64.38%)
Google Search	41 (14.04%)
Google Play Links	21 (7.19%)
Mobile Apps	42 (14.38%)
Total	292 (i.e., 100%)

Finding 4: Device privacy policies can be *extremely difficult* to obtain.

Finding 5: 6.84% of the vendors do not even make their *website privacy policies* easily available.

Why is all this a problem?

Policy Content Findings

Finding 6: 26.05% of the policies describe collection using *broad terms* rather than discussing specific device types or device data *(e.g., usage information).*

Finding 7: 70.42% of device privacy policies specify collection at the granularity of device data (e.g., temperature information collected from thermostat).

Why is all this a problem?

Finding 9: 8 vendors explicitly state that they *do not collect any information* within their privacy policy, which may be inaccurate

Finding 10: 186/284 or 65.49% of device privacy policies only discuss sharing practices for PII or "personal data," but not for device data.

Finding 11: 34.28% of device privacy policies do not specify with whom the data is shared.

Finding 12: 2.1% of vendors do not discuss sharing data and only 3.87% state that they do not share data.

Policy Coverage Findings

Finding 13: 50/200 (25%) of the privacy policies that precisely discuss device data only discuss a subset of their available devices.

Imagine a vendor that sells both light bulbs and motion sensors.

Why is all this a problem?

Finding 14: Vendors do not differentiate their privacy disclosures for devices that produce similar data but have vastly different privacy implications.

Imagine a smart camera vendor that sells both outdoor cameras and baby monitors.

TCP/IP security
(read the Bellovin paper!)

Network Stack, yet again

Application

Transport

Network

Link

Physical

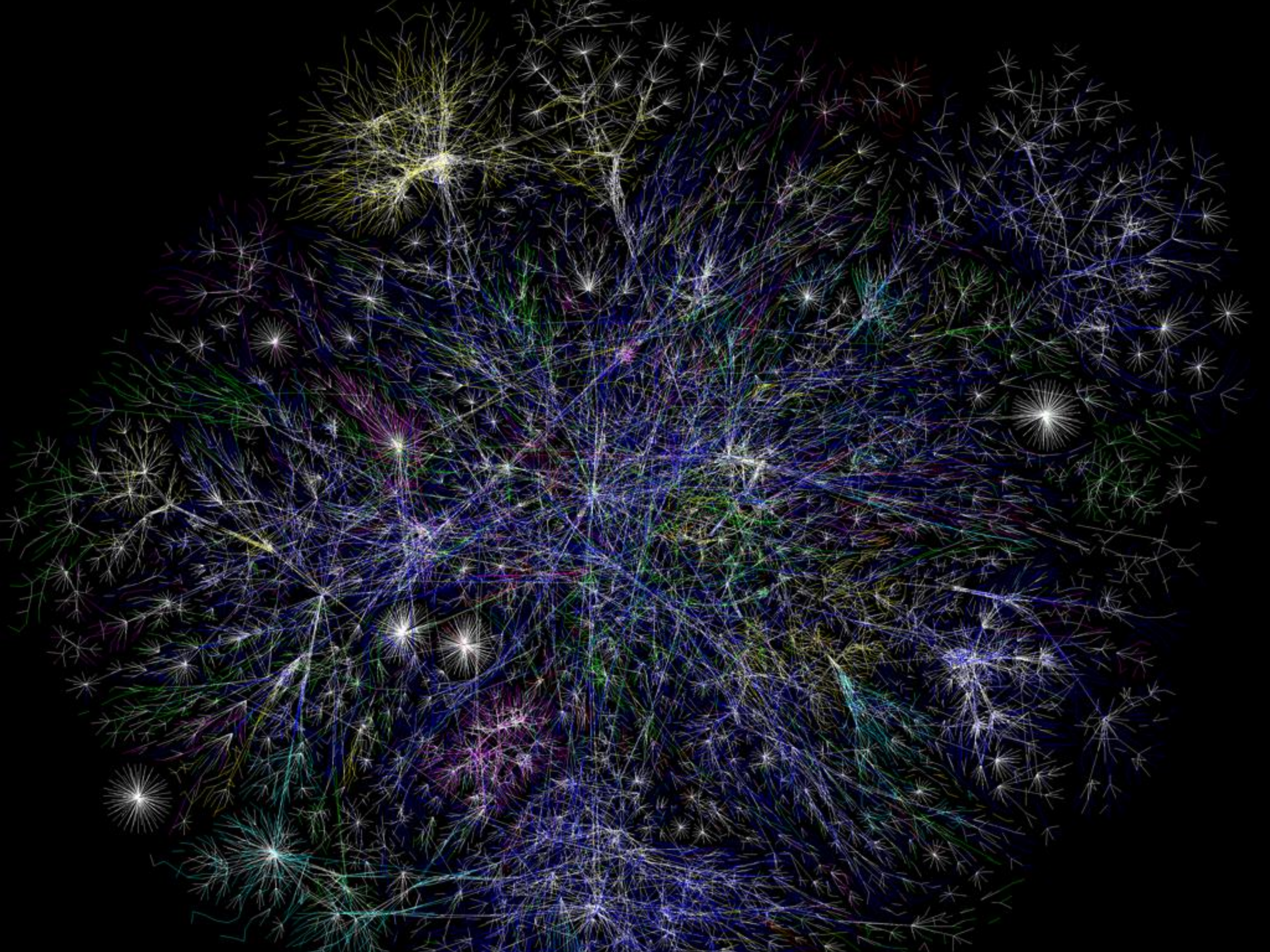
Networking

- Fundamentally about transmitting information between two devices
- Communication is now possible between any two devices anywhere (just about)
  - Lots of abstraction involved (see previous slide)
  - Lots of network components (routers)
  - Standard protocols (e.g., IP, TCP, UDP)
  - Wired and wireless
- What about ensuring *security*?

Network Security

- Every machine is connected
  - No barrier to entry
  - Lots of users.
    No inherent way to identify a specific user operating a specific computer.

"On the Internet, nobody knows you're a dog."



Exploiting the network

- The Internet is extremely vulnerable to attack
  - it is a huge open system ...
  - which adheres to the end-to-end principle
    - *smart end-points, dumb network*
- Can you think of any large-scale attacks that would be enabled by this setup?

Network Security: The high bits

- The network is …
  - … a collection of interconnected computers
  - … with resources that must be protected
  - … from unwanted inspection or modification
  - … while maintaining adequate quality of service.

Network Security: The high bits

- Network Security (one of many possible definitions):
  - Securing the network infrastructure such that the integrity, confidentiality, and availability of the resources is maintained.

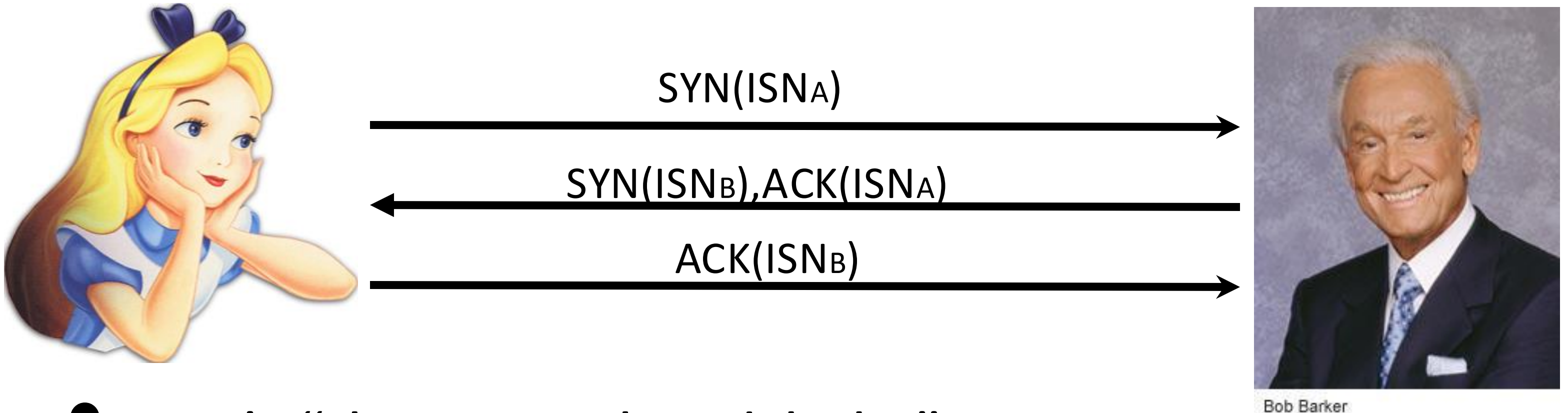
TCP Properties

- Works under the notion of data segmentation and reassembly.
- **Reliable** communication
  - i.e., reliable data transfer
- **Error detection and correction**
  - Has to keep track of the data packets order, and what has been received

Steven Bellovin's Security Problems in the TCP/IP Protocol Suite

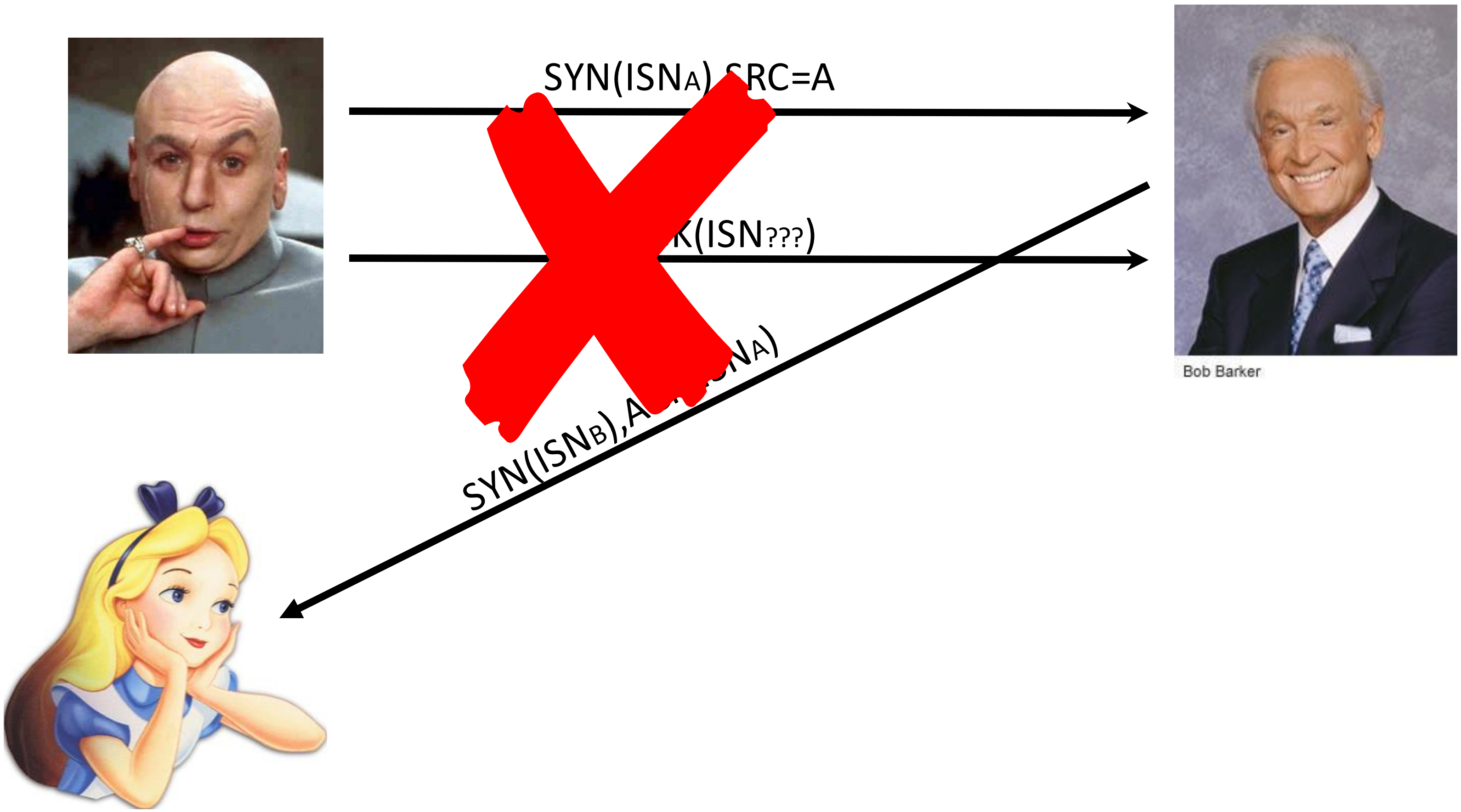
- Bellovin's observations about security problems in IP
  - Not really a study of how IP is misused (e.g., IP addresses for authentication), but rather what is inherently bad about the way in which IP is set up
- A really, really nice overview of the basic ways in which security and the IP design is at odds
  - E.g., TCP/IP protocol suite is not built with malicious attackers in mind.

TCP Sequence Numbers

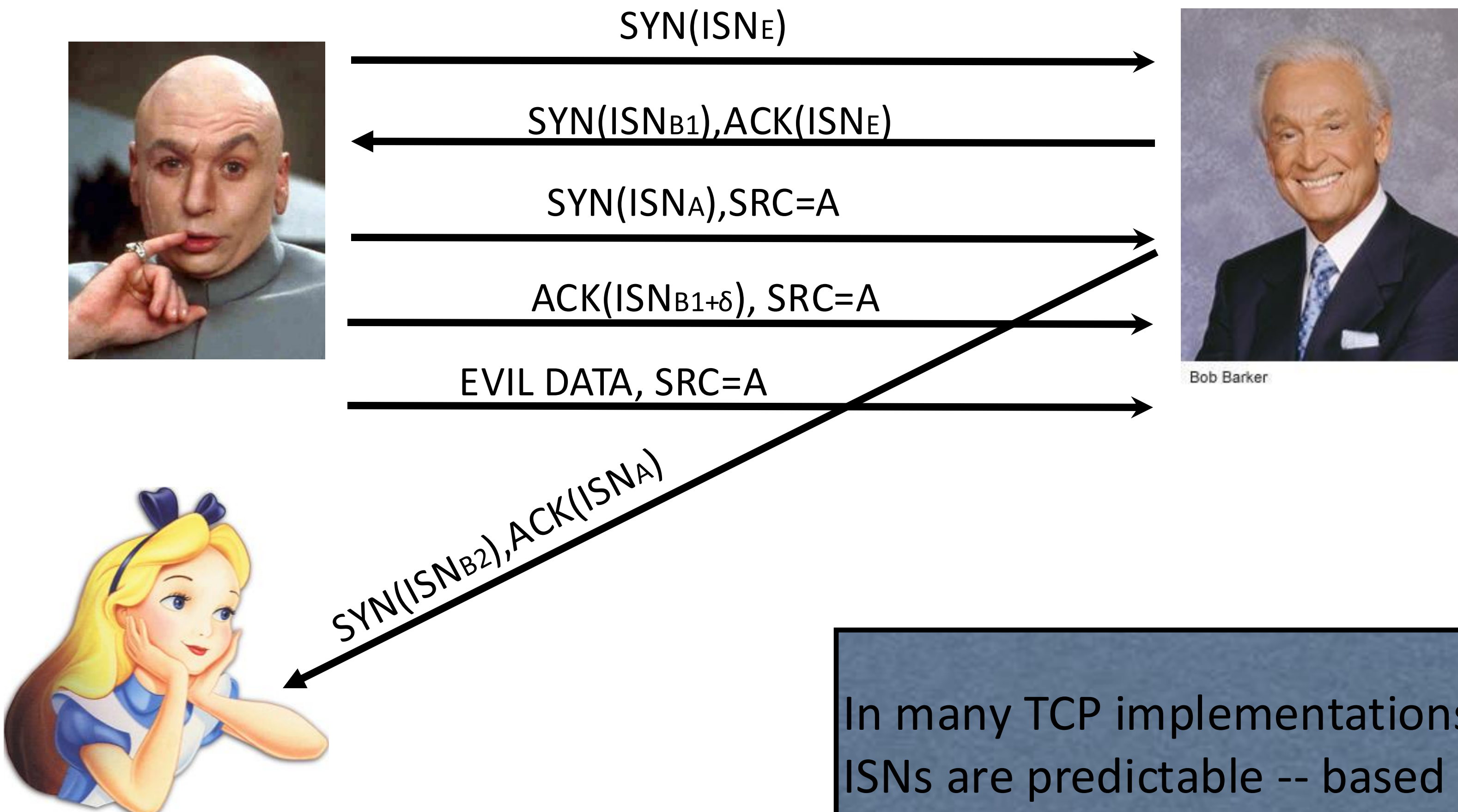


- TCP’s “three-way handshake”:
  - each party selects Initial Sequence Number (ISN)
  - shows both parties are capable of receiving data
  - offers some protection against forgery -- **HOW?**

TCP Sequence Numbers



TCP Sequence Numbers



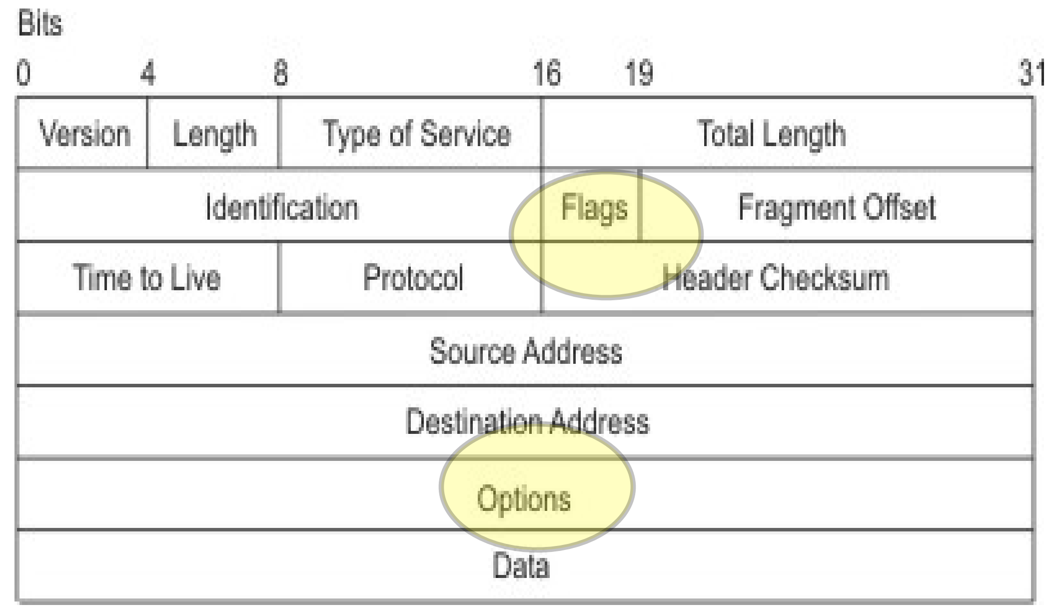
In many TCP implementations, ISNs are predictable -- based on time (e.g,. ++ each 1/128 sec)

How do we fix this?

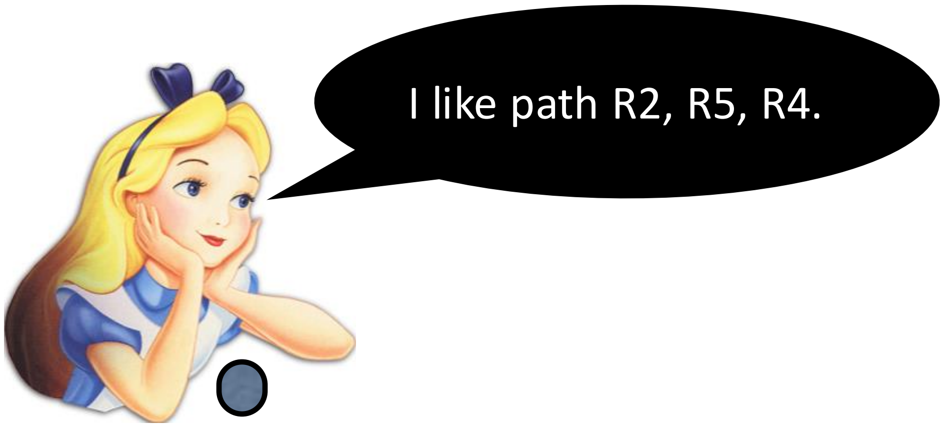
- Randomize ISNs
  - How?
- Hash repeatedly? -> Drawbacks?
  - *Deterministic*
- RNGs? -> Drawbacks?
  - *Slow*
  - *Increased*

Source Routing

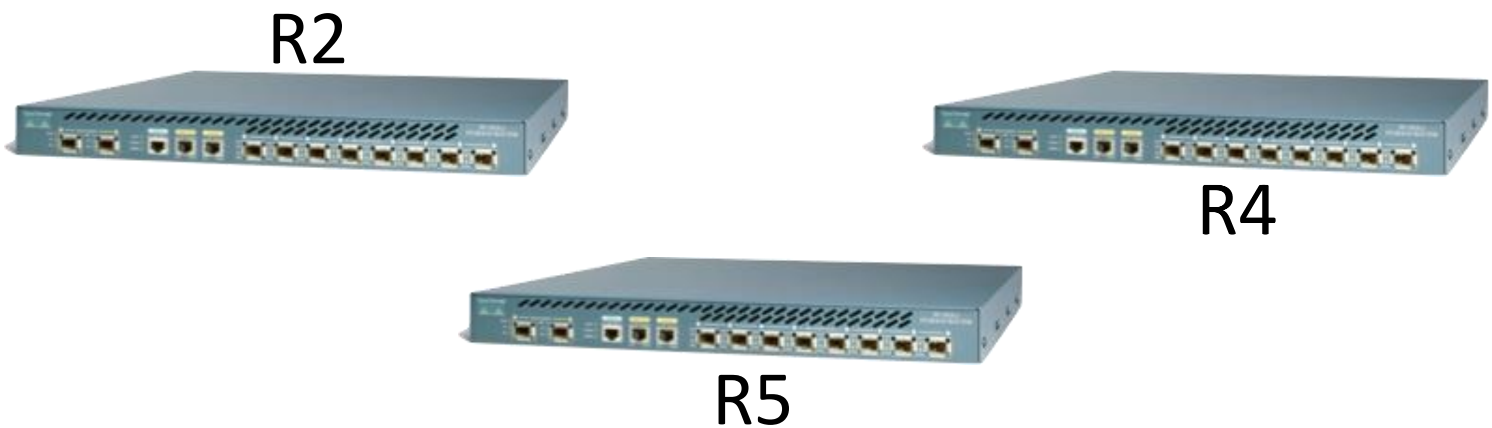
- Standard IP Packet Format (RFC791)
- Source Routing allows sender to specify route
  - Set flag in *Flags* field
  - Specify routes in *Options* field



Source Routing




Bob Barker



Source Routing

- Q: What are the security implications of Source Routing?
  - Access control?
  - DoS?

- Q: What are the possible defenses?
  - A: Block packets with source-routing flag

Routing Manipulation

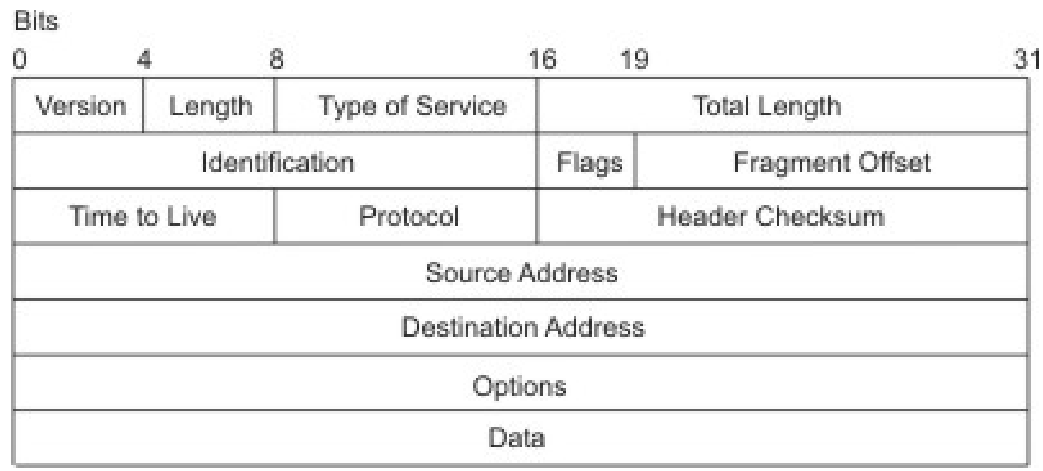
- RIP - Routing Information Protocol
  - Distance vector routing protocol used for the local network
  - Routers exchange reachability and “distance” vectors for all the sub-networks within (a typically small) domain
  - Use vectors to decide which route is best
- **Problem:** Data (vectors) are not authenticated
  - Forge vectors to cause traffic to be routed through adversary
  - or cause DoS
- Solutions: ? (still an open problem)

Internet Control Message Protocol (ICMP)

- ICMP is used as a control plane for IP messages
  - Ping (connectivity probe)
  - Destination unreachable (error notification)
  - Time-to-live exceeded (error notification)
- ICMP messages are easy to spoof: no handshake
- Some ICMP messages cause clients to alter behavior
  - e.g., TCP RSTs on destination unreachable or TTL-exceeded
- Enables attacker to <u>remotely</u> reset others' connections
- Solution:
  - Verify/sanity check sources and content
  - Filter most of ICMP

Ping-of-Death: Background: IP Fragmentation

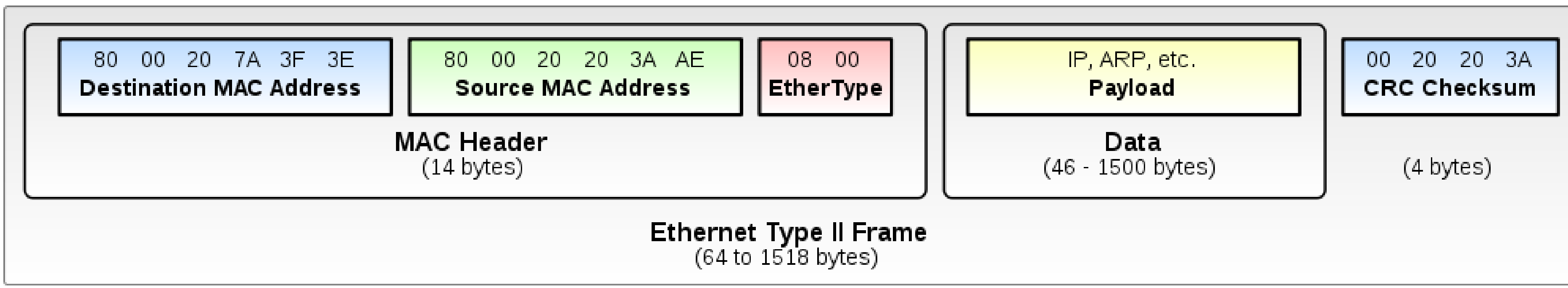
- 16-bit "Total Length" field allows 2^{16}-1=65,535 byte packets
- Data link (layer 2) often imposes significantly smaller **Maximum Transmission Unit** (MTU) (normally 1500 bytes)
- Fragmentation supports packet sizes greater than MTU and less than 2^{16}
- 13-bit Fragment Offset specifies offset of fragmented packet, in units of 8 bytes
- Receiver reconstructs IP packet from fragments, and delivers it to Transport Layer (layer 4) after reassembly



Ping-of-Death

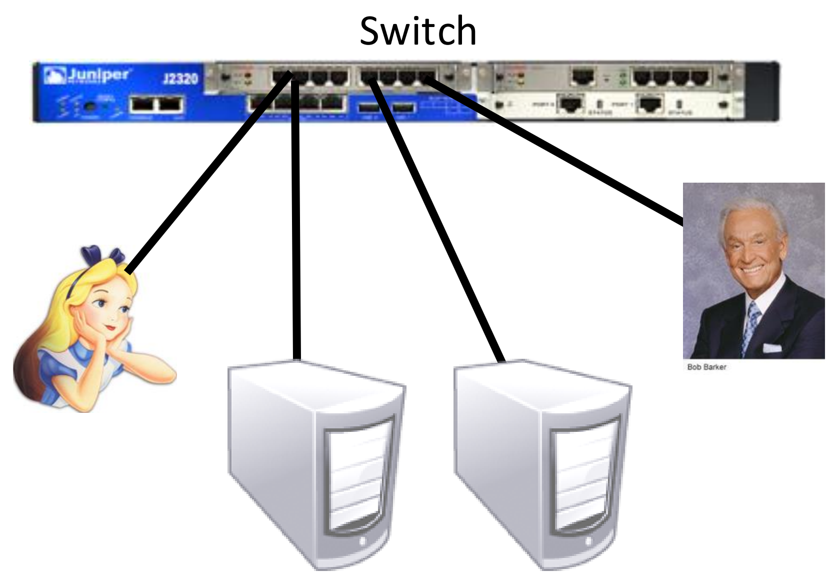
- Maximum packet size: 65,535 bytes
- Maximum 13-bit offset is $(2^{13} - 1) * 8 = 65{,}528$
- In 1996, someone discovered that many operating systems, routers, etc. could be crash/rebooted by sending a **single** malformed packet
  - If packet with maximum possible offset has more than 7 bytes, IP buffers allocated with 65,535 bytes will be overflowed
  - ...causing crashes and reboots
- Not really ICMP specific, but easy
  - % ping -s 65510 your.host.ip.address
- Most OSes and firewalls have been hardened against PODs
- This was a popular pastime of early hackers

ARP Spoofing: Background: Ethernet Frames



ARP Spoofing: Background: ARP

- **Address Resolution Protocol (ARP):** Locates a host's link-layer (MAC) address
- Problem: How does Alice communicate with Bob over a LAN?
  - Assume Alice (10.0.0.1) knows Bob's (10.0.0.2) IP
  - LANs operate at layer 2 (there is no router inside of the LAN)
  - Messages are sent to the switch, and addressed by a host's link-layer (MAC) address
- Protocol:
  - Alice broadcasts: "Who has 10.0.0.2?"
  - Bob responses: "I do! And I'm at MAC f8:1e:df:ab:33:56."



ARP Spoofing

- Each ARP response overwrites the previous entry in ARP table -- **last response wins**!
- Attack: Forge ARP response
- Effects:
  - Man-in-the-Middle
  - Denial-of-service
- Also called **ARP Poisoning** or **ARP Flooding**

ARP Spoofing: Defenses

- Smart switches that remember MAC addresses
- Switches that assign hosts to specific ports

Legacy flawed protocols and services

- Finger user identity
  - host gives up who is logged in, existence of identities

```
ip-128-239-134-5:CSCI680 adwait$ finger adwait
Login: adwait                               Name: Adwait
Directory: /Users/adwait                    Shell: /bin/bash
On since Wed Sep 27 10:27 (EDT) on console, idle 28 days 8:11 (messages off)
On since Wed Sep 27 13:56 (EDT) on ttys000, idle 14 days 3:48
On since Wed Oct 11 14:44 (EDT) on ttys001, idle 14 days 3:50
On since Thu Oct  5 12:32 (EDT) on ttys002, idle 14 days 1:07
On since Wed Oct 18 14:41 (EDT) on ttys003, idle 1 day 6:41
On since Wed Oct 25 18:35 (EDT) on ttys004
No Mail.
No Plan.

Login: adwaitnadkarni                       Name: Adwait Nadkarni
Directory: /Users/adwaitnadkarni            Shell: /bin/bash
Never logged in.
No Mail.
No Plan.
ip-128-239-134-5:CSCI680 adwait$
```

- This is horrible in a distributed environment
  - Privacy, privacy, privacy …
  - Lots of information to start a compromise of the user.

POP/SMTP/FTP

- Post office protocol - mail retrieval
  - Passwords passed in the clear
  - Solution: SSL, SSH, Kerberos
- Simple mail transport protocol (SMTP) - email
  - Nothing authenticated: SPAM
  - Nothing hidden: eavesdropping
  - Solution: ?
- File Transfer protocol - file retrieval
  - Passwords passed in the clear
  - Solution: SSL, SSH, Kerberos

Lessons Learned?

- The Internet was built for robust communication
- Smartness occurs at the end-hosts

  (see End-to-End Principle)
- Does this design support or hinder network security?

And if we had to start all over again, could we do better?