Vulnerabilities in WiFi (w/o encryption)

- Confidentiality
  - Unencrypted message read by anyone in the range and listening
- Integrity
  - Take over communication with stronger signal and forge/tamper data
- Availability
  - Forced disassociation and radio jamming (tuned to same frequency of receiver)
- Adversary: Sniffer, Rogue AP, Jammer

Radio Jammer

Unauthorized WiFi access

- *Weakness of WiFi protocol*

Pick up beacon

Beacon

Device NIC

Fake identifier (MAC)

Request to Authentication

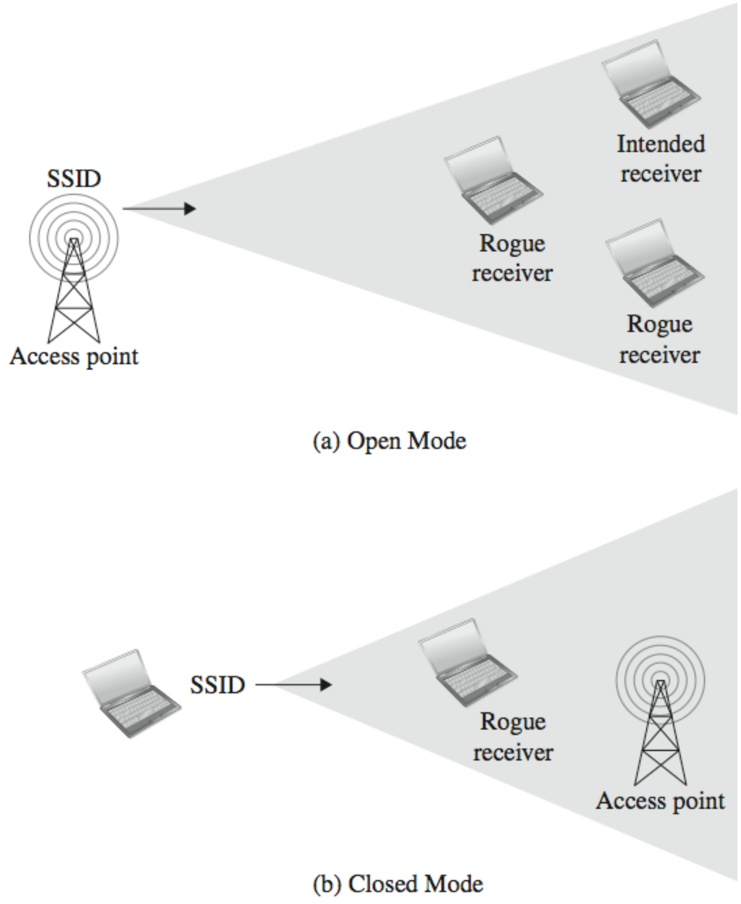
AP

Steal the association

Association establishment

Weakness of WiFi Protocol

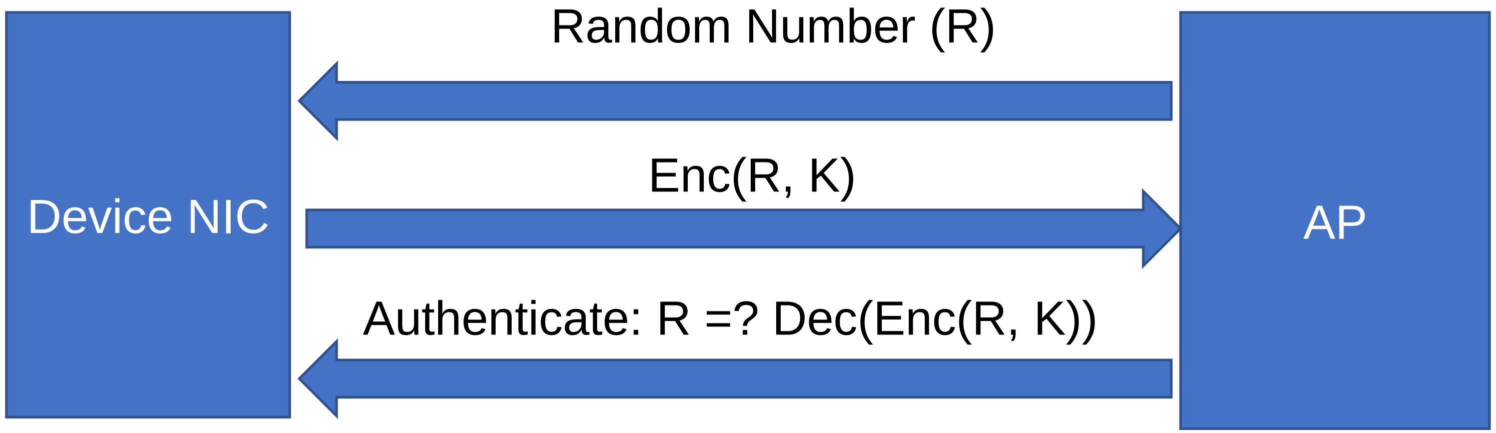
- Pick up beacon
  - Open mode: AP continuously broadcast beacon
  - Closed mode (SSID cloaking): client has to connect to AP with SSID first
  - SSID can be learnt in both cases (all frames)
  - Countermeasure (imperfect): shared temp value instead of SSID for subsequent frames
- Fake MAC address
  - Change only the network card address table
- Stealing association
  - Some vulnerable AP accepts any association



SSID sniffing

Countermeasure: WEP

- WEP (Wired Equivalent Privacy) was intended to provide privacy equivalent to wire communications.
- Published as part of original 802.11 standard in 1997
- Can verify identity (both device and AP) and protect follow-up communications
- Using one symmetric key pre-shared between client & AP (RC4)
- Authentication flow



WEP Weaknesses

- Weak encryption key
  - WEP allows to be either 64- or 128-bit, but 24 of those bits are reserved for initialization vectors (IV), thus reducing effective key size to 40 or 104 bits
  - Keys are either alphanumeric or hexadecimal string that users typed in and were therefore vulnerable to dictionary attacks
- Static key
  - User rarely changed those keys (inconvenience), one key would be used for many months/years of communications
- Weak encryption process
  - 40-bit key can be decrypted in a few minutes (WEPCrack, AirCrack-ng)
  - 104-bit key can be decrypted as well due to RC4 flaws

WEP Weaknesses (cont.)

- Weak encryption algorithm
  - Using RC4 to generate key sequence and XOR with data instead of direct encryption
  - Attacker knows the decrypted value of any single frame learns key segment
  - IV communicated in plaintext
- IV collisions
  - Only 16 million possible values of IVs
  - Predictable (some values being much more common than others)
- Faulty integrity check
  - Hash not protected by encryption

WPA

- WPA (WiFi Protected Access)
  - Designed in 2003 as a replacement for WEP
  - Quickly followed in 2004 by WPA2; Remains the standard
- Non-static encryption key (hierarchy of keys)
  - New keys generated for confidentiality and integrity of each session
  - Encryption key is automatically changed on each packet (Temporal Key Integrity Program, or TKIP)
- Better authentication
  - WPA allows authentication by password, token, or certificate

WPA (cont.)

- Strong encryption
  - WPA adds support for AES
- Integrity protection
  - WPA includes a 64-bit cryptographic integrity check
- Session initiation
  - WPA sessions begin with authentication and a four-way handshake that results in separate keys for encryption and integrity on both ends
- While there are some attacks against WPA, they are either of very limited effectiveness or require weak passwords

Attacks at Network layer

Denial of Service (DoS)

- DoS attacks are attempts to defeat a system's availability
- Goal: consume the network bandwidth/resources of victim or drop connections based on addressing
- E.g., Massive Estonian Web Failure (2007)
  - Sites of president, parliament, banks, telecom firms, etc. are down
- Examples of DoS
  - Ping flood
  - Smurf attack
  - DNS spoofing
  - Syn flood
  - …

Background: Ping

- Ping: test response time of host using ICMP protocol
- Ping requests a recipient to respond

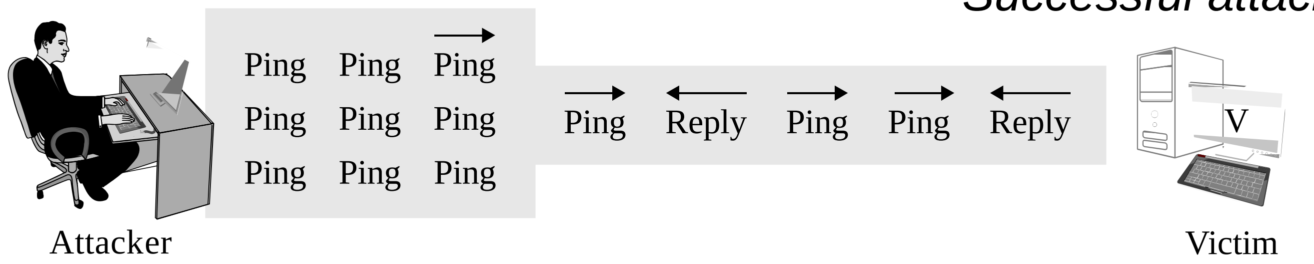
```
$ ping -c 5 www.example.com
PING www.example.com (93.184.216.34): 56 data bytes
64 bytes from 93.184.216.34: icmp_seq=0 ttl=56 time=11.632 ms
64 bytes from 93.184.216.34: icmp_seq=1 ttl=56 time=11.726 ms
64 bytes from 93.184.216.34: icmp_seq=2 ttl=56 time=10.683 ms
64 bytes from 93.184.216.34: icmp_seq=3 ttl=56 time=9.674 ms
64 bytes from 93.184.216.34: icmp_seq=4 ttl=56 time=11.127 ms

--- www.example.com ping statistics ---
5 packets transmitted, 5 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 9.674/10.968/11.726/0.748 ms
```

Ping of Death

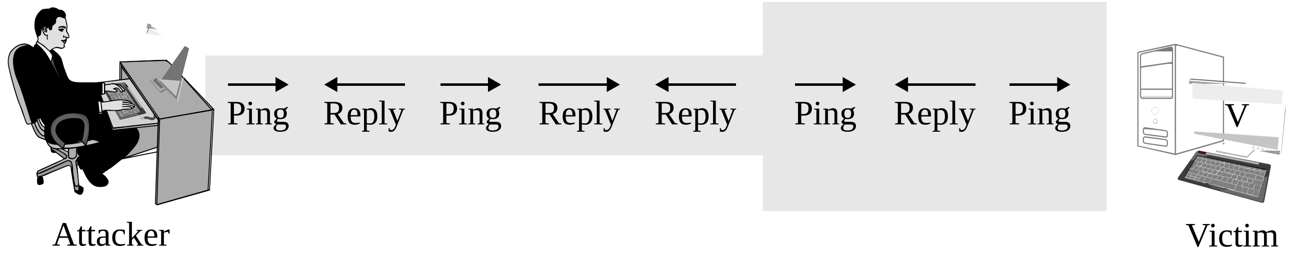
- Attacker sends a flood of pings to the victim
- Attacker's bandwidth has to be larger than victim's

Successful attack



(a) Attacker has greater bandwidth

Unsuccessful attack



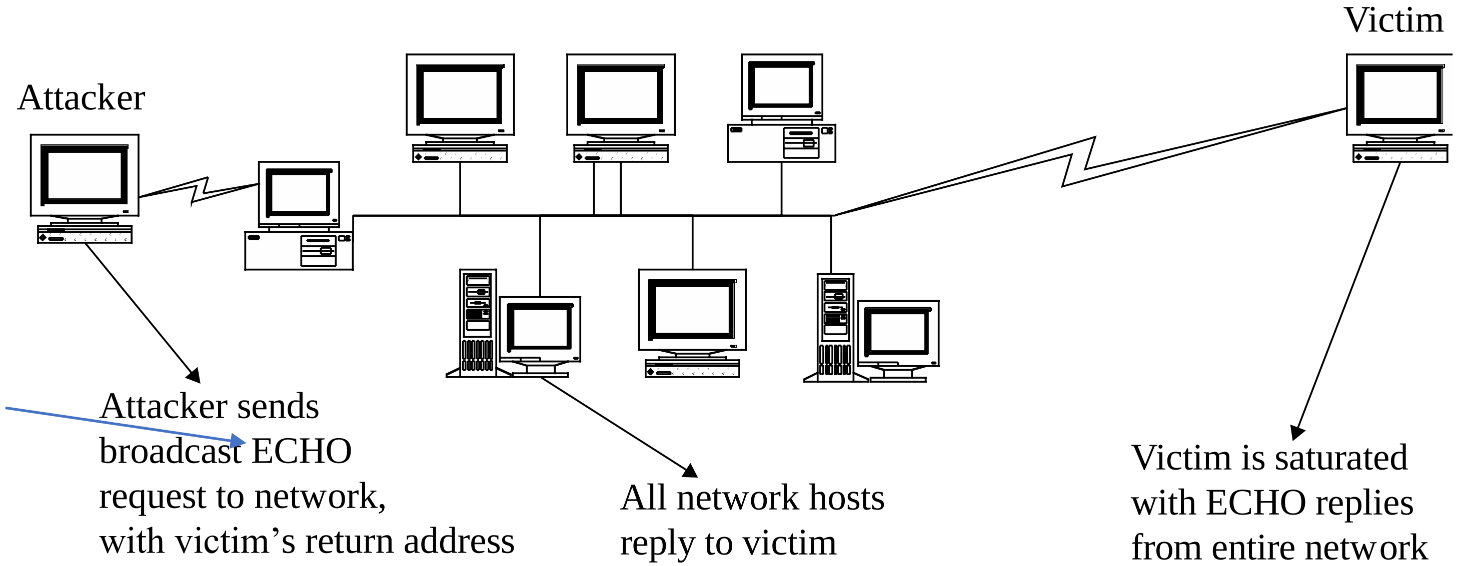
(b) Victim has greater bandwidth

Smurf Attack

Question: *how to fix?*

- A variation of ping flood
- Attacker spoofs the source address in ping packet using victim’s IP address
- Recipients have to respond to victim
- Enhanced using broadcast mode (last byte of src addr to all 1s)

Ping can be replaced with ECHO



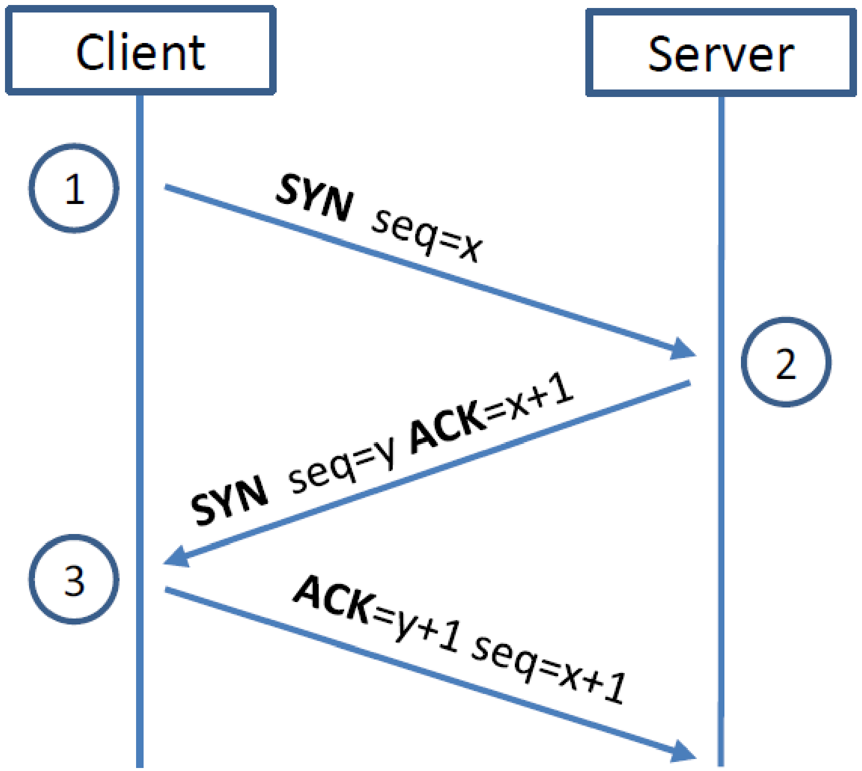
Attacks at Transport layer

UCI Samueli School of Engineering

Background: TCP Protocol

- Transport layer; sits on the top of the IP layer
- Provide host-to-host communication services for applications.
- Two transport Layer protocols
  - **TCP:** provides a reliable and ordered communication channel between applications.
  - **UDP:** lightweight protocol with lower overhead and can be used for applications that do not require reliability or communication order.

TCP 3-way Handshake Protocol



SYN Packet:

- The client sends a special packet called SYN packet to the server using a randomly generated number x as its sequence number.

SYN-ACK Packet:

- On receiving it, the server sends a reply packet using its own randomly generated number y as its sequence number.

ACK Packet

- Client sends out ACK packet to conclude the handshake

TCP 3-way Handshake Protocol

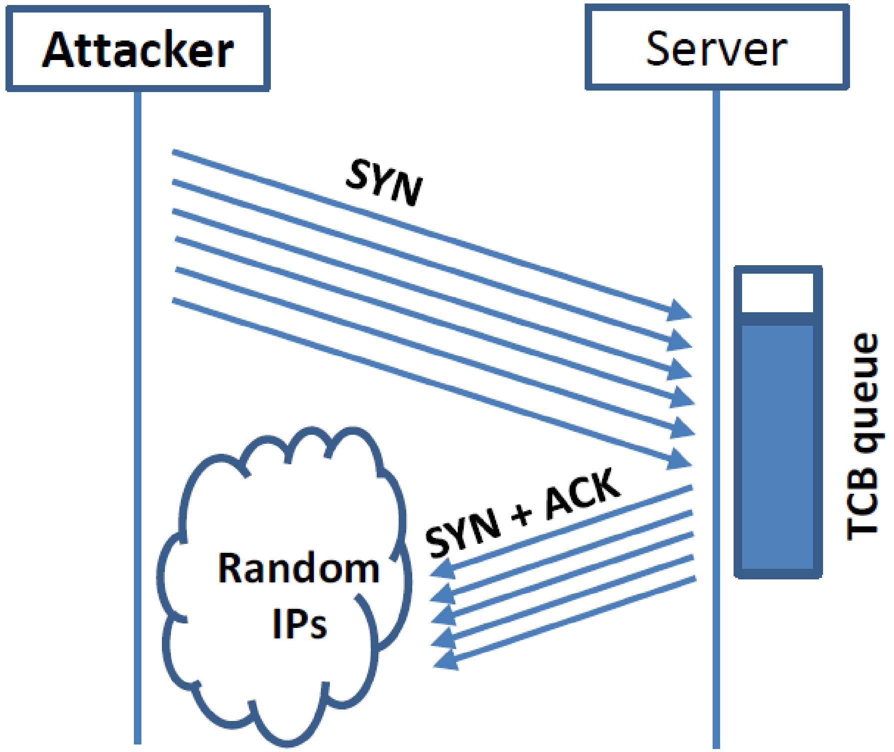
- When the server receives the initial SYN packet, it uses TCB (Transmission Control Block) to store the information about the connection.
- This is called **half-open connection** until client-server connection is confirmed.
- The server stores the TCB in a queue that is only for the half-open connection.
- After the server gets ACK packet, it will take this TCB out of the queue and store in a different place.
- If ACK doesn’t arrive, the server will resend SYN+ACK packet. The TCB will eventually be discarded after a certain time period.

SYN Flooding Attack

Idea : To fill the queue storing the half-open connections so that there will be no space to store TCB for any new half-open connection, basically the server cannot accept any new SYN packets.

Steps to achieve this : Continuously send a lot of SYN packets to the server. This consumes the space in the queue by inserting the TCB record.

- Do not finish the 3rd step of handshake as it will dequeue the TCB record.



SYN Flooding Attack

- When flooding the server with SYN packets, we need to use random source IP addresses; otherwise the attacks may be blocked by the firewalls.
- The SYN+ACK packets sent by the server may be dropped because forged IP address may not be assigned to any machine. If it does reach an existing machine, a RST packet will be sent out, and the TCB will be dequeued.
- As the second option is less likely to happen, TCB records will mostly stay in the queue. This causes *SYN Flooding Attack*.

Question: *how to fix?*