Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Spring Term 2016
Prof. Adrian Perrig

Operating Systems and Networks
Solution 10

Note: these solutions constitute material *supplemental* to the exercise sessions.

1 Port Numbers

(1) Can the source and destination ports be the same in a UDP packet?

Answer: Yes

(2) Is it possible for a web server to listen on port 79? What about port 7900? Explain what the difference is between these two cases, focusing on necessary user privileges to make this configuration possible

Answer: Yes to both ports. The former would require administrator privileges to bind to that port. The latter does not.

2 TCP Three Way Handshake

Explain, using the TCP three way handshake, why it is not possible for one of the parties to "spoof" their source IP address.

Answer: If a hosts spoofs his source address, the response will not be received by the other party. This means that either the SYN ACK will not be received, or the SYN will not be received.

3 TCP/UDP

(a) When host A sends data to host B using TCP, can it happen that two blocks of data generated by the application at A are grouped by TCP into one single IP datagram?

Answers: Yes. TCP does its own packetization.

(b) When an application receives data from TCP, can the application assume that the data was sent as one message by the source?

Answers: No.

(c) When an application receives data from UDP, can the application assume that the data was sent as one message by the source?

Answers: Yes.

(d) Assume host A sends one block of data to host B using UDP. Can it happen that the blocks of data generated by the application at A is fragmented by the IP layer at A into several IP packets?

Answers: Yes.

4 Vulnerabilities on TCP — let's be evil

The last assignment briefly covered IP address spoofing. Let's have a closer look in this exercise. Recall that when crafting IP packets, a host can specify an arbitrary source IP address, that is, instead of putting the host's correct IP address in the IP packet header, the host chooses an arbitrary address to appear in the packet header.

(a) Faking the source address and thus camouflaging the host's real identity can have several reasons. Mention at least two.

Answer: A client may want to stay anonymous while looking up sensitive content on the Internet; or a client may want to launch an attack without being held accountable.

(b) Specify what network requirements are necessary to enable address spoofing. Consider middleboxes, NATs, and ISPs.

Answer: We answer the question in a security-friendly way: In order to *disable* spoofed addresses, a network router should not accept (and forward) IP packets with IP addresses outside the network's address space. An ISP should perform similar checks. NATs usually prevent faked addresses, unless the translation process itself is compromised.

(c) Specify protocol requirements necessary to enable address spoofing. Consider particularities of TCP and UDP. How does TCP's built-in mechanism work to prevent requests with spoofed addresses?

Answer: TCP's three-way handshake ensures that only valid source addresses lead to successful connections. Sequence numbers add increased assurance that endpoints are who they claim to be. UDP has no such handshake. Faked source addresses in the IP packet would thus not be detected/prevented in the transport layer.

(d) Assume address spoofing is ruled out by the network or by the protocol. An adversary can thus not fool a server into believing a connection is established from an IP address different than the client's actual address. How can the adversary still launch an attack against the server, for instance a denial-of-service attack? Think about how (a group of) clients can exhaust the server's resources by not replying to SYN/ACK packets. How exactly does the attack work?

Answer: Every client's SYN packet increases the server's SYN queue by one entry (the server has to remember the connection request). Usually, each client's ACK packet decreases the server's SYN queue. If clients do not send the ACK packet, but instead send a huge number of SYN packets (a *SYN flood attack*), the queue will grow until no more fresh requests can be accepted, which results in a denial-of-service attack.

(e) What are possible countermeasures? Hint: The TCP protocol specification does not impose particular restrictions on the choice of the first sequence number. How can the 4 bytes of the sequence number be used in a clever way to encode the state that would usually be stored in the SYN queue?

Answer: The server chooses the sequence number to be a cryptographic 32-bit hash value over the client's IP and port, for instance the 32 first bits of `sha1(IP,port,serversecret)`, where `serversecret` is a value only known to the server. The server does not create any queue entry.

When the client responds with an ACK packet with sequence number n, the server checks whether the specified client IP address and port with the `serversecret` yield the hash value $n - 1$.

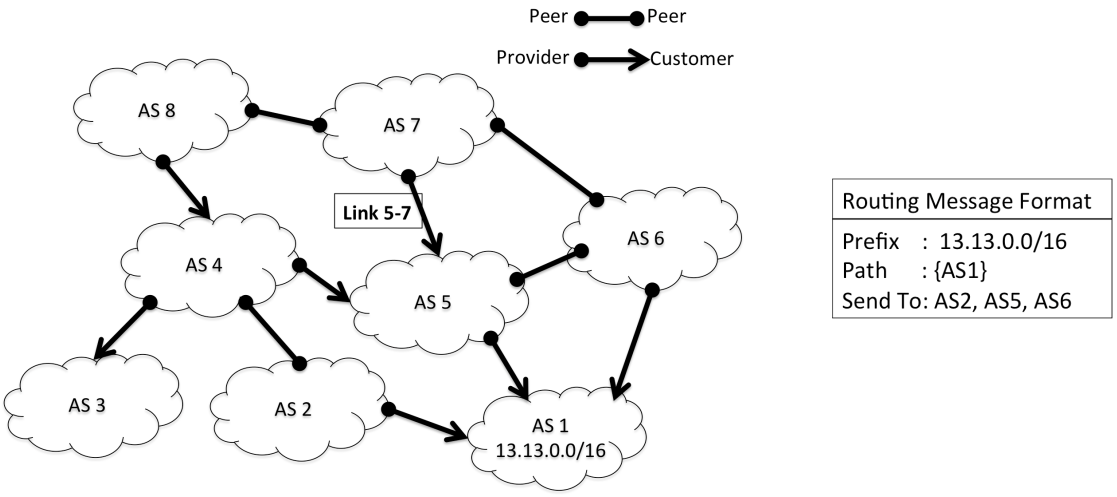
Note that additional data (possibly stored with a queue entry) can be encoded in the sequence number by reducing the length of the hash value and encoding the data instead. Usually, the maximum segment size (MSS) is the only such data that is additionally transmitted in a SYN packet during the handshake.

If freshness of SYN requests is an issue, also information about the time can be encoded in the sequence number. In this case, the time granularity of tens of seconds is fine.

A very similar approach is known in the literature under the name *SYN cookies*.

5 BGP

Consider the AS-level topology in the picture below. Say AS1 starts by announcing prefix 13.13.0.0/16 into the network at time 0. All routing messages exchanged in this problem pertain to the prefix 13.13.0.0/16. At time 1, each of AS1's neighbors – AS2, AS5 and AS6 – propagate the route to their neighbors, and so on. Hint: Note that ASes only announce paths for traffic for which they can earn money and they prefer paths that increase their revenue. Your answers should follow the format shown in the "Routing Message Format" below.



(a) Show the path vector that AS5 sends out at time 1. Which of its neighbors does AS5 send this path vector to?

Answer:
Prefix: 13.13.0.0/16
Path: {AS5, AS1}
Send To: AS4, AS6, AS7

(b) Show the path vector that AS7 sends out at time 2. Which of its neighbors does AS7 send the path vector to?

Answer:
Prefix: 13.13.0.0/16
Path: {AS7, AS5, AS1}
Send To: AS6, AS8

(c) A valid AS level path is one that is allowed by economic policies. List all possible, valid, AS level paths from AS8 to the prefix 13.13.0.0/16. Argue why {AS8,AS7,AS6,AS1} is a valid (or invalid) AS level path from AS 8 to the prefix 13.13.0.0/16?

Answer:
{ AS8, AS4, AS5, AS1 }
{ AS8, AS7, AS5, AS1 }
The path { AS8, AS7, AS6, AS1 } is invalid since AS7 would not announce the prefix 13.13.0.0/16 via AS6 to AS8.

(d) Which path will AS8 use to route to the prefix 13.13.0.0/16? Why?

Answer: AS8 will use the path through its customer AS4. The reasons are financial earnings since AS4 has to pay AS8 for traffic.

(e) Assume that Link 5–7 is removed from the topology. Would AS7 announce the prefix 13.13.0.0/16 with {AS7,AS6,AS1} to AS8? Briefly explain.

Answer: No. AS7 would act as transit AS although it is connect with peering links to both AS8 and AS6.