Closed book. Closed notes. No electronic device.

1.

For each description below, give **one** term that *best* describes it. In many cases, but not all, the term will be in the table at left. In each case, give only **one** answer of **at most** 4 words (otherwise you get zero).

Anonymity
Sender k-anonymity
Receiver k-anonymity
Authoritative nameserver
Autonomous system
BGP
DHCP
DNS query
DNS zone
Dining cryptographers
Forwarding
Internetwork
IP address
IPsec
Link
MAC address
Mitnick attack
MixNet
Nameserver
Protocol
Router
Routing
SYN cookie
SYN flooding
Time-to-live
TCP
Tor
Transport
UDP

1. The activity of a router when it transfers incoming packets to outgoing links.
   Solution: Forwarding
2. The activity by which routers select paths to destination addresses.
   Solution: Routing
3. IP resides in this layer of the Internet.
   Solution: Network layer // aka Internetwork layer
4. BGP resides in this layer of the Internet.
   Solution: Network layer // aka Internetwork layer
5. TCP and UDP reside in this layer of the Internet.
   Solution: Transport layer
6. 802.11 (WiFi) resides in this layer of the Internet.
   Solution: Link layer
7. The field of an IP packet indicating the maximum number of hops it can traverse.
   Solution: Time-to-live (TTL)
8. Routers forward an IP packet based on this field of the packet.
   Solution: Destination address
9. The transport protocol used for voice and video transfer.
   Solution: UDP
10. The transport protocol used by SSH.
    Solution: TCP
11. An independently-administered part of the Internet.
    Solution: Autonomous system (AS)
12. The protocol for selecting routes between ASes.
    Solution: BGP
13. This protocol provides "best-effort" packet transfer.
    Solution: IP // also UDP
14. What a router does when it drops any IP packet whose source address does not belong to the router's AS.
    Solution: Egress filtering
15. The standard protocol for sending encrypted integrity-protected IP packets.
    Solution: IPsec
16. This provides the mapping of domain names to IP addresses.
    Solution: DNS // Domain Name System
17. A server that can provide the IP address of any node in its zone.
    Solution: Authoritative nameserver
18. A server that knows the IP addresses of the top-level-domain nameservers.
    Solution: Root nameserver
19. The protocol by which a new host in a network obtains IP addresses for itself, DNS server, etc.
    Solution: DHCP
20. An attack in which a TCP server is overwhelmed with spurious connect requests.
    Solution: SYN flooding
21. A defense against SYN flooding.
    Solution: SYN cookie

2. What information does a host acquire from a DHCP server.

Solution IP addresses for itself, for the DNS server and for the gateway; and the lease time of its IP address.

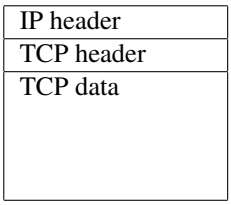
3. A local-area network that uses DHCP has an attacker that can eavesdrop and send messages. A new host, say A, joins the network and later attempts to access `https://a.com`. How can the attacker arrange to have A's attempt go instead to `https://b.com`.

Solution

- When the host joints, it does a "DHCP discover" broadcast (on its link layer, not its IP layer), and waits for the "DHCP offer" response from the DHCP server on the link.
- Attacker crafts its own response and sends it to the host before the actual DHCP server can do so. In the attacker's response, the IP address for the DNS server is set to that of a machine, say B, under attacker control.
- When the host attempts to access `https://a.com`, the host asks B (thinking it is the actual DNS server) for `a.com`'s IP address.
- B returns the IP address of `b.com`.

4.

a. The IP packet at right carries data of an application that uses TCP. Indicate the parts of the packet occupied by IP data, TCP header, TCP data and TCP sequence numbers.



IP data = TCP header + TCP data

b. Can an attacker stop the application after it is connected by tampering with just one IP packet. Explain briefly.

Solution

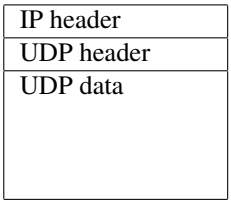
Yes. There are many ways.

1. Attacker can set the FIN bit in a TCP msg in both directions, causing each side's TCP to stop receiving. The apps may keep sending, its TCP will keep sending, but the remote TCP will not receive.
2. Attacker can change the send sequence number in a TCP msg, causing the receiving TCP to misposition the data, eventually delivering it out of order, thus corrupting the receiving application.
3. Attacker can change the data in a TCP msg, resulting in the same damage as in 2.

Note that deleting the packet or tampering with its IP header has no effect. The packet will not be delivered to the receiving TCP, so the sending TCP gets to ack, so it eventually retransmits the lost data.

5.

a. The IP packet at right carries data of an application that uses UDP. Indicate the parts of the packet occupied by IP data, UDP header and UDP data.



IP data = UDP header + UDP data

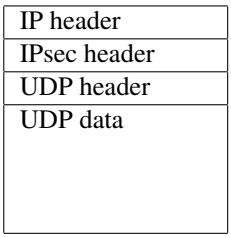
b. Can an attacker stop the application after it is connected by tampering with just one packet. Explain briefly.

Solution

No. The receiving UDP will not get the packet or its app will get a malformed packet. In any case, the app would tolerate it (otherwise it would not be using UDP).

6.

a. The IP packet at right carries data of an application that uses UDP and IPsec in ~~tunnel~~ transport mode. Indicate the parts of the packet occupied by UDP header, UDP data, IPsec header and IPsec data.



IPsec data = UDP header + UDP data
This part (most of it) will be encrypted.

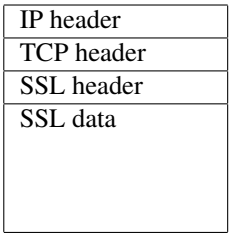
b. Can an attacker stop the application after it is connected by tampering with just one packet. Explain briefly.

Solution

No. Either the receiving IP will not get the packet or the receiving IPsec will stop the packet. So the app will not get the packet, which the app would tolerate (otherwise it would not be using UDP).

7.

a. The IP packet at right carries data of an application that uses SSL. Indicate the parts of the packet occupied by all the headers and data upto (and including) the SSL data.



IP data = TCP header + TCP data

TCP data = SSL header + SSL data
This part will be encrypted.

b. Can an attacker stop the application after it is connected by tampering with just one packet. Explain briefly.

Solution

Yes. Exactly as in problem 4b, the TCP header can be modified so that the receiving TCP gets malformed data, which it passes on to SSL. SSL detects this (stopping the malformed data from reaching the app), but SSL cannot make TCP undo the malformed data reception.

8. What is DNS cacheing?

Solution

When a host obtains a DNS record (eg, IP address of a domain name), it stores the record in a cache for a duration (specified by the record). So when the host wants resolve a domain name, it first looks in the cache and issues a DNS query only if a matching entry is not in the cache.

Without cacheing, the name resolution process would be very inefficient. [The host asks a root nameserver, which directs the host to ask a top-level-domain nameserver, which directs the host to ask the next lower-level zone nameserver, and so on.]

9. Can a DNS attack succeed in the absence of DNS cacheing?

Solution

Yes.

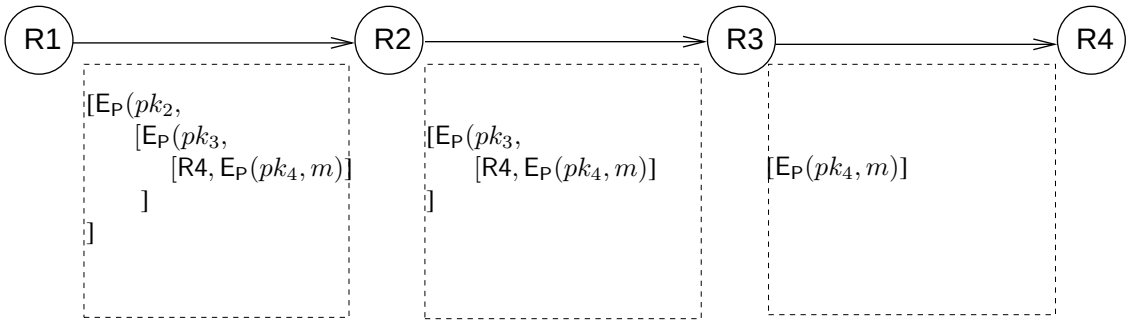
Every DNS query has a numerical "query id", used by the sender to match responses to. The attacker waits for the host to issue a DNS query. The attacker sends to the host a DNS response with an attacker-chosen IP address and a matching query id. [The attacker guesses the id based on ids of previous DNS queries from that host (how does the attacker see these). The attacker may send several DNS responses covering a range of query ids.]

If the attacker's DNS response reaches the host before an authentic DNS response, the host will use the attacker-supplied IP address and ignore the IP address in the authentic response.

But cacheing makes the attack much easier to succeed.

- Firstly, once the attacker's DNS response enters the cache, the host will use it until it expires, and ignore an authentic responses.
- Secondly, the attacker can introduce entries into the cache for domain names that the host has not yet asked for. Host will accept these DNS responses provided the response's id matches that of an outstanding DNS query, even if the domain name is different.)

10. Below is one circuit of a MixNet. Let pk_i be the public key of node Ri Node R1 wants to send message m to node R4. In the box below each link, give the message that is actually sent bover the link.



11. Give the steps to set up a hidden service X on host D in a Tor network.

Solution: See 13-s17-anonymity.pdf slides 27-32.

12. TCP uses a 3-way handshake for connection establishment:

- client sends SYN
- server receives SYN, responds with SYN+ACK
- client receives SYN+ACK, becomes open, responds with ACK.
- server receives ACK, becomes open.

Why does TCP not use a 2-way handshake as follows:

- client sends SYN
- server receives SYN, becomes open, responds with SYN+ACK
- client receives SYN+ACK, becomes open, responds with ACK.

Solution

Because the SYN received by the server may be an old duplicate (e.g., from a connection that has already closed).

12. What is congestion control. How is this achieved with ACKs in TCP.

Solution

Consider data flow from A to B that goes over a path whose links are shared with other data flows. Congestion control is what A does so that its data flow does not consume much more than its "fair" share of the buffers and bandwidth along the path A to B.

Let A be a TCP sender. A sets a limit on how much data can be *unacked* (ie, sent but not yet acked). (This limit is called the *send window*.) At the start, A will send data until the unacked amount reaches this limit. After that, A sends data only upon receiving ACKs. In particular:

- A resends data only upon timeout or upon receiving ACKs with non-increasing ack sequence numbers (indicating that B is receiving out-of-sequence data).
- A sends new data only upon receiving an ACK with an increasing ack sequence number.

Both of these indicate that data was received by B, and so is no longer occupying resources in the A–B path.

13.

a. Give the steps of the SYN flooding attack.
b. What are the steps of the SYN cookie defense.
c. Why does a SYN cookie include a slow-moving timestamp.

Solution to part a:

Normally, when a TCP server receives a SYN, it creates state (eg, initial sequence number, new thread/process for responding), responds with a SYN-ACK, and waits for an ACK response to its SYN-ACK. If no ACK comes, it keeps resending SYN-ACK until it gives up (in about 1 or 2 minutes).

SYN flooding attack: a denial-of-sevice attack where the attacker sends a continuous stream of SYN requests at a high rate (with different source ports and/or IP addresses), without attempting to receive the SYN-ACKs.

The server handles each received SYN as described above. So it gets bogged down, and cannot respond to a SYN from a legitimate client.

Solution to part b:

Instead of creating state when a SYN is received, the server sends a SYN-ACK containing a "SYN cookie", which is a hash of the SYN packet's source/destination fields and a server secret. It expects the client to include the cookie in the ACK response.

When it recieves an ACK:

- If the ACK has no cookie, it ignores the ACK.
- If the ACK has a cookie, it checks that the cookie is valid (by computing the hash of the ACK's source/destination fields and its secret, and checking that the hash equals the cookie in the ACK). Only if it is valid does it create state.

Hence the attacker is forced to receive the SYN-ACK and send an ACK response, otherwise the server does not create state.

Solution to part c:

Once the attacker receives the cookie for a SYN request, it can replay that cookie in subsequent attacks, ie, it can send a SYN followed a short time later by the ACK with cookie, without having to receive the SYN-ACK.

Including a slow-moving timestamp in the cookie's hash defends against this.

14. Give the steps of the Mitnick attack.

Solution: See 14-s17-internet-transport.pdf slides 82–95.

15. What is a defense against the Mitnick attack.

Solution: Non-guessable initial sequence numbers.

16. Give the steps of the opt-ack attack.

Solution: Consider an A–B TCP connection, where A is sending a large amount of data to B.

B does an opt-ack attack by sending ACKs early (before it has received the data). This causes A to send new data more quickly, overwhelming the resources on the A-to-B path.

15. What is a defense against the opt-ack attack.

Solution: Consider an A–B TCP connection, where B is doing an opt-ack attack.

A can defend by randomly skipping some data. If B acks the skipped data, then A closes the connection.