CS144 Finals Practice Problems Solutions
Fall 2010

Answer 1:

DHCP:

Respond to DHCP queries and change the default route to send traffic to MITM

DNS cache poisoning:

Respond to DNS queries with different data (and potentially bypass browser Same Origin Policy this way)

HTTP:

Hijack browser sessions by sniffing and stealing cookies

Answer 2:

Part (a):

Yes, per the end-to-end principle. Packet errors can occur along the way: implementation errors at the ends, memory errors, packet corruption in routers along the way. UDP's checksum alone is also too small to make strong guarantees on data integrity.

Part (b):

One simple way to do this is to send each packet multiple times, but this isn't efficient. Erasure coding techniques would allow the receivers to reconstruct missing data blocks with significantly less data transfer.

Part (c):

Because Eric's protocol sends UDP packets as fast as possible, it's possible that a router along the path is dropping packets due to full queues. The TCP connection supporting the HTTP download will sense the congestion and throttle back, while his protocol will continue sending, never giving the TCP window a chance to recover. To increase his protocol's "TCP Friendliness", Eric could add packet loss detection via ACK messages and reduce the sending rate accordingly.

Answer 3:

Part (a):

The NAT uses a Port Restricted Cone policy. Packets 1, 2, and 3 show that the NAT is not symmetric, as the mapping to A:45920 is reused. Packets 4 and 5 show that the NAT is not Full Cone, because all external hosts can use a mapping if one exists in Full Cone. Packet 6 goes from B to D, allowing D to use the NAT:27558 mapping if the NAT is Restricted Cone, but Packet 7 gets denied. D can only send to NAT:27558 after B:54429 sent a packet to D's source port, 5061, as in Packet 8. Thus, this NAT is Port Restricted Cone.

Part (b):

Yes, if Packet 7 were allowed instead, this NAT would be a Restricted Cone (not a Port Restricted Cone). Restricted Cone NATs do not have the requirement that the internal host must have previously sent to the external host on the external host's source port, only that the internal host must have previously sent to the external host on any port.

Refer to RFC 3489 for full definitions of these NAT variations.

Answer 4:
hidden terminal, different SNRs, or both.

Answer 5:
There's not enough information to determine this. The DNS for that IP address may very well have been controlled by someone else who added a PTR record that resolves the IP address to the Google hostname. If an A-record lookup on client-42.ext.google.com yields the hit's IP address, there's a good chance the hit is from Google.

Answer 6:
No. It doesn't. To start with, consider using MOD_MAC on short messages (less than 80 bits). An attacker sending a single packet knows that the message's corresponding MAC is only 1 bit long, and has a 50% chance of impersonating the sender with a single packet (and could easily send 2 packets, one with MAC=1 and another with MAC=0, guaranteeing one would be accepted).

Similarly, portions of longer messages can easily be replaced. Recall a change in any bit of the message should affect any bit of a good MAC with equal probability. This is not the case with MOD_MAC. Consider a 240 bit long message. If an attacker makes a change to any of the message located from bits [0,79], he/she must only worry about the first bit of the mac created by MOD_MAC. Similarly, if the attacker makes a change to any of the message located from bits [80,159], he/she must only worry about the second bit of the mac created by MOD_MAC.

Answer 7:
Part (a):

The ETX is 3.

Part (b):

Cannot answer from the given information. The ETX is the number of transmissions for a packet to get from source to destination. However, we do not know whether drops on the links are dependent. Consider a case where links SRC:C and D:DST drop packets randomly and independently of each other with a 50% chance, however, the link C:D always drops a packet if the link SRC:C does not. In such a case, a packet will never make it from the source to the destination. This question is very subtle. It takes advantage of a common assumption that researchers make -- independence of links -- that isn't always true. Don't feel awful if you missed it.

Part (c): ****UPDATED***

The ETX of the entire path is the sum of the cost of each link.

Cost src-c: 1/(PRR) = 1/(1-p_drop)= 1/(2/3) = 3/2;
Cost c-d: 1/(PRR) = 1/(1-p_drop)= 1/(1/2) = 2;
Cost d-dst: 1/(PRR) = 1/(1-p_drop)= 1/(4/5) = 5/4;

==> ETX path = 3/2+2+5/4= 4.75.

Answer 8:
This algorithm isn't very good. Every time a new machine is added or removed, lots and lots of records must be moved. Alex should use consistent hashing instead. (Recall lecture 12, or visit the above website for more information.)

Answer 9:

Changing the bit rate allows the signal the adapt to changing amounts of SNR on the channel. If the SNR on a channel decreases, by reducing the bit rate, and hence increasing the strength of error correction, the link can remain usable. If the bit rate were not changed, it is possible that the data on the line could become incomprehensible. Likewise, if the channel conditions are good, by increasing the bit rate, and reducing noise tolerance, the link can support a higher bandwidth of data.

Answer 10:

Part (a):

Packets will be dropped. More packets arrive in the router than can be serviced. The output rate of the router is 60 packets per minute. The input rate from A and B is 120 packets per minute each, giving an overall input rate of 240 packets per second. That means, every minute, 180 packets must be dropped.

Part (b):

59 or 60 packets will be from A. 0 or 1 packets will be from B. One can see this in the following way. Whenever B sends a burst of packets, there is at most one spot open on the internal queue. That means every minute, at most 1 packet will be from B.