Homework 3: Caught in the Net

Due: Friday, April 19 @ 11:59 pm EDT

Overview and instructions

This homework has only two problems. There is no extra CS1620/CS2660 component.

Note on collaboration

You are welcome (and encouraged!) to collaborate with your peers, but the solutions you write down must be **your own work** (ie, written by you). You are responsible for independently understanding all work that you submit—after discussing a problem as a group, you should ensure that you are able to produce your own answers independently to ensure that you understand the problem. For more information, please see the course Collaboration Policy.

In your submission, we ask that you include a brief *collaboration statement* describing how you collaborated with others on each problem—see the next section for details.

How to submit

You will submit your work in PDF form on Gradescope. Your PDF should conform to the following requirements:

- **Do not** include any identifying information (name, CS username, Banner ID, etc.) in your PDF, since all homeworks are graded anonymously
- Each problem (where "problem" is one of the Problems 1–2) should start on a separate page. When you submit on Gradescope, you will be asked to mark which pages correspond to which problem
- At the start of each problem, write a brief *collaboration statement* that lists the names and CS usernames of anyone you collaborated with and what ideas you discussed together
- If you consulted any outside resources while answering any question, you should cite them with your answer

Problem 1: Local network eavesdropping

Relevant lectures: Lectures 18–19

Consider the network represented in Figure 1, a subnet whose addresses all take the form `192.168.1.*` (ie, the subnet 192.168.1.0/24.) Each router and host is labeled with its IP address and MAC address. In all parts of this problem, assume that all hosts (Host A, Host B, and Host C) and the router have entries for all other hosts on the subnet in their respective ARP tables, and thus no device in Figure 1 is actively sending any ARP messages.

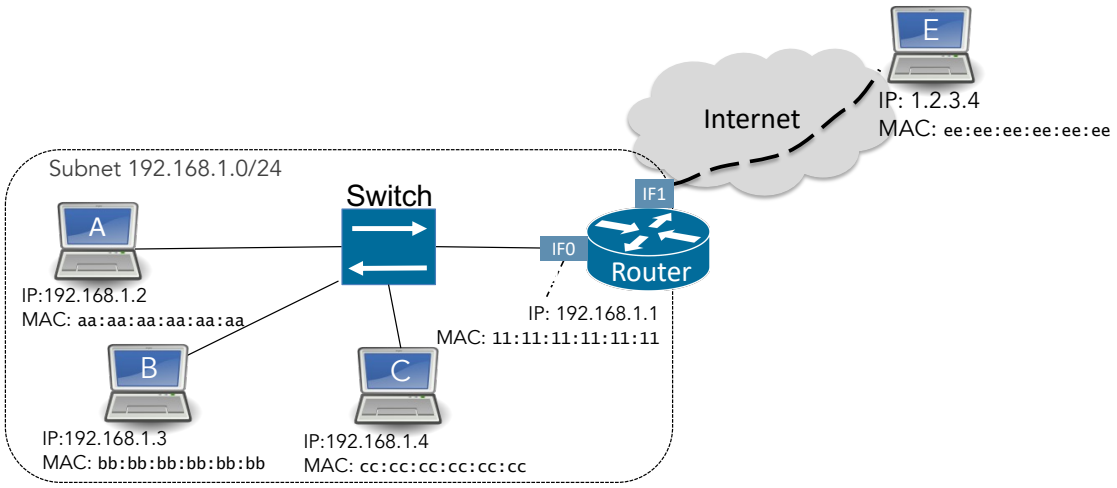

Figure 1: An example network.

Question a) (4 pts) Host B wants to intercept traffic between Host A and Host C. How could B use ARP to cause A to send its traffic to B instead of C? In 1–2 sentences, explain what ARP message(s) you would send, what they would contain, and to which hosts you would send them.
Note: In carrying out this attack, B needs to be careful not to accidentally break other hosts' connections. How can B make sure that their attack only targets A?

Question b) (4 pts) Host B's attack is successful and it's now intercepting traffic sent by Host A to Host C, but this means that Host C isn't getting the traffic. As a result, A may notice that something is going wrong and stop sending data, which won't give you the information you need. How could B make sure that the communication between A and C is retained, while B can intercept it?

Question c) (4 pts) Assume that Host B's attack was successful and it is now intercepting Host A's traffic, and Host C is now receiving the traffic and responding as normal—so A is not aware of the attack. However, B would also like to intercept **Host C's responses to Host A**. How can B accomplish this? Similar to part (a), explain your attack and why it works.

Question d) (4 pts) How would these techniques differ if Host B wanted to intercept Host A's communication with Host E, which is somewhere on the internet (ie, not on this subnet). Once again, we don't want to break the communication between Host A and Host E (ie, similar to part (b). In your response, try to be precise about the content of any attack packets you would send.
Hint: Since Host E is not on B's subnet, it will not suffice to spoof E'S MAC address.

Sample TA Solution

a) Host B can repeatedly send ARP responses directly to A's MAC address (`aa:aa:aa:aa:aa:aa`), specifying that the MAC address corresponding to Host C's IP address (`192.168.1.4`) is Host B's MAC address (`bb:bb:bb:bb:bb:bb`). This way, only A's ARP cache is poisoned, while the ARP caches of

other machines on the network remain unaffected. So when IP packets are sent from A's IP address to C's IP address, the resulting ethernet frames will be sent to B's MAC address.

- 4/4 pts: Full credit: recognizes that B needs to send an ARP packet saying "`192.168.1.4` is at <**mac address of B**> to host A, and provides reasonable explanation
- 3/4 points: Correct intuition, but something important missing
- 2/4 points: Correct intuition, but multiple important things missing
- 1/4 points: Incorrect intuition, major components misssing
- 0/4 points: No credit/missing

Aside: Note that gratuitous ARPs are usually sent to the broadcast address (`FF:FF:FF:FF:FF:FF`), but they don't have to be, as shown in the paragraph above. If sent to the broadcast address, all of the Hosts' ARP caches will associate B's MAC address with C's IP address, and so when any IP packets are sent to C's IP address, the resulting ethernet frames will be sent to B's MAC address.

This sort of ARP response—one that is not in response to a request—is known as a "gratuitous ARP," and is a valid use of ARP (that is, the protocol specifies that hosts should update their ARP caches in response to gratuitous ARP replies). This technique is known as "ARP cache poisoning."

b) (4 pts) Host B can forward packets it receives from A that are destined from C to Host C. In particular, this means that any IP packet that arrives in an Ethernet frame whose source address is A's MAC address, and where the destination IP address is C's should be forwarded to C. B can do this by creating a new ethernet frame with the same IP packet and sending it to C's MAC address.

- Same grading scale as above. For full credit, must recognize that B needs to forward packets, mentions specifically the need to forward packets destined for host A to host C

c) (4 pts) Host B can perform the same attack in reverse—poison C's ARP cache to believe that the MAC address corresponding to A's IP address is actually B's MAC address, and then forward any intercepted IP packets to A.

- Same grading scale as above. For full credit, must recognizethat B needs to send an ARP packet saying "`192.168.1.2` is at <**mac address of B**> to host C, and provides reasonable explanation
- Same grading scale as above. For full credit, must recognize that B needs to forward packets, mentions specifically the need to forward packets destined for host A to host C

d) (4 pts) Since `1.2.3.4` is not in the local subnet, Host A (as described in problem 2) will need to learn the MAC address of the default gateway (`192.168.1.1`), and the IP packets will be encapsulated in ethernet frames and sent to that MAC address. Thus, B will need to instead poison A's ARP cache to associate the default gateway IP address with B's MAC address, and then forward any intercepted traffic to the gateway. In order to get the return traffic, B will need to poison the router's ARP cache to associate A's IP address with B's MAC address so that response IP packets (whose destination IP address will be A's) will also be sent to B.

- Same grading scale as above. For full credit, must recognizethat host E is not in the local subnet, so it's necessary to spoof *the router*'s MAC address (ie, redirect traffic from A to the router) as well as redirecting traffic from the router to A.

Problem 2: Thinking about DNS

Relevant lectures: Lecture 20

Question a) In the DNS lecture, we discussed how each DNS query has a `request identifier`, also called a transaction ID, to identify responses to individual queries. Consider the following questions about how DNS transaction IDs are used (your answers should be no more than 100 words for each):

(i) Why is using a transaction ID more secure than using *no* transaction ID?

(ii) Why is using *randomized* transaction IDs is more secure than having *sequential* transaction IDs.

Question b) Imagine you control the default DNS server for an Internet Service Provider (ISP). In 2–3 sentences, briefly explain how you can attempt to leverage DNS to block access to sites you don't want your customers to access.

Sample TA Solution

a) See subparts, 3pts each: 3/3 if answer is complete, 2/3 if minor details missing, 1/3 for multiple issues, 0/3 for major conceptual problem or answer missing.

(i) An attacker needs to guess the query ID that corresponds to a given DNS request in order to trick the requester.

Long-form explanation (more info than required): DNS query IDs ensure that DNS responses correspond to a given DNS request. In particular, they ensure that a malicious attacker cannot forge a malicious DNS response to a given DNS request—since a DNS ID takes on one of 65,536 distinct values (a 16-bit number), an attacker would have to guess the query ID that corresponds to a given DNS request in order to trick the requester into accepting their malicious DNS response. Without query IDs, an attacker would not have to go through this process.

(*Source:* `https://www.sans.org/reading-room/whitepapers/dns/achilles-heal-dns-565`)

(ii) Sequential IDs allow attackers to guess the ID number for subsequent queries if they can determine ID of a past query; randomized IDs prevent this.

(*Source:* `https://www.sans.org/reading-room/whitepapers/dns/achilles-heal-dns-565`)

b) If you are an ISP, you can do the following:

- Run your own DNS server that gets advertised to customers that blocks domains you don't like (ie, respond with error, or redirect to wrong server)
- Intercept/modify DNS traffic sent to your customers (ie, DNS hijacking, or outright blocking of DNS traffic)

4pts, total, graded on the following scale (adjust if necessary):

- 4/4 pts: Full credit: mentions at least one of these
- 2/4 points: Half-credit: Correct intuition, but something important missing
- 0/4 points: No credit/missing