

PRINT your name: _____, _____
(last) (first)

SIGN your name: _____

PRINT your class account login: cs161-_____

Your TA's name: _____

Your section time: _____

Name of the person
sitting to your left: _____

Name of the person
sitting to your right: _____

You may consult one sheet of paper (double-sided) of notes. You may not consult other notes, textbooks, etc. Calculators and computers are not permitted. Please write your answers in the spaces provided in the test. We will not grade anything on the back of an exam page unless we are clearly told on the front of the page to look there.

You have 80 minutes. There are 6 questions, of varying credit (100 points total). The questions are of varying difficulty, so avoid spending too long on any one question.

Do not turn this page until your instructor tells you to do so.

Problem 1	
Problem 2	
Problem 3	
Problem 4	

Problem 5	
Problem 6	
Total	

Problem 1. [True or false] (14 points)

Circle TRUE or FALSE. Do not justify your answers on this problem.

- (a) TRUE or FALSE: SQL injection vulnerabilities can be avoided by applying the following algorithm to every untrusted input: replace every instance of & with `& ;`, replace every instance of < with `< ;`, replace every instance of > with `> ;`, and replace every instance of " with `" ;`.

- (b) TRUE or FALSE: The same-origin policy generally allows JavaScript from `berkeley.edu` to read cookies associated with `berkeley.edu`.

- (c) TRUE or FALSE: The same-origin policy generally allows JavaScript from `berkeley.edu` to read cookies associated with `stanford.edu`.

- (d) TRUE or FALSE: A benefit of privilege separation is that it can provide an opportunity to reduce the size of the TCB.

- (e) TRUE or FALSE: If we ensure that the attacker does not have permission to read the value of session cookies stored by the victim's browser, session fixation attacks become impossible.

- (f) TRUE or FALSE: It is easier to passively eavesdrop on UDP traffic than on TCP traffic.

- (g) TRUE or FALSE: Ingress filtering refers to carefully escaping meta-characters in URLs.

- (c) The Fortune 500 company FooCorp has an internal web application that its employees can use to fill out travel vouchers. Unfortunately, FooCorp's system administrators have recently discovered that the voucher web application has cross-site request forgery (CSRF) vulnerabilities. FooCorp has a firewall that blocks all inbound connections from the external world to FooCorp's internal network, but allows all web connections initiated from machines on FooCorp's internal network.

Does FooCorp's firewall prevent exploitation of the CSRF vulnerabilities in its travel voucher application? Circle yes or no, then briefly explain (in one line or less).

- (i) Yes. (ii) No.

Justification:

- (d) Suppose we are building a web application that asks the user for their email address and stores it in a variable `m`. We want to invoke the shell to send an email message to the email address `m`, like this:

```
void sendemail(char *m) {
    char cmd[1024];
    snprintf(cmd, sizeof(cmd), "mail %s", m);
    f = popen(cmd, "w");
    ...
}
```

However before we invoke `sendemail(m)`, we want to ensure that `m` is safe to use with this code. Which of the following would be the best way to do that? Circle the best (safest) answer.

- (a) Check that `m` does not contain any of the following characters: `*|'()`.
- (b) Remove all instances of the following characters from `m`: `*|'()`.
- (c) Check that `m` starts with a letter (a-z or A-Z) and is composed solely of the following characters: `abc...zABC...Z0123...9@+_-...`
- (d) None of the above: This code cannot be made safe, no matter what checks you do on `m`.

Explain why briefly (one line or less):

Problem 3. [Printer discovery] (16 points)

A consortium of printer vendors have come up with a great new protocol to help users automatically discover the set of printers on their local network. In this protocol, when the user wants to print something, the user's computer automatically broadcasts a Printer Discovery packet. A Printer Discovery packet is a UDP packet whose destination address is the broadcast address, and whose source and destination port is 56184. Because this is a broadcast packet, every host on the local network will receive it.

Printers constantly listen for Printer Discovery packets. Any time that they receive one, they immediately respond with a Printer Announcement packet. A Printer Announcement packet is a UDP packet whose destination address is the broadcast address, and whose source and destination port is 56185; its payload identifies the name of the printer, the printer's IP address, and any special options supported by the printer (e.g., 2-sided printing, color printing). The Printer Announcement packet is broadcast to the entire network, so that other hosts on the local network can also learn about this printer.

Whenever a machine receives a Printer Announcement packet, it checks that the source address of the packet matches the printer's IP address found in the payload. In case of a mismatch, it ignores the packet. Otherwise, it accepts the packet and adds this printer to its list of known printers. If the machine's list of known printers already contains a printer with the same name, the machine overwrites the previous entry in its list with the information found in the newly received packet.

Vicky the Victim is about to connect her laptop to a local switched Ethernet network. Her laptop will use this printer discovery protocol to look for a printer, and then Vicky will connect to one of the printers found in this way and send it a sensitive corporate document to be printed. Meanwhile, Attila the Attacker's computer is attached to this same network. Attila has the ability to inject packets onto this network and to receive all broadcast packets, but he cannot eavesdrop on other traffic. The printers are in locked rooms that Attila does not have access to, and Attila has not been able to hack or access any of the machines or printers attached to this network, so his only hope is to attack the printer discovery protocol.

- (a) Can Attila arrange to learn the contents of Vicky's document, without physically accessing any of the printers? Circle either "yes" or "no", then briefly justify your answer. If you circle "yes", describe the attack; if you circle "no", explain why this kind of attack is not possible.

(i) Yes. (ii) No.

Justification:

- (b) Can Attila modify what is printed on the printer? In other words, Attila wants to *replace* Vicky's chosen document with something else Attila has chosen, hopefully without Vicky noticing. It's not acceptable if Vicky's original document gets printed in addition to Attila's replacement, because then Vicky might notice and get suspicious; Attila is only interested in an attack that causes his document to be printed *instead of* Vicky's. Can Attila mount such an attack, without physically accessing any of the printer? Circle either "yes" or "no", then briefly justify your answer. If you circle "yes", describe the attack; if you circle "no", explain why this kind of attack is not possible.

(i) Yes. (ii) No.

Justification:

Problem 4. [DDoS] (19 points)

This question asks you to consider a (hypothetical) anti-spam company called Turquoise Security Inc. Turquoise Security uses a vigilante approach to fighting spam: when one of Turquoise Security's users identifies an email they've received as spam, Turquoise Security's servers automatically visits all the websites advertised in the spam message and leaves generic complaints on those websites. Turquoise Security operates on the assumption that as their user base grows, the flow of complaints from hundreds of thousands of computers will apply enough pressure on spammers and their clients to convince them to stop spamming.

Yesterday, Turquoise Security's web site came under a massive DDoS attack using a variety of techniques. The attackers are using DNS amplification: the attackers identified several third-party DNS servers that will respond to any DNS query, and are sending many spoofed DNS queries to those DNS servers with a forged source address. In particular, each query is sent in a spoofed UDP packet, where the source address on each of these DNS queries is forged to be the IP address of Turquoise Security's web server. Also, each query has been chosen so that it will trigger a response that is much larger than the query itself, amplifying the effect of the attack. This attack has overloaded Turquoise Security's web server with huge amounts of traffic.

- (a) Consider the packets that Turquoise Security's web server received as a result of this DNS amplification attack. For each of the following fields in the IP header, state whether you expect that field to be the same for all of these packets or to differ from packet to packet (circle one choice). If you select "same", describe the value of that field, in the space to the right. If you select "differs", briefly justify your answer, in the space to the right.

Source address: (i) same: _____

(ii) differs: _____

Destination address: (i) same: _____

(ii) differs: _____

Source port: (i) same: _____

(ii) differs: _____

Destination port: (i) same: _____

(ii) differs: _____

- (b) Suppose that Turquoise Security's ISP, the company who provides Turquoise Security with their Internet connection, would like to help Turquoise Security survive this attack. Name one thing that their ISP could do in response to this attack, to relieve the load on Turquoise Security's servers for now.

(continued on next page)

(c) Today, DNS servers accept queries via the UDP protocol. But imagine that DNS had been designed differently, so that DNS used only TCP (not UDP) and DNS servers accepted queries only via TCP (ignoring all UDP packets). Would this make the DNS amplification attack described above easier, harder, or have no effect? Circle one answer, then briefly explain your answer (in one line or less).

(i) The attack would be easier. (ii) No effect. (iii) The attack would be harder.

Justification:

(d) Setting aside the current DDoS attack on Turquoise Security, how could the Turquoise Security service itself be used to mount a DoS attack on others?

Problem 5. [Firewalls and NATs] (14 points)

(a) Can a stateless firewall (such as a packet filter) enforce the following policy?

Policy: Block TCP connection initiation requests from any external host to any internal host. Allow TCP connection initiation requests from any internal host to any external host, and also allow returning traffic on these connections initiated by internal hosts.

You may assume that the internal hosts (those on the inside of the firewall) all have IP addresses of the form 128.32.153.x, where the x can be anything in the range 0–255, and no external host has an IP address of this form. You may assume that the TCP/IP stack on every internal host operates correctly.

Circle “yes” or “no”, depending on whether you think a stateless firewall (such as a packet filter) can enforce the policy above or not, then briefly explain your answer (in one line or less).

(i) Yes, it can enforce the policy. (ii) No, it cannot.

Justification:

(b) Name one security benefit that NAT provides.

Problem 6. [Secure coding] (15 points)

Consider the following C code:

```
/* Information about the current CD. */
struct cd {
    int numtracks; /* The number of tracks on this disc. */
    int tracklen[16]; /* The length of each track on the disc, in seconds. */
    void (*notify)(struct cd *); /* Call this whenever the CD info changes. */
};

struct cd *curcd = makestructcd();

/* Update the length of track number 'track'. */
void update_cdinfo(int track, int newtracklen) {
    if (track > 16)
        return;
    curcd->tracklen[track] = newtracklen;
    (curcd->notify)(curcd);
}
```

(Don't worry about `makestructcd()`; it just allocates and initializes a `struct cd`.) Assume the adversary can arrange for `update_cdinfo()` to be called with whatever values of `track` and `newtracklen` he likes (those values may have been read directly off the CD, for instance). Answer the following questions about this code, *concisely*:

- (a) What is the security vulnerability in this code?

- (b) How could an attacker exploit this vulnerability to trigger the execution of malicious code? Describe how the attacker should choose the values of `track` and `newtracklen`.