CS 155 Final Exam

This exam is open books and open notes. You have 2 hours. Make sure you print your name legibly and sign the honor code below. All of the intended answers may be written well within the space provided. You may use the back of the preceding page for scratch work. If you want to use the back side of a page to write part of your answer, be sure to mark your answer clearly.

The following is a statement of the Stanford University Honor Code:

A. *The Honor Code is an undertaking of the students, individually and collectively:*

   (1) *that they will not give or receive aid in examinations; that they will not give or receive unpermitted aid in class work, in the preparation of reports, or in any other work that is to be used by the instructor as the basis of grading;*

   (2) *that they will do their share and take an active part in seeing to it that others as well as themselves uphold the spirit and letter of the Honor Code.*

B. *The faculty on its part manifests its confidence in the honor of its students by refraining from proctoring examinations and from taking unusual and unreasonable precautions to prevent the forms of dishonesty mentioned above. The faculty will also avoid, as far as practicable, academic procedures that create temptations to violate the Honor Code.*

C. *While the faculty alone has the right and obligation to set academic requirements, the students and faculty will work together to establish optimal conditions for honorable academic work.*

I acknowledge and accept the Honor Code.

____
(Signature)

____
(Print your name, legibly!*)*

Prob	# 1	# 2	# 3	# 4	# 5	# 6	# 7	# 8	Total
Score									
Max	12	16	10	14	9	15	15	9	100

1. (*12 points*) .... General attacks

Describe the following general kinds of attacks, in a few sentences each. Say what kind of user or system is being attacked, what point of access the attacker requires, and how the attack proceeds.

(a) (*4 points*) social engineering attack

(b) (*4 points*) cross-site scripting attack

(c) (*4 points*) DNS cache poisoning

2. (*16 points*) .... Short Answer

(a) (*4 points*) One of the homework problems asked about finding source code instructions that free a buffer twice. Why is double-free a security problem?

(b) (*4 points*) What is the principle of least privilege? Why is it important?

(c) (*4 points*) TCP packet sequence numbers may follow a predictable order, such as 1, 2, 3, ..., or a less predictable one. What is the purpose of sequence numbers? Why are unpredictable sequence numbers beneficial?

(d) (*4 points*) You are setting up a home computer for a friend and you want to provide some firewall protection. One option is a small hardware device that operates as a stand-alone firewall. Another is firewall software that runs on your friend's computer. What are the relative advantages of each?

3. (*10 points*) .... Weak Password Checking

What's wrong with this password-checking code, resembling the code used in an early version of Unix? Describe a *simple* attack that allows you to log in. Assume this code is compiled using StackGuard. LEN is some constant, say 8.

```
void check_passwd(char *name, passwd) {
    char buffer1[LEN];
    char buffer2[LEN];
    /* omitted instructions place password for name in buffer 1 */
    strcpy(buffer2,passwd);
    if (!memcmp(buffer1, buffer2, LEN)) {
     /* allow login */
    }
    else { /* disallow login */
    }
}
```

4. (*14 points*) .... Upside-down stack

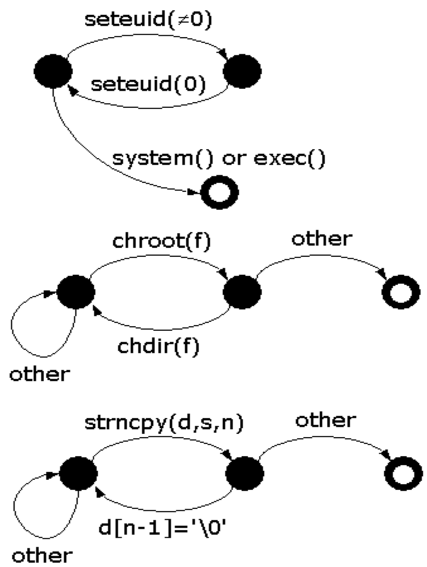
Under UNIX, as discussed in class, the run-time stack begins at a high memory address and grows "downward" in memory. In other words, when a function `f` is called, the address of the activation record for `f` is a lower address number than the activation record for the caller of `f`.

(a) (*7 points*) Explain why the order of activation records in memory, and the relative position of entries in an individual activation record, is relevant to buffer overflow attacks.

(b) (*7 points*) What if the stack grows the other direction and the organization of each individual activation record is changed? Are buffer overflow attacks still possible? Explain.

5. (*9 points*) .... Prudent coding practices

Here are three automata expressing rules for prudent coding practices. In each case, the checker begins in the state on the left and report an error if the state that is not drawn as a solid circle is reached. Explain what each automaton checks for and why this is a good coding rule to follow.



6. (*15 points*) .... Certificates and Public-Key Infrastructure

John Smith creates a public-private keypair, and obtains from VeriSign a personal certificate for *John Smith* on his public key. He then creates a second public-private keypair, and uses his *John Smith* private key to create a certificate on the second public key, supposedly for `www.amazon.com`.

Now, John, being the malicious sort, intercepts an SSL connection to Amazon.com. John presents his forged certificate for `www.amazon.com` to the intercepted user. The intercepted user's browser recognizes the certificate for the fake Amazon site as legitimate because there is a valid certificate chain; but John can decrypt the communication because he knows the alleged *www.amazon.com* certificate's corresponding private key.

(a) (*5 points*) What's wrong with this scenario?

(b) (*5 points*) Assuming the attack works as described, who is at fault?

(c) (*5 points*) How can we fix this problem?

7. (*15 points*) .... Firewalls

Suppose an administrator wants to enforce the following policy with a packet-filtering firewall: the firewall should block all incoming HTTP responses that contain the word "APPLET".

(a) (*5 points*) Explain how a packet filtering firewall can be used to enforce this policy.

(b) (*5 points*) Consider an attacker who tries to bypass firewall enforcement by exploiting IP packet fragmentation by sending overlapping packets. Explain how this attack works and explain how the firewall can defend against it.

(c) (*5 points*) What other common firewall design would make enforcing this policy easier? Why?

8. (*9 points*) .... Bugbear and SMTP

Mail on the internet is transmitted using SMTP, the Simple Mail Transport Protocol.

A Windows user who wishes to send mail to, say, *bob@aol.com* does not connect directly to AOL's SMTP server. Instead, she transmits the e-mail message to a local SMTP server that sends the message to AOL on the user's behalf. For Stanford users, this local relaying server is *smtp.stanford.edu*.

The Bugbear.B worm currently rampaging through Stanford uses SMTP to send copies of itself to addresses in an infected computer's address book. While Bugbear.B could examine the user's setting for SMTP relay server and use that server to send mail, Bugbear.B instead includes its own SMTP code, and sends mail without recourse to the relay server. In other words, if Bugbear.B chooses to send mail to *bob@aol.com*, it contacts AOL's mail servers directly, not *smtp.stanford.edu*.

(a) (*5 points*) What propagational advantage does Bugbear.B gain by not relaying through a user's configured SMTP relay server?

(b) (*4 points*) How might Stanford's administrators go about preventing Bugbear.B's propagation through e-mail?