Wagner Spring 2014 — CS 161 Computer Security — Final Exam

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

I am aware of the Berkeley Campus Code of Student Conduct and acknowledge that any academic misconduct on this exam will lead to a "F"-grade for the course and that the misconduct will be reported to the Center for Student Conduct.

SIGN your name: _____

PRINT your class account login: `cs161-`_____ and SID: _____

Your TA's name: _____

Name of the person sitting to your left: _____

Name of the person sitting to your right: _____

You may consult three sheets of notes (each double-sided). You may not consult other notes, textbooks, etc. Calculators, computers, and other electronic devices 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 180 minutes. There are 14 questions, of varying credit (200 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 ***True or False*** **(17 points)**

Circle True or False. Do not justify your answer.

(a) True or False: The SHA256 hash of a user-generated passphrase makes a good key for an encryption scheme because hashes are random.

(b) True or False: The reason that AES-CBC mode is preferred over ECB mode is because CBC provides integrity.

(c) True or False: Comparing a hashed password to a precomputed list of hashes of commonly used passwords is one example of a side-channel attack.

(d) True or False: In Bitcoin, if there are multiple versions of the block chain, the longest version will be accepted as the winner/valid block chain.

(e) True or False: A denial-of-service attack requires injecting spoofed packets.

(f) True or False: TLS provides both confidentiality and integrity for messages sent using it.

(g) True or False: If there are only a few possibilities for M, an eavesdropper who sees SHA256(M) can figure out what M was.

(h) True or False: If there are only a few possibilities for M, an eavesdropper who sees $E_{K_B}(M)$ can figure out what M was. (Assume $E_{K_B}(M)$ is the El Gamal encryption of M under Bob's public key and the eavesdropper doesn't know Bob's private key.)

(i) True or False: If there are only a few possibilities for M, an eavesdropper who sees the AES-CBC encryption of M can figure out what M was. (Assume the eavesdropper does not know the key used for AES-CBC encryption.)

(continued on next page)

(j) TRUE or FALSE: TLS with RSA key exchange provides perfect forward secrecy.

(k) TRUE or FALSE: TLS with Diffie-Hellman key exchange provides perfect forward secrecy.

(l) TRUE or FALSE: Given $x^y \bmod p$, x, and p (where p is a 2048-bit prime), there is no known, efficient way to compute y.

(m) TRUE or FALSE: A major reason that SSL/TLS is not used everywhere is because of the high cost of symmetric-key cryptography.

(n) TRUE or FALSE: In onion routing (e.g., Tor), if one of the mixes is dishonest, then there is no guarantee of anonymity for the user.

(o) TRUE or FALSE: A surefire way to avoid censorship systems is to just encrypt all of your communications.

(p) TRUE or FALSE: Fuzz testing looks for security bugs in your code by running it on random or semi-random inputs.

(q) TRUE or FALSE: Neo is the man. Also, down with Governor Stalloon.

Problem 2 ***Fill in the blank*** **(8 points)**

Fill in the blank.

(a) When you click on a link, the _____ header tells the server which URL you were at that took you to the link.

(b) An attack that reconstructs what you are typing on your keyboard by recording and then analyzing the specific sounds made as you type would be an example of a _____ attack.

(c) To evaluate an intrusion detection system, we usually need to know its false _____ rate, its false _____ rate, the base rate of attacks, and the relative cost of a _____ vs. a _____.

Problem 3 ***Very Short Answer*** **(12 points)**

Answer each of the following briefly. Do not explain or justify your answer.

(a) All else being equal, would a smart attacker prefer a denial-of-service attack with amplification factor 3 or amplification factor 17?

(b) Name two techniques to avoid or defend against replay attacks.

(c) AirBears is an open (unencrypted) Wifi network. Suppose you are using AirBears to connect to HTTP websites on the Internet. Can another Berkeley student who is nearby eavesdrop on the payload of all the packets you send over this Wifi link? (Answer yes or no.)

(d) Assume you are using AirBears to connect to HTTP websites on the Internet, as before. Can a random stranger off the street (who has no affiliation with the campus and doesn't have any Berkeley account) eavesdrop on the payload of all the packets you send over this Wifi link, assuming the stranger is nearby? (Answer yes or no.)

(e) You're still using AirBears. You visit your bank web site in your browser. Assume your bank uses HTTPS for everything (no HTTP). Can a nearby Berkeley student observe your bank account balance by eavesdropping on the Wifi connection? (Answer yes or no.)

Problem 4 ***Short Answer*** **(15 points)**

Answer each of the following. A sentence or two will suffice.

(a) Does TLS defend against SQL injection attacks? Why or why not?

(b) What is the difference between reflected XSS and stored XSS?

(c) What is leap-of-faith authentication?

Problem 5 ***TLS security*** **(9 points)**

Alice goes to `https://paypal.com/`, logs in by entering her Paypal username and password, adds her credit card number to her account, and makes a payment—all through Paypal's web site. Assume her browser is using the latest and greatest version of TLS and that Paypal uses HTTPS for everything (no HTTP).

Which of the following could an on-path eavesdropper deduce? Circle all that the eavesdropper could deduce.

A. The approximate size of the HTTPS requests from Alice's browser

B. The approximate size of the responses from the Paypal server

C. The approximate number of requests made by Alice's browser

D. Alice's Paypal username and password

E. Alice's credit card number

F. The fact that Alice is visiting Paypal (and not, say, `https://wellsfargo.com/`)

G. The session cookie for this connection

H. Any CSRF tokens that the Paypal server uses during this connection

I. The TCP initial sequence numbers used for this connection

Assume there are no security bugs or flaws in Alice's browser or Paypal's server, beyond what is implied by the statement of the problem. Assume the attacker can only passively eavesdrop; no active attacks, no man-in-the-middle.

Problem 6 ***Security principles*** **(8 points)**

Each of the following scenarios represents a failure to respect some security principle. Identify the most relevant security principle that was violated, and briefly justify your answer. A sentence or two will suffice.

(a) `www.luser.com` has a NIDS that extensively logs useful information about each incoming packet that it observes; however, this leads to log files which are cumbersome and difficult for the sysadmin to parse. This increases the time to respond to real alarms.

Principle:

Justification:

(b) When designing their cryptography system, `www.doh.com` assumed that no one could figure out their data encryption scheme since they guard their source code very carefully.

Principle:

Justification:

Problem 7 ***Memory safety*** **(9 points)**

In Project 1, you had to write memory-safe code. Your project partner got you started by writing the following snippet of code to print a non-nul-terminated UserComment field from a JPG file:

```
/* requires: data != NULL && sizeof(data) == len */
int print_usercomment(char *data, size_t len) {
    size_t n;
    unsigned char *val;

    if (/*TODO*/)
        return -1;
    if (memcmp(data, "ASCII\0\0\0", 8) != 0)
        return -1;

    n = len - 8 + 1;
    val = malloc(n);
    if (val == NULL)
        return -1;
    memcpy(val, data+8, n-1);
    val[n-1] = '\0';

    printf("UserComment: %s\n", val);
    return 0;
}
```

Unfortunately your partner disappeared without filling in the part marked `/*TODO*/` and left that to you.

The user comment that follows the header `ASCII\0\0\0` is allowed to be any number of characters, including possibly the empty string.

What should you replace the `/*TODO*/` with, to ensure this code will be memory-safe, assuming that the caller satisfies the documented precondition?

As a reminder, `memcpy()` and `memcmp()` are library functions declared as:

```
void memcpy(char *dst, char *src, size_t n);
void memcmp(char *p, char *q, size_t n);
```

`memcpy(dst, src, n)` copies `n` bytes starting at the address `src` to `dst`.
`memcmp(p, q, n)` compares `n` bytes starting at the address `p` to the `n` bytes starting at the address `q`, and returns 0 if they are all equal.

Problem 8 ***Software security*** **(12 points)**

The following C code has a vulnerability:

```
/* requires: len == size(in) */
void vuln(uint64_t in[], size_t len) {
    char a[20];
    size_t i;
    for (i=0; i<len && i<20; i++) {
        memcpy(&(a[i]), &(in[i]), sizeof(uint64_t));
    }
}
```

Assume that the elements and length of `in` are controlled by the attacker, but it is guaranteed that `vuln()` will be called with arguments where `len` is equal to the number of elements in `in` (i.e., assume that the precondition of `vuln()` always holds).

`vuln()` is vulnerable to a buffer overflow attack. Explain why, and given an example value of `len` that would enable such an attack.

Explanation:

Example value of `len`:

As a reminder, `memcpy()` is a library function declared as:

```
void memcpy(char *dst, char *src, size_t n);
```

`memcpy(dst, src, n)` copies `n` bytes starting at the address `src` to `dst`.
Also, `uint64_t` is a 64-bit unsigned integer.

Problem 9 ***Network security*** **(22 points)**

Consider the following capabilities that an attacker might or might not have:

A. Eavesdrop (on packets whose destination IP address is not the attacker's IP address)

B. Inject packets with a forged source address

C. Modify packets sent by someone else

D. Drop packets sent by someone else

E. Access to a precomputed list of hashes of many candidate passwords

For each attack below, write down which capabilities from the above list are needed to mount the attack in practice, or write "None" if none of the above are needed.

(a) Steal a session cookie used at `http://cnn.com/`

(b) Perform a DNS amplification denial-of-service attack against a web server

(c) Perform a SYN flooding denial-of-service attack against a web server

(d) Perform reflected XSS against some web site that has a reflected XSS vulnerability

(e) Hijack a modern TCP connection

(f) Perform DNS spoofing, against a modern DNS implementation

(g) Which of the following attacks can be carried out by an off-path attacker? Circle all that can be carried out by an off-path attacker.

a. Steal a session cookie used at `http://cnn.com/`

b. Perform a DNS amplification denial-of-service attack against a web server

c. Perform a SYN flooding denial-of-service attack against a web server

d. Perform reflected XSS against a web server with a reflected XSS vulnerability

e. Hijack a modern TCP connection

f. Perform DNS spoofing, against a modern DNS implementation

g. Perform ARP spoofing

h. Man-in-the-middle a TLS connection

Do not justify your answer. You are not allowed to assume that any endpoint has site-specific vulnerabilities in its code, beyond those that are implied by the above.

Problem 10 ***Web security*** **(24 points)**

FaceSpace has implemented an all-new friend finder feature! Now, users of FaceSpace can enter a search string such as "`rohin`" in order to find people to friend. FaceSpace also keeps track of the most popular search strings. Since the feature is new, the most popular search strings at the moment have only been searched for a few hundred times.

When the user visits a URL such as
`https://www.facespace.com/friendfinder.php?name=rohin`
(note the use of HTTPS), Facespace runs the following (in pseudocode):

```
name = getParameterFromUrl(url, 'name');
increment_number_of_searches(name);
command = "SELECT username FROM users WHERE name = '" + name + "'";
results = execSQLForTable(command, "users");
html = "<p>You searched for: " + name + ".</p><p>" + results + "</p>";
send_to_client(html);
```

The code above first extracts "`rohin`" from the URL (simply by scanning for "`name=`" and returning whatever is after that). It issues an SQL query to the `users` table, which contains the username, password (in cleartext), and name of all of the FaceSpace users. The developers have made sure to ensure that the SQL command can access only the "`users`" table. The code sends back the results to the user.

When a user asks for the most popular search strings, FaceSpace goes through the database containing the number of searches for each search string, and returns the top 100 search strings sorted by number of searches.

- Alice is a FaceSpace user. She does not care about the most searched names, so she will never view the top 100 search queries. However, since she knows Mallory, she will visit any site that Mallory sends to her.
- Bob is another FaceSpace user. He does not know Mallory and is cautious by nature, so he will not visit any sites recommended by Mallory. He likes to follow celebrities and so views the top 100 search queries page frequently.
- Charlie is another FaceSpace user. He will not visit any sites recommended by Mallory and does not care about the most searched names, so he will never view the top 100 search queries. Like the Governor of Project 3, he is very careful and if anything seems suspicious, he will close his browser and go do something else.
- Mallory is an off-path attacker who has an account on FaceSpace.

(continued on next page)

For each of the following goals, say whether or not Mallory can achieve that goal: yes or no. If your answer is "yes" (i.e., Mallory can achieve the goal), then list the name of the attack technique she could use. You don't need to describe the attack in detail; just the name. If your answer is "no", you don't need to provide any further justification or explanation.

Do not assume any software vulnerabilities or design flaws in anyone's software, other than what is implied by this question.

(a) Mallory wants to steal Alice's FaceSpace cookie.

Can she?
Attack name:

(b) Mallory wants to steal Bob's FaceSpace cookie.

Can she?
Attack name:

(c) Mallory wants to steal Charlie's FaceSpace cookie.

Can she?
Attack name:

(d) Mallory wants to steal Alice's FaceSpace password.

Can she?
Attack name:

(e) Mallory wants to steal Bob's FaceSpace password.

Can she?
Attack name:

(f) Mallory wants to steal Charlie's FaceSpace password.

Can she?
Attack name:

(continued on next page)

Finally, answer the following question:

(g) FaceSpace hires a security auditor, who recommends that they add an unpredictable CSRF token to the search form and search URL. Thus, the search URL now looks like

`https://www.facespace.com/friendfinder.php?token=1A0B743FC08DA37A&name=rohin`

The code for that page is modified to first check that the token is correct; if it is incorrect or missing, none of the rest of the code is executed and the page doesn't load. Nothing is changed about the code for the page with the top 100 search strings.

Which of goals (a)–(f) become impossible, once this change is made? Do not justify your answer; just list the set of goals that Mallory cannot achieve.

Problem 11 ***Cryptography*** **(16 points)**

Professor Goodhearted at the University of Birkland has created a course project for her students, but she's worried the project is too hard, so she wants to include a way for students to unlock a hint that will help them with the project. To see the hint, students will have to solve a separate, optional math problem (a "math puzzle").

The math puzzle has been constructed so its solution is a random 5-digit number N. Prof. Goodhearted has computed $K = \text{SHA256}(N)$ and $C = E_K(M)$, where M is the hint, N is the solution to the puzzle, and E is a secure symmetric-key encryption scheme. (In other words, C is the encryption of message M under the key $K = \text{SHA256}(N)$, using a secure symmetric-key encryption scheme.) Prof. Goodhearted includes in the project VM image a program that has the ciphertext C hardcoded in it; when a student runs the program, the program prints out the math puzzle, prompts the student for their answer to the puzzle, hashes their answer with SHA256, then decrypts C using the hash of the student's answer as the key and prints the decryption.

(a) Explain the problem with Prof. Goodhearted's scheme. How can a smart but dishonest student learn the hint, even if he can't figure out how to solve the math puzzle?

(b) Waiting at the bus stop the next day, Prof. Goodhearted realizes the flaw in her scheme. To repair the flaw, she has the idea of letting $K = H_{\text{slow}}(N)$, where H_{slow} is a slow hash function chosen so that computing H_{slow} on a single input will take 100 milliseconds. Is this adequate to fix the flaw? In other words, is this enough that students will have to solve the math puzzle to learn the hint? Write "yes" or "no", then explain why or why not.

(continued on next page)

(c) Prof. Goodhearted has so much fun with this that she ends up creating three really tough math puzzles; their answers are random 5-digit answers N_1, N_2, N_3, respectively. As before, the hint is M. She wants to arrange that a student can see the hint M if the student solves all three math puzzles correctly (and it should be as hard as possible for dishonest students to cheat; for her purposes, it suffices if a dishonest student would need to solve at least two of the puzzles to learn the hint). She plans to do this by computing a key K somehow, computing $C = E_K(M)$, and writing a program with C encoded in it that she'll give to students. How should she compute K? (You don't need to describe how the program should work.)

$K =$ _____

(d) The next day, Prof. Goodhearted has second thoughts: asking students to solve all three puzzles might be a bit too much. Instead, she wants students to be able to unlock the hint M if they solve any two out of the three math puzzles (and it should be as hard as possible for dishonest students to learn the hint if they've solved fewer of the puzzles; for her purposes, it suffices if a dishonest student would need to solve at least one of the puzzles to learn the hint). She spends all day trying to figure out a way to do this, without any luck, so she gives up. That night, she wakes up in the middle of the night with a brilliant idea for how to do this: she will compute three ciphertexts $C_1 = E_{K_1}(M), C_2 = E_{K_2}(M), C_3 = E_{K_3}(M)$, hardcode C_1, C_2, C_3 into her program, and then the program will ask the student to input their two solutions and the program will do something clever. Pleased with herself, she goes back to sleep.

Unfortunately, when she wakes up in the morning she can't quite remember how her program was going to work or how she was planning to choose the keys K_1, K_2, K_3. Help her out. How should she define K_1, K_2, K_3? (You are not allowed to use modular arithmetic or public-key cryptography; Prof. Goodhearted is sure she didn't need them. Don't describe how her program should work; she can figure that out on her own.)

$K_1 =$ _____

$K_2 =$ _____

$K_3 =$ _____

Problem 12 ***TCP*** **(15 points)**

A bunch of Stanford students have decided to build their own operating system, TreeOS. They are considering several possible schemes for how to choose TCP initial sequence numbers in TreeOS. They've hired you to figure out which of them are secure. A secure scheme should prevent TCP blind spoofing by off-path attackers.

Notation: $H(\cdot)$ denotes the SHA256 hash function; t is the number of milliseconds since January 1, 1970 (remember that there are one thousand milliseconds in a second); s is the number of seconds since January 1, 1970; k is a random 128-bit value chosen at boot time using a cryptographically-secure pseudorandom number generator (k remains unchanged until the machine is rebooted). Each time a TCP initial sequence number is needed, we use the scheme to generate a new one. Assume that the clock is accurate and never "goes backwards."

For each scheme below, say whether it is secure or not. If it is insecure, describe the attack; if it is secure, don't provide any justification.

(a) The initial sequence number is $t \bmod 2^{32}$.

Is it secure?
If insecure, the attack:

(b) The initial sequence number is $t + k \bmod 2^{32}$.

Is it secure?
If insecure, the attack:

(c) The initial sequence number is $H(t) + k \bmod 2^{32}$.

Is it secure?
If insecure, the attack:

(continued on next page)

(d) The initial sequence number is $H(k \mid\mid s \mid\mid p_l \mid\mid p_r) \bmod 2^{32}$, where p_l is the local TCP port number (on the local machine) and p_r is the remote TCP port number (on the other machine). For instance, if this machine is initiating a TCP connection, p_l is the TCP source port number and p_r is the TCP destination port number.

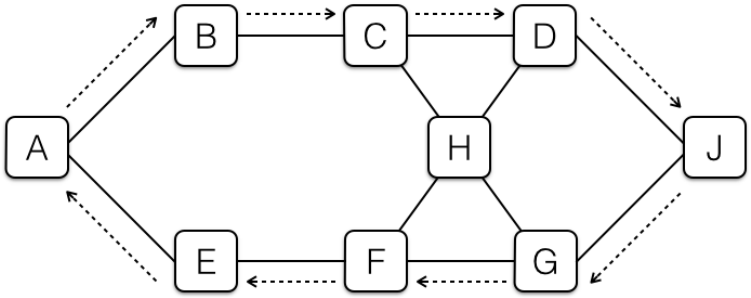
Is it secure?
If insecure, the attack:

(e) The initial sequence number is $H(k \mid\mid g) \bmod 2^{32}$, where g is a global counter that is initialized to zero at boot time and is incremented once each time a new initial sequence number is needed.

Is it secure?
If insecure, the attack:

Problem 13 ***TCP*** **(18 points)**

Consider the following network topology:



The machine A has initiated a TCP connection to machine J. As it turns out, all packets from A to J happen to follow the path indicated by the right-facing dotted arrows, and all packets from J to A happen to follow the path indicated by the left-facing dotted arrows. Machines A and J use modern TCP software and do not have any special defenses against attack.

(a) Suppose that Mallory controls (only) machine C. Can she inject RST packets destined for machine J into this TCP connection, such that they will be accepted by machine J? Why or why not?

(b) Suppose that Mallory controls (only) machine C. Can she inject spoofed data into this TCP connection, so machine J will accept the spoofed data thinking that it came from machine A? Why or why not?

(c) Suppose that Mallory controls (only) machine H. Can she inject spoofed data into this TCP connection, so machine J will accept the spoofed data thinking that it came from machine A? Why or why not?

(d) Suppose that Mallory can eavesdrop on all packets that go through machine C (but cannot inject forged packets from C). Also Mallory can run software on machine F that lets her inject forged packets from F (but cannot eavesdrop on packets going through F). Can Mallory injected spoofed data into the TCP connection, so that it will be accepted by machine J as though it came from A? Why or why not?

(e) Suppose that Mallory can eavesdrop on all packets that go through machine F (but cannot inject forged packets from F). Also Mallory can run software on machine F that lets her inject forged packets from C (but cannot eavesdrop on packets going through C). Can Mallory injected spoofed data into the TCP connection, so that it will be accepted by machine J as though it came from A? Why or why not?

Problem 14 ***Cryptography*** **(15 points)**

The IRS wants to help employers perform background checks on their employees. They compile a list of names and 9-digit social security numbers (SSNs) of all US citizens and residents who are allowed to work in the US. They want to distribute an scrambled copy of this list to employers, so that employers can check candidate hires without needing to contact the IRS.

The IRS chooses a random 2048-bit prime p. They also choose a random number r uniformly at random from the range $2 \ldots p-2$. To scrambled a SSN s, they compute $y = r^{H(s)} \bmod p$ and use y as the scrambled version of s. Here $H(\cdot)$ is the SHA256 hash function. So, the IRS compiles a list $(f_1, s_1), (f_2, s_2), \ldots, (f_n, s_n)$ of allowed full names and SSNs, where f_i is the full name of the ith allowable person and s_i is their SSN. Then, the IRS scrambles each of the SSNs one-by-one to get the scrambled list $(f_1, y_1), (f_2, y_2), \ldots, (f_n, y_n)$ where $y_i = r^{H(s_i)} \bmod p$. The IRS publishes on their website the scrambled list $(f_1, y_1), (f_2, y_2), \ldots, (f_n, y_n)$, the prime p, and the number r.

The idea is that before an employer hires a candidate whose name is f and whose SSN is t, the employer will check whether there is an entry on the scrambled list that matches (f, t). If it matches, they will know they are allowed to hire the candidate; if it doesn't match, they will contact the IRS to inquire further.

(a) Suppose that an employer is considering hiring a candidate whose full name is f and whose SSN is t. How can the employer check whether this person is on the IRS's list of people who are allowed to work in the US, i.e., whether it matches any entry on the scrambled list?

(b) Explain why this is not a secure way of scrambling the SSNs.

(c) The IRS proposes to improve their scheme by choosing a different value of r for each SSN they scramble. In their improved scheme, the scrambled list is $(f_1, r_1, y_1), (f_2, r_2, y_2), \ldots, (f_n, r_n, y_n)$ where each r_i is random (chosen independently of everything else) and $y_i = r_i^{H(s_i)} \bmod p$. The IRS will publish the scrambled list $(f_1, r_1, y_1), \ldots, (f_n, r_n, y_n)$ and the prime p on their website.

Is this improved scrambling scheme secure? Why or why not?