Final Exam

This exam is closed book and closed notes. However, you may consult a single two-sided reference sheet. You may not use any electronic devices or communicate with anyone other than course staff.

Print your answers legibly. The intended answers fit within the spaces provided on the question sheets. You may use the back of the preceding page for scratch work. If you run out of room for an answer, continue on the back of the page and clearly mark your answer.

Do not open this exam booklet until instructed to do so.

Security is hard, and so is this exam. Do your best, and *don't panic!*

Time limit: **90 minutes**.

Write and sign the honor code pledge:

> "*I have neither given nor received unauthorized aid on this examination, nor have I concealed any violations of the Honor Code.*"

_____

_____

_____

_____
(Signature)

_____
(Print your name)

_____
(Uniqname)

Question:	1	2	3	4	5	Bonus	Total
Points:	10	10	10	10	10	0	50
Score:							

1. Short Answer

(a) [2 points] What is Kerckhoffs's principle? What is the justification for it?

(b) [2 points] Tor encrypts traffic all the way from the sender to the exit node, but not from the exit node to the final destination. Why not?

(c) [2 points] What is forward secrecy? Give an example scenario where this property might be desirable.

(d) [2 points] What is the same-origin policy? What is its role in browser security?

(e) [2 points] In a secure channel, why should the sender encrypt the plaintext first and then apply integrity protection (rather than the reverse order)?

2. Attacks and Defenses

Briefly describe each of the following attacks and give an example of the damage it can cause. Then give a defense that can potentially mitigate it.

(a) [2 points] Shellshock exploits

(b) [2 points] Van Eck phreaking

(c) [2 points] Length extension

(d) [2 points] Spear phishing

(e) [2 points] Null-prefix attacks against HTTPS

3. Applied Cryptography

You've been hired to perform a security audit for Shushmail, a new "secure" email service.

Each Shushmail user has an RSA public key with public exponent 3 and a unique 4096-bit modulus N. Another user may encrypt a message m to this key by choosing a random 256-bit key k, using AES in counter mode to encrypt m using key k: $c_m = \mathrm{AES_k(m)}$ and then using RSA to encrypt k to the recipient's public key: $c_k = k^3 \bmod N$. The ciphertext is then (c_k, c_m). Assume that the keys are properly generated, that users have a way of looking up correct public keys for recipients, and that the private RSA keys are stored securely.

(a) [3 points] Under this protocol, an eavesdropper who intercepts (c_k, c_m) can easily learn m. Explain the vulnerability, and state precisely how to change the protocol to fix it. (*Hint:* Try using pencil and paper to encrypt with, say, $k = 2^8$ and $N = 2^{128} - 2$. What happens?)

Suppose we fix Shushmail to properly protect confidentiality. A bank uses Shushmail to accept instructions from its customers. The bank, which knows its customers' public keys, first encrypts a message to the customer containing a secret value that is treated as a single-use authorization code. The customer can send money to another account by encrypting a message to the bank of the form: "`Transfer $`*`vvvvv`*` to account `*`xxxxxx`*`; auth=`*`authorization_code.`*"

(b) [3 points] Consider a man-in-the-middle attacker who can guess the destination account number. How can they subvert the protocol to steal money? How should we change Shushmail to fix this? (*Hint:* Recall that counter mode works like a stream cipher. You XOR each plaintext byte with a keystream generated by encrypting successive integer values.)

Shushmail has decided to share "anonymized" server logs with a group of researchers. Shushmail wants to ensure that users' IP addresses aren't revealed, but the researchers need to be able to associate different requests that come from the same IP address. The logs are huge, and anonymization has to be applied efficiently with only a small, fixed amount of storage.

(c) [2 points] Shushmail plans to replace each IP address with the SHA-256 hash of the IP address. Why is this insufficient to provide strong protection for the secrecy of the IPs?

(d) [2 points] Propose a stronger scheme based on HMAC, and briefly argue why it is better.

4. Binary Exploitation

You are developing a proof-of-concept exploit for a traditional stack-based buffer overflow. After trapping on a breakpoint inside the vulnerable function, `gdb` yields the following:

```
==> x/28wx $esp
0xbffee770:      0xbffee786  0xbffff4b3  0xbffee798  0x08048ef2
0xbffee780:      0x080c6008  0x00000000  0x00000000  0x00000000
0xbffee790:      0x00000000  0x00000000  0x00000000  0x00000000
0xbffee7a0:      0x00000000  0x00000000  0xbffee7b8  0x08048f5e
0xbffee7b0:      0xbffff4b3  0x00000000  0x00000000  0x00000000
0xbffee7c0:      0x00000000  0x00000000  0xbffff278  0x0804901f
0xbffee7d0:      0x00000002  0xbffff314  0xbffff320  0x00000000

==> x/2wx $ebp
0xbffee7a8:      0xbffee7b8  0x08048f5e
```

Make these assumptions:

- An unbounded buffer begins at `0xbffee784`. You can cause arbitrary data to be copied into the buffer by a call to `memcpy`, and the vulnerable function then immediately returns.
- The machine is a 32-bit little-endian system that behaves like the VM from Project 4.
- There are no defenses such as ASLR, stack canaries, or a non-executable stack (DEP).

(a) [7 points] You want to run a payload that is 24 bytes long and works like the shellcode provided in Project 4. Write the bytes that should be copied to the buffer for the most concise possible exploit. Fill in one byte (in hex) per blank; use as many or few as you need. Draw a box around the positions that will contain the payload and write shellcode.

__ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __
__ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __
__ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __
__ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __

(b) [1 point] Suppose instead that the machine *does* use a non-executable stack (DEP). What will go wrong when you attempt the attack from part (a)? Be as precise as you can.

(c) [2 points] List two techniques you could use to exploit this vulnerability in spite of DEP.

5. Privacy and Surveillance

Consider a hypothetical classified order of the Foreign Intelligence Surveillance Court, ordering Verizon to supply to the National Security Agency, each day for a three-month period,

> an electronic copy of the following tangible things: all call detail records or "telephony metadata" created by Verizon for communications (i) between the United States and abroad; or (ii) wholly within the United States, including local telephone calls. [...] Telephony metadata includes comprehensive communications routing information, including but not limited to session identifying information (e.g., originating and terminating telephone number, International Mobile Subscriber Identity (IMSI) number, International Mobile station Equipment Identity (IMEI) number, etc.), trunk identifier, telephone calling card numbers, and time and duration of call.

(a) [4 points] NSA might point out that the order does not require Verizon to supply "the name, address, or financial information of [any] subscriber or customer." Is this sufficient to protect innocent Verizon customers from being personally identified? Explain in detail.

(b) [6 points] NSA might point out that the order does not require Verizon to supply the content of any phone calls. Does the order still raise privacy concerns, beyond those emphasized in part (a)? If so, explain and give a concrete scenario as an example.

6. Extra Credit

(a) [5 points (bonus)] Which of the following security practices have you personally adopted?

Honor code . . . be honest! (Check all that apply.)

- ◯ I've chosen hard-to-crack passwords for all my important accounts.
- ◯ I've enabled two-factor authentication for my most important accounts that support it.
- ◯ I've turned on disk encryption on my personal computer.
- ◯ I've turned on encrypted storage on my smartphone, or I don't have one.
- ◯ I've installed at least one of the following tools: Tor, GnuPG, or OTR.

Describe one additional good computer security practice you've personally adopted that you will recommend to your friends.

(b) [5 points (bonus)] You receive an email, purportedly from a friend, with a suspicious attachment. The attachment is an encrypted and signed file created using Tar and GnuPG. You decide to extract it by running this command:

```
$ gpg --decrypt file.tar.gz.gpg | tar xz
```

You already have your friend's public key set up in GnuPG. The gpg manpage says:

> `--decrypt [file]` Decrypt `file` (or `stdin` if no file is specified) and write it to `stdout`. If the decrypted file is signed, the signature is also verified.

Assume the crypto itself is secure, that `tar` and `gpg` are the most recent versions, and that neither has any 0-day vulns. Still, you are making a *particularly dangerous* mistake. Why?

(c) [0 points (bonus)] That's it. The semester's over. How are you feeling?