Name: _____

ECE568S: Final Exam

Examiner: C. Gibson

Duration: 2.5 hours
Exam Type: C (single reference sheet, both sides)
Calculator Type: 4 (none)

1. Please use a **pen** to complete all of your answers.
2. Do not turn this page over until you have received the signal to start.
3. This exam is closed book. One double-sided, letter-sized (8.5"x11") reference sheet may be used. Use of calculators, computing and/or communicating devices is not permitted. Work independently.
4. Do not remove any sheets from this test book. Answer all questions in the space provided, unless otherwise indicated. No additional sheets are permitted.
5. The value of each question or part-question is noted next to the question. The total of all marks is 100.
6. Please write your name and student number in the space below, and write your name on the top of each sheet.

First Name: _____

Last Name: _____

Student Number: _____

Q1. _____ / 20	Q4. _____ / 20
Q2. _____ / 25	Q5. _____ / 15
Q3. _____ / 10	Q6. _____ / 10
	Total: _____ / 100

Name: _____

1. Short-Answer Questions (20 marks)

Please *briefly* answer the following questions, in the space provided. If you require more space, please indicate that you are continuing your answer on the back of the sheet, and put the question number next to the continuation of your answer.

a) Name *two* examples of standard Unix commands that require root privilege to run and, for each command, briefly explain why it requires this. **(4 marks)**

b) Name *three* techniques that can be used to make buffer-overflow attacks more difficult and, in a sentence, briefly explain each one. **(6 marks)**

c) The Bell-LaPadula security model defines an information flow that is sometimes referred to as "read down, write up". What security property Bell-LaPadula is trying to protect, and how might that property could be compromised if the information flow were reversed? **(2 marks)**

d) Briefly explain what is meant by *mandatory access control*. **(2 marks)**

e) Describe *two* different mechanisms that a computer virus might use to hide its presence from an anti-virus scanner. **(4 marks)**

f) 3DES uses a 168-bit key that is split into three pieces and used to run the DES algorithm three times. The chaining between the three blocks is structured as [encode] → [decode] → [decode]. Why was this design chosen, rather than using one new [encode] block that took the entire 168-bit key? **(2 marks)**

Name: _____

2. Software Vulnerabilities (25 marks)

a) You are asked to write a program that asks the user for a filename and then prints the file. The program should only allow the user to access files in the current directory. What character(s) would you filter out from the user's input, and why? **(2 marks)**

b) You are asked to examine a program that adds new users into a database. Some special accounts are "administrators", but most user accounts are not. User accounts are normally created and, as a final step, the following SQL command is executed:

```
update USER_LIST set ADMINISTRATOR='N' where USERNAME='jskule';
```

Assuming that the string **jskule** in the example above came directly from user input, what *alternate string* could an attacker provide for the username, in order to create a user named 'jskule' that is an Administrator? **(3 marks)**

c) Consider the following two functions. Each of them expose the program to the possibility of a stack overflow attack. For each function, describe the input required to exploit it, and exactly what the exploit would allow the attacker to do (*e.g.*, "*a string containing at least three vowels would overflow into the ____ of ____, allowing the attacker to ____*."). **(10 marks)**

```
#define MAX_BUFFER 80

void A ( const char * userInput )
{
      char    buffer[MAX_BUFFER];
      short   len = strlen(userInput);
      short   i;

      for ( i = 0 ; i <= len ; i++ ) buffer[i] = userInput[i];

      // ....
}

void B ( const char * userInput )
{
      char    buffer[MAX_BUFFER];
      short   len = strlen(userInput);
      short   i;

      // Abort if userInput is too long
      if ( len >= MAX_BUFFER ) return;

      for ( i = 0 ; i <= len ; i++ ) buffer[i] = userInput[i];

      // ....
}
```

d) **(10 marks)** Consider the following program, and assume that it is running in the ECF lab:

```
01 #include <stdio.h>
02 #include <string.h>
03
04 void foo (char * b)
05 {
06      char buffer[10];
07      strncpy(buffer, b, 10);
08      printf("%s", buffer);
09 }
10
11 int main()
12 {
13      char buffer[80];
14      gets(buffer);
15      foo(buffer);
16      gets(buffer);
17      foo(buffer);
18 }
```

The program uses stack canaries to prevent against buffer overflows; the "canary" value is placed immediately above the local variables on each stack frame. The canary value is picked randomly when the program first starts, and the same value is then used for the remainder of that processes' execution.

Describe an approach that might be able to bypass the stack canaries and successfully complete a buffer overflow attack. You do not need to provide exact attack strings, but please do be specific about what input you would provide, and what it is meant to do (*e.g.*, "*I would start by providing two bytes of input in order to _____ main's stack frame...*"). It is okay if your solution needs to be run a few times, but it needs to run in a practical amount of time; assuming the stack canary is "6" and running the attack several billion times is *not* acceptable.

Name: _____

3. Covert and Side Channels (10 marks)

a) **(5 marks)** The following function is used to check a user's input against a valid password:

```
01 int
02 checkPass ( char * enteredPassword, char * validPassword )
03 {
04      int i = 0;
05      int len = strlen(validPassword);
06
07      // Compare each character; return "0" if they don't match
08      for (; i <= len ; i++)
09           if (enteredPassword[i] != validPassword[i]) return(0);
10
11      // The user entered the correct password!
12      return(1);
13 }
```

Describe a side-channel attack that someone could use to fairly quickly crack the password. (Assume that the function runs properly as written, and both strings are a reasonable length.)

b) **(5 marks)** Your company recently got a very good deal on a used printer for your office network. A couple of weeks later you discover that some of your confidential documents have been leaked to an outside party. You review the network logs, and find nothing unusual, except for a large number of DNS requests from the printer to `a.attacker.com` and `b.attacker.com`. The requests do not contain anything other than a normal DNS request, but you note that `attacker.com` is handing back DNS replies with an odd TTL of 0 (meaning "do not cache").

What covert channel is likely being exploited here, and how is the data being communicated out to the outside party? Why is the TTL of 0 significant?

Name: _____

4. Encryption and Hashing (20 marks)

a) A well-designed authentication system will verify passwords against a *hashed* copy of the user's actual password, stored in a protected file or database. What is the purpose of storing password hashes, rather than storing the users' plain (unhashed) passwords? **(2 marks)**

b) Current Linux systems pick a random "salt" value for each user, which further randomizes the hash value stored in the system password file. Describe *two* benefits that this provides, in the event that an attacker is able to obtain the list of hashed user passwords. **(8 marks)**

c) Explain the purpose chiper-block chaining (CBC), and explain how the "chaining" works in its encryption operation (feel free to make use of a diagram). **(5 marks)**

d) The Bolted Fast Company produces two safes with electronic locks. The safes both have keypads with ten digits (numbered 0 through 9), and will open only after the correct code has been entered. The BFC1000, a low-cost civilian model, uses a 6-digit code. After all six digits have been entered, it will either open or will signal that the code was wrong and ask for another try. The BFC2000, a far more expensive government version, expects a 40-digit code. Users of a beta-test version of the BFC2000 complained about the difficulty of entering such a long code correctly. As a result, the manufacturer made a last-minute modification: after every four digits, the BFC2000 now either confirms that the code has been entered correctly so far, or it asks for the previous four digits again. Compare the keyspaces of the BFC1000 and BFC2000 (both before and after the modification). If all other elements of the construction are otherwise identical, which safe is the most secure? **(5 marks)**

Name: _____

5. Network Communication (15 marks)

a) You discover that one of your companies' Internet-based products is subject to a replay attack. You are allowed to add any fields you would like to the message protocol; suggest *two* possible changes that would fix the problem and, for each solution, explain how the receiver would check to see if this message was a replay of a previous message. **(6 marks)**

b) Briefly provide an example of how the Right-To-Left Override (RLO) UTF-8 character can create problems presented with web links in phishing emails (*e.g.*, `http://www.paypal.com`). **(4 marks)**

c) In class we discussed a "SYN-ACK cookies" as a protection that servers can use to protect against SYN flood attacks. One key benefit of this approach is that the server does not need to allocate resources when the initial SYN packet is received from the client; it only allocates resources when the final ACK is received and authenticated. However, there is a potential vulnerability that this technique creates.

Assume you have a server that has a public IP address, but is protected behind a firewall, and the firewall is attempting to block all incoming connections. Many simple firewalls prevent incoming connections by dropping SYN packets. Describe an attack that could potentially be used to bypass the firewall and connect to this server if it is using SYN-ACK cookies. **(5 marks)**

6. Authentication and Authorization (10 marks)

Three parties ("A", "B" and "C") need to communicate with one another. Their communications do not need to be encrypted, but both parties in the conversation need to know who they are talking to. (*i.e.*, if A and B are communicating with one another, they each need to know this for certain at the start of their conversation.)

A, B and C are using a protocol based on public key cryptography in order to authenticate one another. Every time one process starts communicating with the other, they exchange a series of three messages encrypted with the *public key* of the other party:

A → B: $\{A, n_A\}K_B$	**Meaning:** *I am A, and I'm sending a nonce that only you can read.*
B → A: $\{n_A, n_B\}K_A$	**Meaning:** *Here is your nonce, proving that I decoded it, and my own nonce value that only you can read.*
A → B: $\{n_B\}K_B$	**Meaning:** *Here is your nonce, proving that I decoded it.*

Assume that this is an ideal situation: nobody can compromise any public keys, private keys or any part of the network.

a) There is a fundamental problem with the above protocol. If A starts a conversation with B, then B can simultaneously start a conversation with C and pretend to be A. Show a sequence of messages that demonstrates this flaw. **(6 marks)**

b) How might you fix this problem by changing contents of *one* of the three messages exchanged during the authentication? **(4 marks)**