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 noted below.
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. _____ / 33	Q4. _____ / 14
Q2. _____ / 25	Q5. _____ / 18
Q3. _____ / 10	Q6. _____ / 25
	Total: _____ / 125

Name: _____

1. Short-Answer Questions (33 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) Our study of computer security encompassed three main goals; two of them are *confidentiality* and *integrity*. What is the third goal? Provide an example of how it might be breached by an attacker. **(3 marks)**

b) What is the difference between a *vulnerability* and a *compromise*? **(3 marks)**

c) What are two fundamental differences between a *virus* and a *worm*? **(4 marks)**

d) What is a *reordering attack*? How might it be prevented? **(3 marks)**

e) What is a *deletion attack*? How does the SSL protocol protect against it? **(4 marks)**

f) Current Linux systems pick a random *salt* value for each user, which further randomizes the hash value stored in the system password file. Describe <u>two</u> benefits that this provides, in the event that an attacker is able to obtain the list of hashed user passwords. **(4 marks)**

g) Briefly explain the difference between an *Access Control List* and a *Capability*. **(4 marks)**

h) Most *rootkits* infect the operating system, and run in kernel mode. What technique do they often use to hook into the OS and actively avoid detection? **(5 marks)**

i) What is the goal of the *Biba* security model? In this model, which objects can a subject write to? **(3 marks)**

2. Software Vulnerabilities (25 marks)

a) You are writing an exploit for a *double-free* vulnerability on a 32-bit machine. The Return Instruction Pointer for the current function is saved on the stack at address 0x44444444, and your exploit code is located at an address of 0x22222222. Assume that each tag contains the following three elements (and that they appear in this order in memory):

```
chunkTag *  prev;
size_t      chunk_size;
chunkTag *  next;
```

and the *free* function contains the following code:

```
tag->next->prev = tag->prev;
```

What values would you place in a fake *chunk tag* to exploit this vulnerability? (Be specific and include the addresses you would use.) State any assumptions. **(6 marks)**

b) *Continuing from your answer above…* Assume the "free" function also contains the following code:

```
tag->prev->next = tag->next;
```

How would you construct your exploit to avoid this line from breaking your exploit code? **(5 marks)**

c) You discover that one of your web services contains a vulnerable line of code:

```
printf(buffer);
```

Assume that an attacker can connect to your application over the Internet and control the contents of *buffer*. She can connect multiple times, changing *buffer* each time, but she has no means of locally running commands, has no copy of your code, and has no prior knowledge of any of the addresses used in your program. What step(s) might the attacker take to successfully exploit this vulnerability in an efficient manner (*i.e.*, without just randomly guessing addresses)? **(6 marks)**

d) You are asked to consider a system that has the Non-Executable flag set on the memory pages occupied by the stack. Your task is to implement an exploit similar to Lab #1, but, in place of the usual shellcode, you are to use the Return-Oriented Programming (ROP) approach we discussed in lecture.

Your buffer overflow has succeeded in placing the following values onto the stack:

Absolute Address	Relative Address	Variable	Value
0x1039304	%EBP+0x54	y	0x0
0x1039300	%EBP+0x50	x	0x1039298
0x1039298	%EBP+0x48	buffer	/bin/sh\0

If we were able to place executable code on the stack, we might wish to execute the following:

```
movl 0x48(%ebp),%ebx      # argv[0]
leal 0x50(%ebp),%ecx      # argv
movl $0x0,%edx            # NULL (env)
movl $0xb,%eax
int  $0x80
```

The existing program code and libraries contain the following four code fragments (*widgets*):

```
...
0x10200  movl $0xf3,eax
0x10202  int $0x80
0x10205  movl %ebp,%eex
0x10207  ret
...
0x11700  call 0x12384
0x11705  subl $0x1,%eax
0x11707  movl 0x48(%eex),%ebx
0x1170a  ret
```

```
...
0x12100  leal 0x50(%eex),%ecx
0x12103  movl 0x60(%ebp),%ebx
0x12106  movl $0x1,%edx
0x12108  ret
...
0x13400  subl $0x1,%edx
0x13402  movl $0xc,%eax
0x13406  ret
...
```

Fill in the list below with the sequence of return address(es) that you should place onto the stack, in order for your exploit to properly run. (You just need to list the address(es): you do not need to show the stack or how to put the values onto it.) **(8 marks)**

RA1:		RA4:	
RA2:		RA5:	
RA3:		RA6:	

Name: _____

3. Security Models and Covert Channels (10 marks)

a) A rogue employee at your company has modified the code to your web server, and installed a covert channel that allows him to secretly send out the server's private key. As you investigate, you notice an odd behaviour with the file that contains your company's logo:

http://www.yourcompany.com/images/logo.jpg

The image is sometimes served up in 10ms and sometimes in 20ms, and you suspect this is being done intentionally. Briefly explain how someone could be using this to secretly leak information. **(5 marks)**

b) Your bank has recently released a new mobile banking app for your smartphone; it uses a Bluetooth signal that automatically authenticates you to ATMs when you approach, allowing you to conveniently withdraw money with a single button push, and eliminating the need to carry your bank card. From a security standpoint, is this an improvement over the approach we use today for authentication with bank machines? If it is not an improvement, explain how you might fix it. **(5 marks)**

Name: _____

4. Encryption and Hashing (14 marks)

a) You are asked to create a digital signature algorithm, using public key cryptography, to allow people to sign documents they create. Your algorithm starts by computing a SHA1 hash of the file. Explain how public/private keys would then be used by the Author of the document (*i.e.*, to create the signature), and the Recipient of the document (i.e., to verify the signature). **(4 marks)**

b) *Continuing from the previous question…* If the Recipient receives a document with a matching valid signature, can she rely 100% on the document being genuine? (Assume the public/private keys have not been disclosed or tampered with.) Explain why or why not. **(4 marks)**

c) Banks use an Electronic Clearing House (ECH) network to process cheques. Suppose that Bank A may send a file to Bank B, with each record in the file representing one cheque deposited in Bank A that is to be withdrawn from Bank B. Assume that each record contains only two account numbers and the amount of money to be transferred. The ECH file format specifies an integrity hash for each record, but not for the entire file. In what two ways could this be exploited by an insider at Bank A who does not have the ability to write valid hashes, but can modify a file? **(6 marks)**

5. Network Security (18 marks)

a) Explain how the ARP protocol can by exploited to enable a "*Man-in-the-Middle*" attack. **(5 marks)**

b) Explain why IPSec has an advantage over SSL in securing off-the-shelf software. **(3 marks)**

c) You are asked to review a design for a new firewall that defends against network Denial-of-Service attacks. The firewall monitors each incoming packet and maintains a count of how much traffic can be attributed to each source IP address. If any individual source IP contributes more than 1% of the total incoming traffic over the past 5 minutes, then the firewall drops all traffic from that IP address for the next hour. Explain why this is not a good solution to the Denial-of-Service problem (*i.e.*, how might an attacker avoid the firewall?). **(5 marks)**

d) *Continuing from the previous question...* How could an attacker abuse this design to create a problem for a legitimate user? **(5 marks)**

Name: _____

6. Web Security (25 marks)

a) While auditing a web application, you discover that it does no filtering on user input or output. You also find in the code that it constructs a SQL query as follows:

```
sql = "SELECT name FROM usertable WHERE email='" + buffer + "';"
```

The contents of *buffer* is taken directly from user input, and the contents of *name* is displayed to the user on a subsequent webpage. Show how an attacker could exploit this to perform a XSS attack.

For reference, the SQL syntax is for updating a database record is:

UPDATE *tablename* SET *fieldname*='*value*' [WHERE *fieldname*='*key*']

...but pseudocode is fine. **(10 marks)**

b) Explain the purpose of SIDF (Sender ID Framework) for email, and how it is implemented. **(5 marks)**

c) If a spammer knows that a company is using Spamhaus for its email *whitelist* (as discussed in class), explain how they might use DNS hijacking to ensure their spam is still delivered to that company's employees. **(10 marks)**