ECE 422 / CS 461, Midterm Exam

Wednesday, March 9th, 2016

Name:_____

NetID:_____

- Be sure that your exam booklet has 17 pages.
- Absolutely no interaction between students is allowed.
- Show all of your work.
- Write all answers in the space provided.
- Closed book, closed notes.
- No electronic devices allowed.
- You have **TWO HOURS** to complete this exam.

Page	Points	Score
2	16	
3	18	
4	5	
5	13	
7	5	
8	9	
9	4	
10	9	
11	3	
12	11	
14	7	
15	9	
16	14	
17	7	
Total:	130	

Question 1: Multiple Choice . . . . . *39 points*

For each question, circle all that apply.

(a) (2 points) Discretionary access control, which takes a hierarchical approach to controlling access, is more secure and strict than role-based access control.

A. True
B. False

(b) (2 points) As a defender, thinking about the weaknesses of a system falls under which category?

A. Security Policy
B. Risk Assessment
C. Threat Model
D. Countermeasures

(c) (2 points) The Unix set-user-ID ("suid") bit is used to ensure that folder contents are deleted only by the user who created them and the root user.

A. True
B. False

(d) (2 points) How are arguments passed to system calls in x86?

A. On the Stack
B. Through registers
C. The stack and registers
D. System calls don't take arguments

(e) (2 points) The x86 architecture uses Big Endian byte order for storage in memory.

A. True
B. False

(f) (2 points) To use RSA for confidentiality, you must encrypt with the private key and decrypt with the public key.

A. True
B. False

(g) (2 points) Which type(s) of malware require human assistance in order to replicate?

A. Worm
B. Virus
C. Trojan Horse
D. Bot

(h) (2 points) What type of virus code will include a code rewriter to generate semantically different virus code upon propagation?

A. Polymorphic Code
B. Heteromorphic Code
C. Metamorphic Code
D. Homomorphic Code

(i) (2 points) This type of Access Control Design relies on a system administrator to define permissions on files regardless of ownership.

A. Role-Based Access Control
B. Discretionary Access Control
C. Administrator-Based Access Control
D. Mandatory Access Control

(j) (2 points) If we want to provide a unique identifier for a message, we use

A. Cryptographic Hashing
B. Symmetric Key Cryptography
C. Public Key Cryptography
D. None of the above

(k) (2 points) What is a method of preventing SQL injection?

A. Public key Cryptography
B. Prepared Statements
C. Stack Canary
D. Salting

(l) (2 points) Which of the following ciphers can be used in Cipher Block Chaining (CBC) mode?

A. Caesar Cipher
B. Diffie-Hellman
C. AES
D. None of the above

(m) (2 points) In virtual machines, data can move between two guest OSes.

A. True
B. False

(n) (2 points) Suppose you want to crack a password. You know it's a 48-bit binary number. You know it's encrypted as H(password) where H is a perfect hashing function outputs 32 bits. How many trials do you need to crack the password in the worst case?

A. 2^{16}
B. 2^{32}
C. 2^{48}
D. 2^{64}

(o) (2 points) My website allows users to create usernames using any characters they want. It also displays this username on their profile page. What attack is this vulnerable to?

A. CSRF attack
B. XSS attack
C. SQL Injection attack
D. Brute Force attack

(p) (2 points) An iframe with different domains embedded in a webpage do not subject to same origin policy.

A. True
B. False

(q) (2 points) In the context of RSA, how would Alice digitally sign a document before sending it to Bob?

A. Encrypt the PRF with Alice's private key
B. Encrypt the PRF with Alice's public key
C. Encrypt the PRF with Bob's private key
D. Encrypt the PRF with Bob's public key

(r) (3 points) For each of the followings, identify one or more relevant security properties.
Choose from **C** (Confidentiality), **I** (Integrity), **A** (Availability).

____ Denial-of-Service

____ Man-in-the-Middle

____ RSA

____ One-time Pad

____ HMAC

(s) (2 points) Identify the access control design used for each of the followings.
Choose from **M** (MAC), **D** (DAC), and **R** (RBAC).

____ User's control over Facebook Privacy Settings (e.g. set the visibility of profile or photos to public, friends, or private)

____ Unix user-group-other permissions (e.g. -rwxr-xr-x)

____ Course subversion repository access control (e.g. students have access to _shared and their own directories only; staffs have access to all directories)

____ SELinux (Security-Enhanced Linux)

Question 2: Short Answer . . . . . *31 points*

(a) (2 points) Usually, we need two specific instructions at the end of a x86 function. What are they?

(b) (4 points) Identify and describe two defenses against rainbow tables.

(c) (4 points) Identify and describe two techniques for reverse code engineering (RCE).

(d) (3 points) Name 3 different kinds of code injection attacks.

(e) (5 points) CS461 course SVN repository used for MP submission is managed based on Access Control. It is set up to allow only authorized subjects to access permitted objects. Assume that you are the administrator to design access control list for users and groups accessing directories based on below rules. Fill in the access control table below with r (read), w (write) and/or - (none) to provide the most secured scenario.

- There are three different groups: Admins, Staffs, Students.
- Alice is a registered student, with netid "alice1".
- Bob is assigned as TA/course staff, with netid "bob2".
- Everybody can access and create files on their own directory with their netID.
- Admins are the owners who manage SVN repository and assign users and groups' permissions.
- Staffs are the course instructors and TAs who can access both public/shared, private areas.
- Staffs generate MP template files and runs the autogrder on the submissions on all staffs' and students' directory.
- Students are the rosters of the course who can access to shared area for downloading MP related files.
- Students submit MPs on their own directory with their netID.
- "private" folder is where only owner and the course staffs can access. Course staffs keep and update autograder in this folder.
- "roster" folder is created by owner and shouldn't be modified by any others except the owner. Course staffs use roster/staff list in this directory for assignment distribution and grading on all staffs' and students' directories.
- "shared" folder is open to anybody in the course. Course staffs upload necessary MP files, e.g. VMs, on this folder.

Directory	Admin	Staff	Student	Alice	Bob
private	r,w				
roster	r,w				
shared	r,w				
alice1	r,w				
bob2	r,w				

(f) (1 point) Suppose that while trying to access a collection of short videos on some Website, you see a pop-up window stating that you need to install this custom codec in order to view the videos. What threat might this pose to your computer system if you approve this request?

(g) (2 points) What is Kerckhoff's principle?

(h) (1 point) Assume that a block cipher operates on blocks of size 256 bits. What would be the length of the padding (in bits) generated by the algorithm if you apply the cipher to a message that is 64 bits long?

(i) (1 point) What is the downside of RSA compared to AES?

(j) (4 points) Alice wanted to send an encrypted message to Bob using 128-bit AES, but first she needs to share the AES key k with Bob. Alice knows Bob's 4096-bit RSA public key is (3, N), so she encrypted the key and sent $c_k = k^3 \bmod N$ to Bob. Then, she uses k to encrypt a message m with AES and send $c_m = AES_k(m)$ to Bob. Assume that the AES key is represented by an unsigned integer, and that RSA private key is stored securely, Explain how an eavesdropper who intercepts both messages can easily learn m and propose a way to fix the problem.

(k) (4 points) Let's say the ECE department wanted to share an "anonymous" logs of I-card swipe-in record for ECEB with a group of researchers. The department wants to ensure that the user's UIN aren't revealed, but the researchers need to be able to associate different swipe-in entries from the same person. The logs are huge, and anonymization has to be applied efficiently with only a small, fixed amount of storage. If the department replace each UIN with the SHA-256 hash of the UIN, is this sufficient to provide strong protection for the anonymity of the UINs? Briefly explain.

Question 3: AppSec MP Question . . . . . *30 points*

Consider the following function for all of the parts except for parts (g), (h), and (i):

```
void foo(char *arg)
{
  ...
  char buf[4];
  strcpy(buf, arg);
}
```

arg is a pointer to a char string that is the command line input from the user. Make these assumptions:

- The machine behaves just like the VM from MP1.
- All the defences mentioned in lectures are off
- You see the following information when the program arrives to the breakpoint at foo that you set earlier with the command `break foo`:
  - **buf** begins at 0xbffebfc8.
  - (gdb) x/2wx $ebp
    0xbffebfd8: 0xbffec064 0x08048fe5

(a) (2 points) Assume another function(e.g. main) is trying to call foo, what would be the size of the argument (number of bytes) that main needs to pass to foo?

(b) (2 points) What is the value of return address?

(c) (5 points) Describe parts of the input (**arg**) that you would give to the program to overflow the buffer (**buf**) and execute the same shellcode that was given for the MP. The shellcode is 23 bytes. Be specific and include exact numbers.

(d) (3 points) If instead, you see the following information when the program arrives to the breakpoint at foo that you set earlier with the command `break foo` :

- **buf** begins at 0xbffebfa0.
- (gdb) x/2wx $ebp
  0xbffebfd4: 0xbffec064 0x08048fe5

Would you need to change your solution from part(c) to achieve the same goal? Explain your answer.

(e) (8 points) Assume you are required to do return-oriented programming with the same piece of code, consider the following gadgets. The first column is the address in hexadecimal representation followed by the instruction at that address:

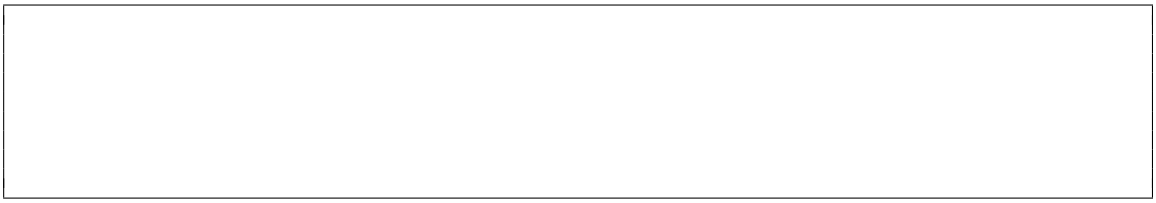
```
8051750: xor    %eax,%eax
8051752: ret

8057360: inc    %eax
8057361: pop    %edx
8057362: pop    %ecx
8057363: ret

8058680: cmp    $0xffffff83,%eax
8058683: jne    80586f8 <_exit>
8058689: pop    %eax
805868f: ret

8057ae0: int    $0x80
```

Assume these are the only gadgets that you can use. Your task is to set the value of eax to 2 then invoke a system call. System call number 2 is sys_fork, and the only argument it takes is stored in ebx. You can assume the argument has already been set up correctly. Draw a picture of the stack showing how you would chain the gadgets to complete your task. (Label the start of the chain and label which way is the top/bottom of the stack)

(f) (3 points) Continue from part(e): assume the second gadget has been changed to the following:

```
8057300: inc    %eax
8057301: pop    %edx
8057302: pop    %ecx
8057303: ret
```

Can you formulate a solution similar to part(e) to achieve the same goal? Why?



(g) (3 points) If stack canary was turned on for MP1.2, would your answers still work? Why?

(h) (3 points) Why would one want to use a callback shell(1.2.10) as the payload instead of a regular shell(shellcode.py)?

(i) (1 point) Which format specifier makes printf vulnerable to format string attack (1.2.11)?

Question 4: WebSec MP Question ..... *30 points*

Please refer to following python code for part (a), (b).
Recall that Bungle had a python code named database.py which processes user input into SQL queries. A Bungler implemented `validateUser()` function in `database.py` as shown below.

```
import MySQLdb as mdb
...

def validateUser(username, password):
db_rw = connect()
cur = db_rw.cursor()
username = mdb.escape_string(username) #escapes special characters
cur.execute("SELECT id FROM users WHERE username='"+username
            +"'AND password='"+password+"'")
if cur.rowcount < 1:
   return False
return True
```

(a) (2 points) Write an input pair `(username,password)` which bypasses authentication procedures and logs in as a user named `admin`.

(b) (3 points) Assume that this Bungler implemented all other parts of Bungle the same as you have done in MP2.1. Can an adversary attack Bungle by using SQL injection attacks like `DROP TABLE`? Why or why not?

(c) (4 points) Consider following PHP code which resembles one from 2.2.1.3.

```
if (isset($_POST['username']) and isset($_POST['password'])) {
    $username = $_POST['username'];
    $password = md5($_POST['password'], false);
    $sql_s = "SELECT * FROM users WHERE username='$username' and pw='$password'";
    $rs = mysql_query($sql_s);
    if (mysql_num_rows($rs) > 0) {
        echo "Login successful!";
    } else {
        echo "Incorrect username or password";
    }
}
```

Note that this code uses false as second parameter of `md5()` function so that the function returns the hash as a 32-character hexadecimal number in string. Is this code secure against any SQL injection? If not, provide an input pair `(username,password)` which bypasses authentication procedures by logging in as a user

named `admin`. In addition, suggest a solution to fix this code and protect against any SQL injection if you think this code is vulnerable.

(d) (2 points) Assume that Bungle uses GET for `\login` instead of POST. Write a URL so that when victim visits this URL, he would open a Bungle which is already logged in as a user named attacker with password l33th4x.

(e) (6 points) Recall in 2.2.2.2, you were asked to create an HTML file that, when opened by a victim, logs their browser into Bungle under the account `attacker`. In addition, you were asked to do this attack against a server which uses token validation mechanism. The server sets a cookie named `csrf_token` to a random 16-byte value and also include this value as a hidden field in the login form. When the form is submitted via POST, the server verifies that the client's cookie matches the value in the form.
Explain why exploiting Bungle's vulnerability on XSS was necessary for 2.2.2.2.

(f) (6 points) Write an injection script which will report all cookies of target website when injected via XSS. The script should report cookies, like you have done for Spying requirement on 2.2.3, to a URL with a format shown below where key$i$ is name of ith cookie and value$i$ is value of ith cookie.
`http://www.evilsite.com:31337/stolen_cookies?key1=value1&key2=value2...`
If you are not sure about exact syntax of any Javascript or jQuery function, you may use the function in pseudo-code style.

(g) (7 points) Recall that for 2.2.3, you were asked to demonstrate XSS attacks against Bungle's search box, which does not properly filter search terms before echoing them to the results page. Your goal was to construct a URL that, if loaded in the victim's browser, correctly executes the payload which required persistency as one of the requirements. Explain in detail how you made your payload persist when a user logged into Bungle with his account.