ECE 422 / CS 461, Midterm Exam

Tuesday, October 11th, 2016

Name:____

NetID:____

- Be sure that your exam booklet has 14 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	6	
4	16	
5	9	
6	12	
7	12	
9	11	
11	10	
12	13	
13	6	
14	5	
Total:	116	

Question 1: Multiple Choice . . . . *22 points*

For each question, circle *all* that apply. Some questions may have more than one correct answer.

(a) (2 points) ROP can bypass DEP/NX.

A. True

B. False

(b) (2 points) Which of the following has the same origin as **http://example.com/dev/index.html** ?

A. http://www.example.com

B. http://example.com

C. http://example.com:9000/index.html

D. https://example.com/dev/index.html

E. http://example.com/dev/about.html

(c) (2 points) ___________ exploits the trust a website has in a user's browser that includes credentials in requests, and ___________ exploits a web site that does not properly sanitize inputs.

A. CSRF, XSS

B. XSS, CSRF

C. XSS, Shellshock

D. Shellshock, XSS

E. Shellshock, SQL injection

(d) (2 points) Which of the following is/are NOT a phase of computer virus' lifecycle?

A. Propagation Phase

B. Triggering Phase

C. Contagious Phase

D. Dormant Phase

E. Immune Phase

(e) (2 points) Which of the following is/are true about rainbow tables?

A. Rainbow tables are used in password cracking.

B. Rainbow tables trade space for time by precomputing hashes.

C. Rainbow tables efficiently store all possible hashes.

D. Rainbow tables are an attack against salted passwords.

(f) (2 points) Given the Unix permission line "-rwxr-xr--", which of the following is/are true?

A. This is a permission line for a directory.

B. All users can execute.

C. Group can write.

D. Group can execute.

(g) (2 points) Worms are able to spread without human interaction.

A. True

B. False

(h) (2 points) Signature scanning is a malware defense method that is useful in identifying "zero day" malware.

A. True

B. False

(i) (2 points) Biometric authentication is an example of an authentication method based on what a user:

A. is
B. likes
C. knows
D. has
E. eats

(j) (2 points) Which of the following accurately describes a rootkit?

A. Rootkits self-replicate across systems.
B. Rootkits are a technique for exploiting a vulnerability to execute a payload.
C. Rootkits modify the operating system to hide their existence.
D. Rootkits are a means of ensuring malware persistence.

(k) (2 points) Which of the following is/are true about system call interposition?

A. Can only be implemented in kernel space
B. systrace is more efficient than ptrace
C. ptrace uses policy files to determine which systems calls to monitor
D. systrace cannot abort a system call without killing the application

Question 2: Short Answer . . . . *37 points*

(a) (3 points) You are given an application binary (i.e. no source code). Name and describe one method you would use to search for potential vulnerabilities in the software.

(b) (2 points) Name or describe one property that distinguishes a trojan from other kinds of malware.

(c) (2 points) Ben Bitdiddle writes a program using gets(). What attack might his code be vulnerable to?

(d) (2 points) List two defenses against SQL injection.

(e) (3 points) Name three common access control designs.

(f) (2 points) What do DEP and prepared statements have in common?

(g) (2 points) Give an example of input value for buffsize that would cause the following function to allocate and return a buffer longer than 100 bytes. Briefly explain your answer.

```
char* foo(int buffsize){
    int maxlen=100;
    if(buffsize>maxlen){
        buffsize=maxlen;
    }
    return malloc( ((unsigned)buffsize) *sizeof(char));
}
```

(h) (2 points) Worms and viruses are two kinds of malware that can replicate themselves. Please list two differences in their behavior.

(i) (2 points) Why is web-based advertising considered a potential security risk?

(j) (3 points) Name three mitigation techniques used to defend against buffer overflow control flow hijacking (just write the name or an abbreviation).

(k) (2 points) What might you look for in order to detect an encrypted virus?

(l) (3 points) Name three software technologies used to achieve confinement.

(m) (2 points) What is the same-origin policy? What role does it play in browser security?

(n) (2 points) Name two things you might overwrite with a heap overflow to execute arbitrary attacker code?

(o) (2 points) As an attacker, what type of virus would you create to defeat a defense that looks for patterns in your code? How would that virus bypass that defense?

(p) (3 points) Name three code injection attacks.

Question 3: AppSec MP Question . . . . *33 points*

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

```
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 0xbffe3658.
  - (gdb) x/2wx $ebp
    0xbffe3668: 0xbffe3750 0x0805cc46

(a) (2 points) The function strcpy takes 2 arguments, buf and arg. What is the size(number of bytes) of buf? What is the size(number of bytes) of arg?

(b) (2 points) What is the value of **previous** base/frame pointer?

(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 0xbffe3658.
- (gdb) x/2wx $ebp
  0xbffe3668: 0xbffe3880 0x0805cc46

What changes are necessary to be made so your solution from part(c) can still 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:

```
#gadget1
8051750:  xor    %ebx,%ebx
8051752:  ret

#gadget2
8057360:  inc    %ebx
8057361:  pop    %edx
8057362:  pop    %ecx
8057363:  ret

#gadget3
8058680:  cmp    $0xffffff83,%eax
8058683:  jne    80586f8 <_exit>
8058689:  pop    %ebx
805868a:  ret

#gadget4
8057cd0:  int    $0x80
```

Assume these are the only gadgets that you can use. Your task is to set the value of ebx to 0xbffe1200, so that ebx points to a struct(the ONLY argument to this system call, so feel free to clobber ecx and edx), then invoke a system call. **Assume that the struct and the value(s) for other register(s) have already been set up for the system call**. 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 gadget2 has been changed to the following:

```
8057360:  inc    %ebx
8057361:  pop    %edx
8057362:  pop    %eax
8057363:  ret
```

Does your solution from part(e) still work? Why?

(g) (2 points) Ben Bitdiddle is trying to complete MP1.1.5 but is unable to launch the shell. Please help him to find the bugs. (Assume address 0xbffffba0 is accessible)

```
movb $0xb,%ax           #0xb: sys_execve number
mov $0xbffffba0,%ebx
lea 8(%ebx),%ecx        #ecx=ebx+8(argv)
xorl %edx,%edx
movl $0x6e69622f,(%ebx) #/bin
movl $0x68732f,8(%ebx)  #/sh\x00
mov %ebx,(%ecx)         #argv[0]=/bin/sh
mov %edx,4(%ecx)        #argv[1]=NULL
int $0x80               #sys_execve()
```

(h) (3 points) If ASLR was turned on for MP1.2, would your answers for the MP still work? Why?

(i) (2 points) When would one want to construct a ROP attack(1.2.9) instead of redirecting control to a regular shellcode(shellcode.py)?

(j) (3 points) Which format specifier makes printf vulnerable to format string attack (1.2.11)? Why does that format specifier make printf vulnerable?

Question 4: WebSec MP Question . . . . *24 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`. He also added an extra sanitization step that replaces all single quotes in the password with two single quotes. The entire function is 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
password = str.replace("'", "''", password);
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) (2 points) Replace the call to cur.execute() with one that executes the same query, but in a way that protects against SQL Injection

(c) The next two questions refer to the same Bungle that you implemented for MP2.1, and attacked in MP2.2

i. (3 points) Could you log someone into an account on Bungle by creating a URL that executes a GET request? Explain why or why not.

ii. (6 points) On Bungle, when you set `csrfdefense` level to 1, what sort of changes happened to the page, and how the site works? What sort of security does this change provide to Bungle?

(d) Imagine that `http://www.faceboard.com` has a form which allows you to send messages to a user. When Alice sends sends a message to Bob via this form, the website receives a GET request to `http://www.faceboard.com` with parameters listed below.

```
send_to: "bob"
message_content: "Hi"
```

i. (1 point) Malory wants to exploit this request mechanism. Write a URL so that when that URL is clicked by Alice, she will send the message ”Ihateyou” to Bob.

ii. (5 points) What if `http://www.faceboard.com` also uses the same defenses that setting the `csrfdefense` level to 1 on Bungle will provide. What if there is a different form on `http://www.faceboard.com` that is vulnerable to XSS attacks. How can Malory use this form to get around the security that the `csrfdefense` level provides, and execute her attack.?

(e) (5 points) Recall for 2.2.3, you had to report on all user events that occurred, such as nav and login/logout. Lets say you have a website with a login form which has both username and password input fields. You have managed to find an XSS vulnerability, and you are trying to exploit it to get the usernames/passwords of everyone who tries to login into the site.

Write a reporting script that will report the values in the username and password input fields to the following URL:
`http://www.evilsite.com/stolen_data`
The data should be sent in the same format as what you used for 2.2.3. The username input field has an id of `username`, and the password input field has an id of `userpass`. Use these ids to identify the username and password respectively when you send a request to the URL.

You can assume that this script will automatically get called when a user presses the login button, so you only have to worry about the reporting part. If you are not sure about exact syntax of any Javascript or jQuery function, you may use the function in pseudo-code style.