ECE 422 / CS 461, Midterm Exam Study Guide

Name:_____

NetID:_____

- Be sure that your exam booklet has 15 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	10	
3	12	
4	4	
5	9	
6	9	
7	4	
8	8	
9	9	
10	8	
11	8	
12	6	
13	6	
14	8	
15	4	
Total:	105	

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

For each question, circle all that apply.

(a) (1 point) Confidentiality ensures anonymity.

A. True
B. False

(b) (1 point) Malware that propogates itself without any human interaction is called:

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

(c) (1 point) An attacker places the address of a series of gadgets on the stack. What is she doing?

A. Return oriented programming
B. Smashing the stack
C. Formatted string attack
D. Dictionary attack

(d) (1 point) If a file should have permissions read/write for owner, read for group, and write for others, what should the permission bits look like?

A. -rwxr—w-
B. -rw-r—w-
C. -rw–w-r–
D. –w-rw-r–

(e) (1 point) In MP1, you used a buffer overflow attack to result in transferring control to your shellcode. What did you overwrite that would result in the program transferring control to your shellcode?

A. Local variables
B. Saved base pointer
C. Return Address
D. Function arguments

(f) (1 point) Consider the following C function signature:

```
void foo(int var1, int var2, int var3)
```

In the 32-bit C calling convention learned in class, which of the following correctly describes how parameters are passed to the function?

A. Pushed onto the stack in this order: var1, var2, var3
B. Pushed onto the stack in this order: var3, var2, var1
C. Placed in registers: var1 in EAX, var2 in EBX, var3 in ECX
D. Placed in registers: var3 in EAX, var2 in EBX, var1 in ECX

(g) (2 points) What is a good source of randomness?

A. Time to boot up the operating system in seconds.
B. Ambient noise in the room.
C. A random seed generated a month ago.
D. 32-bit word stored at memory address 0x1000.
E. None of the above.

(h) (2 points) Diffie-Hellman key exchange will assure a secure connection between two trusted parties.

A. True
B. False

(i) (2 points) Sending a message in the presence of an eavesdropper without revealing the contents of the message itself is ensuring which aspect(s) of security?

A. Confidentiality
B. Integrity
C. Availability
D. Authenticity

(j) (2 points) A digital signature is used to ensure which aspect(s) of security?

A. Confidentiality
B. Integrity
C. Availability
D. Authenticity

(k) (2 points) In MP3, you convinced us that you correctly "guessed" the random number by exploiting one of the MD5 vulnerabilities. Which attack did you use to accomplish this?

A. Pre-image attack
B. Collision attack
C. Length extension attack
D. Birthday attack
E. Rainbow attack

(l) (2 points) In the previous question's "guessing" scenario, which security property did the attack compromise?

A. Confidentiality
B. Integrity
C. Availability
D. Authenticity
E. Accountability

(m) (2 points) P(m) is an application of a RSA public key on message m. K(m) is an application of a RSA private key on message m. P(K(P(K(P(P(K(P(K(K(m)))))))))) results in m.

A. True
B. False

(n) (2 points) Since there are 10000 possibilities for a 4 digit PIN, in real life 1234 is the pin for about 0.01% of people's credit cards.

A. True
B. False

Question 2: Short Answer ..... *26 points*

(a) (2 points) A novice programmer has written the code "movb $11, %ax; int $0x80", expecting execve to be called, but that did not happen, explain why.

there might be something in the top half of eax.

(b) (1 point) In MP1, the spec introduced a helper function called `pack("<I", addr)` when writing a solution in python. Why would one need to modify each word with pack()?

x86 stores words in little endian, while our strings are by endian

(c) (1 point) Why is strcpy more vulnerable than strncpy?

strcpy only breaks on [redacted] \0, while strncpy also stops copying when it hits n bytes.

(d) (2 points) When writing shellcode, an adversary is prevented from using some specific characters. Provide an example and describe why.

NOA, it is the newline character and will break shellcode. (gets)

(e) (2 points) What is Data Execution Prevention (DEP)? What is a similar conceptual protection measure that prevents SQL injection in web programming?

prevents memory outside of code segments from being executed.
prepared statements

(f) (2 points) Although DEP is a strong protection measure against stack smashing, implementing only DEP still leaves a room of vulnerability against advanced stack smashing. What kind of attack is it still vulnerable against? Why?

ROP - the attacker can still use gadgets from code segment to execute code.

(g) (1 point) Assuming you have answered problem (k) correctly, suggest an additional protection measure which could strengthen your system against the attack from part (k).

ASLR - randomizes location of stack

(ASMR)

(h) (2 points) Describe the dormant phase and action phase of a computer virus.

dormant - waits and does nothing
action - executes payload

(i) (3 points) Identify three access control designs.

Mandatory - central admin decides
discretionary - users decide permissions (user+group) most common in OS.
role-based - permissions defined by resource and role

(j) (4 points) Name two properties of a viable hash function.

(k) (2 points) Why is the Merkle-Damgard construction susceptible to length extension attacks? Explain.

(l) (4 points) List two drawbacks of RSA.

Question 3: Symmetric and Asymmetric Cryptography . . . . . *8 points*

Client Alice wants to send a message *M* to Bob. Assume Alice and Bob share a symmetric key *K* and have securely distributed their public keys *P_A* and *P_B* to each other. Private keys of Alice and Bob are *S_A* and *S_B* respectively. Design messages that Alice must send to meet the security requirement below.

Notation:

- $x \parallel y$ (concatenation)
- $\{x\}_y$ (x is encrypted using key y)
- $MAC_y(x)$ (MAC of x using key y)
- $A \xrightarrow{x} B$ (A sending x to B)

Examples:

- $A \xrightarrow{M} B$ The message *M* is sent from Alice to Bob
- $A \xrightarrow{\{S\_A \| M\}_{S\_A}} B$ The message *M* is concatenated with Alice's private key *S_A* and the resulting concatenation is encrypted with Alice's private key *S_A*. The encrypted message is sent to Bob.

(a) (2 points) Using the symmetric key, design a message that enables Bob to verify the message is from Alice where only integrity is preserved.

(b) (2 points) Using public key cryptography, design a message that enables Bob to verify the message source, Alice, and preserves only integrity.

(c) (2 points) Using public key cryptography, design a message that protects only the confidentiality of the message sent from Alice to Bob.

(d) (2 points) Using public key cryptography, design a message that enables Bob to verify the message source, Alice, and when both integrity and confidentiality are protected.

Question 4: Code Injection ..... *9 points*

(a) (2 points) What is a fundamental problem of any code injection attack?

finding a way to execute arbitrary code on a target system (get code, expect data)

(b) (2 points) What is Shellshock?

Bash executing arbitrary code from env variables

(c) (2 points) Consider following php code snippet for SQL query.

```
$query = "SELECT * FROM users WHERE id='$id'"; //type of id is an integer
```

This query is undoubtably vulnerable against any SQL injection. Explain a protection measure we can take to protect this code against SQL injection.

real_escape_strings
prep stmt

(d) (3 points) A new web application has a page named "faceboard" which is composed of a list of comments. Any user can anonymously write a comment, which can be viewed by any visitor of the webpage. When the webserver of this application receives a message input from any user, the backend interprets and/or sanitizes the input using a protection measure you have suggested in part c. After applying this security measure, is "faceboard" secure against adversaries? If not, list one vulnerability and a security measure which can improve the protection of "faceboard".

no
CSRF → token validation

Question 5: Web Application Security . . . . . *8 points*

(a) (3 points) Which of following URLs share the same origin with http://www.cs461.com/dir/page1.html?

(a) `http://www.cs461.com/dir2/page2.html`
(b) `http://www.cs461.com/dir/dir3/page3.html`
(c) `http://www.cs461.co.kr/dir/page1.html`
(d) `https://www.cs461.com/dir/page1.html`
(e) `http://cs461.com/dir/page1.html`
(f) `http://en.cs461.com/dir/page1.html`
(g) `http://username:password@www.cs461.com/dir/page1.html`

a, b, g

(b) When Alice sends 100.00 dollars to Bob via http://www.bank.com, the website receives a GET request to http://www.bank.com with parameters listed below.

```
to_username: "bob"
transaction_type: "transfer"
amount: 100.00
```

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 200.00 dollars to Malory.

http://bank.com?to_username=malory&transaction_type=transfer&amount=200.00

ii. (2 points) Does changing type of request from GET to POST solve the problem? Explain.

no, you would need to implement some other kind of validation

(c) (2 points) A website uses token validation in order to prevent CSRF attack. The website generates the token using a rand() function which generates a pseudorandom number from 0 to RAND_MAX. What is a potential problem for this website? Assume this website is secure against any other type of attacks including XSS.

generate several tokens and figure out the pattern

Question 6: Applied Cryptography . . . . . *14 points*

(a) (2 points) Assume that a block cipher operates on blocks of size 512 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 128 bits long?

(b) (2 points) Assume that a block cipher operates on blocks of size 512 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 1024 bits long?

(c) (4 points) Recall that a one-time pad is a symmetric encryption scheme where a random bit string of the same length as the message is generated to be use as a key, and each bit c_i of the encrypted message is compute by $c_i = m_i$ XOR k_i, where m_i is the i^{th} bit of the message, and k_i is the i^{th} bit of the key. Why do we use XOR instead of other logic operation such as AND or OR?

(d) (4 points) Consider the following hash function:

```
def strong_hash(m):
    hash_val = 0xFF
    for each byte of m:
        hash_val = hash_val XOR byte

    return hash_val
```

The hash function basically compute a 8-bit digest of the message by computing the XOR of each byte in the message, then XOR the result with 0xFF. Also, recall that a hash function is second-preimage resistance if given x, it is hard to find $x' \neq x$ such that $strong\_hash(x) == strong\_hash(x')$. Is strong_hash second-preimage resistance? If yes, explain why and if not, find the second preimage x' for x = 0xAA.

(e) (2 points) In MP1 checkpoint 2, We ask you to find the private key of an RSA key pair given a public key and RSA modulo of a 2048-bit RSA. As the size of the modulo is 2048 bits, it is not feasible to try to factorize the modulo to find the two prime roots, so we suggest that you use Wiener's attack to recover the private key. What is the weakness in our RSA keypair that allows Wiener's attack to work?

Question 7: AppSec MP Question (MP-specific) . . . . . *18 points*

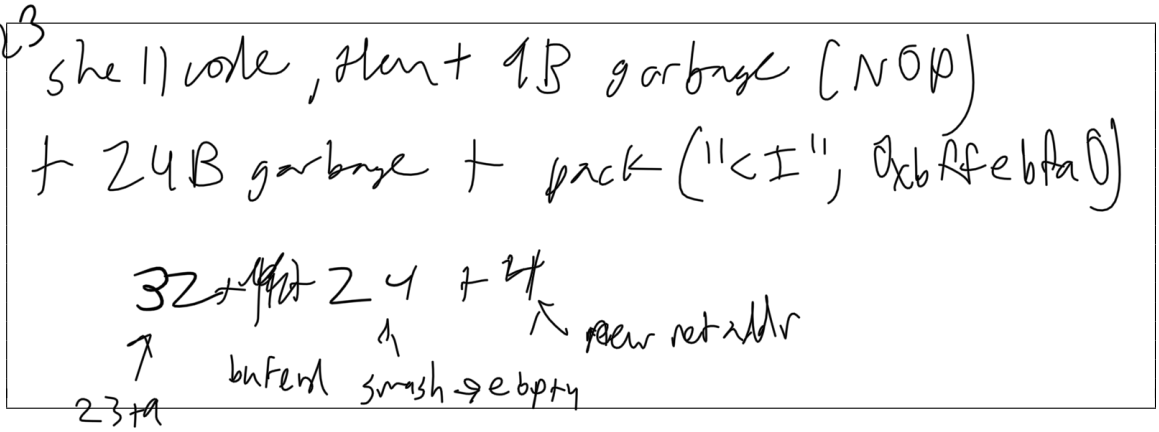
(a) (4 points) Consider the following function:

```
void foo(char *arg)
{
  ...
  char buf[32];
  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 is a 32-bit little-endian system that behaves 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 0xbffebfa0.
  - (gdb) x/2wx $ebp
    0xbffebfd8: 0xbffec064 0x08048fe5

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 file shellcode.py has size of 23 bytes. Be specific and include exact numbers.



(b) (2 points) Continue from part(a): 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
  0xbffebfd8: 0xbffec064 0x08034586

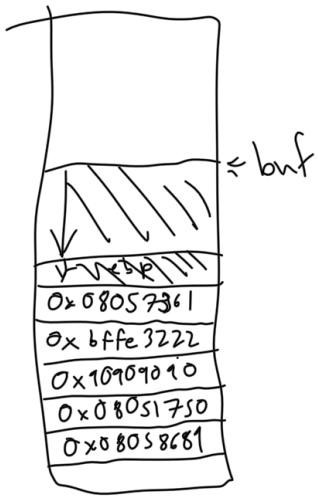
Would you need to change your solution from part(a) to achive the same goal? Explain your answer.

no, the return address is not relevant, only its address matters.

(c) (4 points) Consider the following gadgets. The first column is the address in hexadecimal representation, and the second column is the instruction at that address:

```
1. → 8051750: xor    %eax,%eax
     8051752: ret

     8058680: cmp    $0xffffff83,%eax
     8058683: jne    80586f8 <_exit>
2. → 8058689: mov    %eax,(%ecx)
     805868f: ret

     8058679: mov    %ecx,(%eax)
     805867f: ret

     8057360:        pop    %edx
0. → 8057361: pop    %ecx
     8057362: pop    %ebx
     8057363: ret

     8057ae0: int    $0x80
```

Assume these are the only gadgets that you can use, how would you set the value at memory address 0xbffe3222 to be 0x00000000 (a null pointer)? Draw a picture of the stack showing how you would chain the gadgets. (Label the location of the original return address and label which way is the top/bottom of the stack)



call ret

enter / leave → ebp

(d) (4 points) Continue from part(c): assume the value at memory address 0xbffe3222 is originally 0xc0a8ea66, and you want to change the value to 0x00000066. How do you need to modify your answer from part(c) to achieve this goal?

we need to shift and zero out the top 3 bytes 0xbffe3223

(e) (2 points) Why would one want to use a callback shell(4.2.10) as the payload instead of a regular shell(shellcode.py)?

In case the attacker does not have physical access to the machine

(f) (2 points) If ASLR was on for MP1.2, would your answers still work? Why?

No, because