CS 161
Computer Security

Dawn Song
Fall 2012

Midterm 1

Your Full Name: _____

Your Berkeley Email: _____

This is a closed-book midterm. You may use one letter sized paper as a cheatsheet. Apart from your own cheatsheet, you may not consult any lecture or written notes, textbooks, etc. Calculators and computers are not permitted. Please write your answers in the spaces provided in the test. We will not grade anything on the back of an exam page unless we are clearly told on the front of the page to look there.

You have 80 minutes. There are 5 questions, of varying credit (73 points total). The questions are of varying difficulty, so avoid spending too long on any one question.

Unless mentioned otherwise, for a true/false question asking for a reason, half of the points are for the correct option (true or false), and the remaining half is for the correct reason.

Do not turn this page until your instructor tells you to do so.

Write your name on all pages.

Question	Points	Total
Problem 1		20
Problem 2		16
Problem 3		20
Problem 4		9
Problem 5		8
Total		73

Name _____

1. (20 points) Memory Safety Reasoning Techniques

   Consider the following program:

```
1: void foo (unsigned char x, unsigned char y) {
2:   unsigned char buf[3], z;
3:
4:   z = x + 5;
5:   z = z * 2;
6:   if (z <= 4)
7:     buf[z] = y;
8: }
```

   (a) (2 points) Write down the assertion that you would insert at (just before) line 7 to ensure memory safety.

Answer: assert(z <= 2); or equivalent.

0 <= z is redundant because z is unsigned and cannot be negative.

   (b) (8 points) Using symbolic execution, what is the path constraint formula to solve for checking memory safety at line 7? You may optionally express your answer in Static Single Assignment (SSA) form if desired.

Name _____

Answer:

$$
\begin{aligned}
&(z_1 == x_0 + 5) && (1)\\
&\wedge\, (z_2 == z_1 * 2) && (2)\\
&\wedge\, (z_2 <= 4) && (3)\\
&\wedge\, (z_2 > 2) && (4)
\end{aligned}
$$

2 pts for each line

Name _____

(c) (2 points) By solving your formula in the above question in bitvector arithmetic, write down an instance of values for inputs x and y if the formula is satisfiable. If the formula is unsatisfiable, write "unsatisfiable".

Answer: x == 253 (or 125) and y can be any value in [0,255].
Equivalents are accepted. Minor off-by-one calculation errors are ok. 1 pt for each

(d) (4 points) Write down the precondition to `foo` that must hold to ensure memory safety. If the program requires no precondition to be memory safe, write "true". You may use the space below to work out your reasoning, but only the answer written on the line is graded.

Answer: x != 253 && x != 125 or equivalent.

(e) (4 points) Using blackbox fuzzing, each test case is a call to `foo` with a random value of (x, y). For *each test case*, what is the probability that the blackbox fuzzer would find a memory violation?

Answer: 1/128. There are only two values of x out of 256 possible values that trigger the violation, and y can be any value in [0,255].

Name _____

2. (16 points) Memory Safety

(a) (2 points) Consider the following program:

```
void foo(char *args)
{
  int x = 0;
  char buf[16];
  int y = 1;
  int z = 2;
  strncpy(buf, args, 24);
}

int main(int argc, char *argv[]) {
  foo(argv[1]);
  return 0;
}
```

Assuming a 32-bit x86 architecture like the one used for Lab 1 and a stack frame with space allocated only for the variables shown, which of the following variables (if any) can be overwritten by `strncpy`? **(circle all that apply)**

- ○ `int x`
- ○ `int y`
- ○ `int z`
- ○ None of the above

Answer: Only x can be overwritten because the stack is written upwards from the location of buf.

(b) (2 points) How many bytes of the saved frame pointer can be overwritten?

Answer: We can copy a total of 24 bytes into buf. 16 are allocated for the buffer; 4 for x; the remaining 4 are the saved frame pointer. So 4 bytes.

(c) (2 points) How many bytes of the return instruction pointer can be overwritten?

Name _____

Answer: 0 bytes

Name _____

(d) (4 points) Assuming an attacker controls the input to the program shown, a successful buffer overflow can be launched.

- ○ True. Reason: _____
- ○ False. Reason: _____

Answer: True. By overwriting the saved frame pointer, a false frame can be constructed that points to the buffer. This is similar to exploit2 in the first lab. Answers must have included a mention of the saved frame pointer. Also accepted false *only* if students mentioned there is not enough room to in the buffer to inject shellcode. In practice, this is not true because you could redirect the frame to the args provided on the command line (or use arc injection) instead of to buf, but that nuance was not part of the lab.

(e) (2 points) No longer considering the previous program, given a stack overflow, the use of a No-Execute (NX) bit prevents running shellcode provided by the attacker as an input into a stack buffer.

- ○ True. Reason: _____
- ○ False. Reason: _____

Answer: True. The NX bit prevents code on the stack from being executed. An attacker must instead use arc-injection.

(f) (2 points) Return-to-libc (arc-injection) is a viable technique to use to defeat stack canaries.

- ○ True. Reason: _____
- ○ False. Reason: _____

Answer: False. Return orientated programming still requires overwriting the return instruction pointer which would mean overwriting the stack canary. I also accepted True if exception handlers or format string vulnerabilities (or some similar attack) were mentioned as means to avoid overwriting the canary.

(g) (2 points) Your choice of programming language cannot help reduce security risks.

Name _____

- ○ True. Reason: _____
- ○ False. Reason: _____

Answer: False. A quick example would be using memory-safe languages which prevent arbitrary pointer access or unbounded copies. Examples include Python, C-sharp and Java.

Name _____

3. (20 points) Cryptography

   (a) (4 points) Consider two parties – Alice and Bob – trying to communicate securely. Alice sends a message $Auth_K(Enc_K(m)), Enc_K(m)$ which is both authenticated and encrypted. (Assume that encryption and authentication are performed properly.) Which of the following statements hold?

      i. Alice's communication with Bob is secure from a passive attacker:

         ◦ True. Reason: _____

         ◦ False. Reason: _____

         Answer: True. Encryption is enough to secure a channel against a passive attacker.

      ii. Alice's communication with Bob is secure from an active attacker:

         ◦ True. Reason: _____

         ◦ False. Reason: _____

         Answer: False. In order to establish a secure channel, Alice must both encrypt and authenticate a message as described, but also include a *nonce* to prevent replay attacks. Other acceptable answers include how the re-use of the same key for encryption and authentication may lead to leaking information; that you should authenticate what you mean, not the encrypted copy.

   (b) (8 points) Consider two messages each of n blocks m_i: $M_1 = m_1||m_2||..||m_n$ and $M_2 = m'_1||m_2||..||m_n$, where M_1 and M_2 differ only in the first block. Which of the following statements hold? Assume the same key is used for all encryption.

      i. If M_1 and M_2 are encrypted using electronic code book mode (ECB), none of the cipher text blocks will repeat between each message.

         ◦ True. Reason: _____

         ◦ False. Reason: _____

         Answer: False. In ECB, each message is encrypted independently, so c_2 through c_n will repeat between M_1 and M_2.

Name _____

ii. If M_1 and M_2 are encrypted using cipher block chaining (CBC), but the same initialization vector (IV) is used for encrypting both messages, none of the cipher text blocks will repeat between each message.

- ○ True. Reason: _____
- ○ False. Reason: _____

Answer: True. Even though the IV repeats, the first part of the messages differ and this variation will cascade through all subsequent blocks.

iii. If M_1 and M_2 are encrypted using counter mode (CTR), but the same initialization vector (IV) is used for encrypting both messages, none of the cipher text blocks will repeat between each message.

- ○ True. Reason: _____
- ○ False. Reason: _____

Answer: False. Repeating the same IV will mean the subsequent k_1 through k_n repeat, and thus $m_i \; XOR \; k_i = c_i$ will be the same for $i \in \{2, n\}$

iv. If M_1 and M_2 are encrypted using counter mode (CTR) and a unique (IV) is used when encrypting each message, none of the cipher text blocks will repeat between each message.

- ○ True. Reason: _____
- ○ False. Reason: _____

Answer: True. This is the correct operation of CTR mode. The keys will differ for both messages so there should not be any redundant ciphertext.

(c) (8 points) Consider a new alternate package management system to `yum` or `up2date` that distributes Linux packages. Explain whether the following approaches are secure against an active attacker such that the binaries a client downloads from a package management server have not been tampered with.

i. The server encrypts the binary using a 128-bit AES key known only to the client and the server and sends the client the resulting ciphertext.

- ○ Secure. Reason: _____
- ○ Insecure. Reason: _____

Answer: Insecure. Encryption does not provide any form of authentication; an attacker can easily flip random bits in the encrypted message, resulting in a different binary for the client. Also accepted answers about how an attacker could just download the same package manager and obtain the symmetric key. (The intent was

the key be secret foreach each client). Note that many people spoke on how AES is broken; this is not true!

ii. The server signs a SHA-1 (160-bit) hash of the binary using its RSA private key, while the associated RSA public key is advertised on the server's website and accessed by the client using HTTP.

- ○ Secure. Reason: _____
- ○ Insecure. Reason: _____

Answer: Insecure. When viewing the website under HTTP, the advertised public key can be swapped with one belonging to an attacker. The SHA-1 hash can then be recomputed by the attacker and signed with this fake identity.

Name _____

iii. The server generates an HMAC of the binary using a key known only to the client and the server and distributes the binary along with the HMAC.

- ○ Secure. Reason: _____
- ○ Insecure. Reason: _____

Answer: Secure. While impractical to have a secret key between all clients and the server, this is secure. HMAC will produce a digest for the binary, while an attacker will be unable to produce a similar digest for their malicious binary without knowing K. Also accepted answers about how an attacker could just download the same package manager and obtain the symmetric key, breaking this system. (The intent was the key be secret foreach each client).

iv. The client receives a binary along with a SHA-256 hash signed by the server's public key which is in turn signed by a certificate authority. For you answer, consider whether this is secure for all possible certificate authorities.

- ○ Secure. Reason: _____
- ○ Insecure. Reason: _____

Answer: Insecure. One correct answer is to note that the binary is signed by the server's public key. This is encryption, not authentication! Anyone can encrypt a binary and send it to the client. The client can't even decrypt the contents so this system is unusable. A second correct answer is that a CA can be compromised, resulting in the signature scheme becoming insecure.

Name _____

4. (9 points) Software based Fault Isolation (SFI)

   (a) (3 points) Which of the following should use Software based Fault Isolation (SFI)?. (Circle all that apply)

   - ○ An operating system kernel using device drivers which are loaded into the same address space as the main kernel.
   - ○ An operating system kernel where device drivers run as separate processes and they communicate with the kernel via inter-process communication.
   - ○ A 3rd party browser plugin, where both the browser and the plugin execute in the same address space.

   Answer: a, c. One point each for choosing a, not choosing b, and choosing c

   (b) (6 points) **SFI for a new architecture:**
   Let us assume that we have a hypothetical 32-bit architecture. In this hypothetical architecture, we have support for partitions, where partition is a contiguous 256MB subrange of the 32-bit virtual address space. Also, we have an additional register called partition-register (`$epr`). This partition register can be set to the base address of a partition, and the memory addresses in that partition can be referenced as an offset to the value in the partition-register. In case the offset value is specified at runtime using a register, only the lower 28 bits are used.

   As an example, if

   $$\$epr = 1010\ 0000\ 0000\ 0000$$
   $$\$eax = 0101\ 1111\ 1111\ 1111$$

   then an instruction like '`mov $ebx, [$eax]`' translates to '`mov $ebx, (1010 1111 1111 1111)`'. (`[$eax]` is a shorthand for specifying memory address relative to the partition-register).
   We now want to implement SFI for this new architecture. What conditions would you enforce on the load/store instructions in the untrusted code so that there are no loads or stores outside of the untrusted data by the untrusted code? Be as *efficient* as possible.

Answer: Restrict untrusted data to one partition and set the base address in $epr, force all memory references in load-/store instructions to be relative to $epr (i.e. in [$reg] form), and prevent writes to $epr by untrusted code. Full points only if all points mentioned

Name _____

5. (8 points) Secure Architecture Principles

   (a) (2 points) Suppose all system calls from any untrusted process (e.g., A) are redirected to a reference monitor process B. When A makes a system call open() on the file (e.g., "data.dat"), the request is redirected to B, which looks up the policy for A. If A is allowed to access "data.dat", B makes the open() system call on behalf of A with the requested and allowed privileges, gets the file handle h from the kernel, and passes the file handle to A. Then, all future read() and write() system calls on h are passed straight to the kernel. A can pass h to some other process C if it chooses.

   Which of the following best describes this security architecture?

   - ○ Access control is based on ACL (access control list).
   - ○ Access control is based on capabilities.
   - ○ Access control is based on both ACL and capabilities.
   - ○ Access control is not based on ACL nor capabilities.

   Answer: c.

   (b) (2 points) Suppose all system calls from any untrusted process (e.g., A) are redirected to a reference monitor process B. When A makes a system call open() on the file (e.g., "/etc/data.dat"), B confines A to its local root directory by prepending "/tmp/A/" to A's requested file path, and then passes the request to the kernel.

   Which of the following is true for this security architecture?

   - ○ A may be able to access "/etc/data.dat".

     Reason: _____

   - ○ There is no way for A to access "/etc/data.dat".

     Reason: _____

   Answer: a. B may still be able to access "/etc/data.dat" by requesting to open "../../etc/data.dat", which is prepended to "/tmp/A/../../etc/data.dat" by the reference monitor and then resolved to "/etc/data.dat" by the kernel.

Name _____

(c) (2 points) Suppose the reference monitor process B wants to allow an untrusted process A to access a file (e.g., "data.dat") only if a privileged process (e.g., `root`) has *not* yet written to that file. When A requests to access the file "data.dat", B checks if `root` has ever written to "data.dat". If `root` has never written to "data.dat", B grants A access. Otherwise, B denies that request. In contrast, `root` may write to any file whenever it wants without any restriction.

Which of the following is true for this security architecture?

- ○ A may be able to access "data.dat" after `root` writes to it.

  Reason: _____

- ○ There is no way for A to access "data.dat" after `root` writes to it.

  Reason: _____

Answer: a. It might be possible for A to access "data.dat" after `root` writes to it due to TOCTOU.

(d) (2 points) In the Google Chrome browser architecture, the browser is separated into the following components:

(a) A single **browser kernel** process with full system and network access privileges;

(b) Multiple sandboxed **rendering engine** processes with limited privileges, where each rendering engine renders up to 20 different websites, and all network and file system accesses are redirected to the browser kernel through IPC (inter procedure calls);

(c) Multiple **plugin** processes, each with full system and network access privileges.

Which of the following are true for this security architecture (**circle all that apply**)?

- ○ If there is a buffer overflow vulnerability in the rendering engine, an attacker might be able to modify the appearance of another website sharing the same rendering engine.
- ○ If there is a buffer overflow vulnerability in the plugin, an attacker might be able to control the browser kernel.
- ○ If there is a buffer overflow vulnerability in the browser kernel, an attacker might be able to modify the appearance of a website.

Answer: a, b and c. Full pts for selecting all, 1 pt for selecting at least one.