UNIVERSITY OF EDINBURGH

COLLEGE OF SCIENCE AND ENGINEERING

SCHOOL OF INFORMATICS

INFR10058 COMPUTER SECURITY

Friday 14th August 2015

14:30 to 16:30

INSTRUCTIONS TO CANDIDATES

Answer any TWO questions.

All questions carry equal weight.

CALCULATORS MAY NOT BE USED IN THIS EXAMINATION

Year 3 Courses

Convener: S. Viglas
External Examiners: A. Cohn, T. Field

THIS EXAMINATION WILL BE MARKED ANONYMOUSLY

1. [**Cryptography**]

(a) [**One-time pads**] Inspired by the one-time pad, Alice decides to design her own protocol to send messages confidentially to Bob. Alice's protocol works as follows:

- When Alice is ready to send her message $M \in \{0,1\}^\ell$, she randomly selects $K_A \in \{0,1\}^\ell$, and sends to Bob the message $M_1 = M \oplus K_A$.
- Bob then randomly selects $K_B \in \{0,1\}^\ell$ and sends to Alice the message $M_2 = M_1 \oplus K_B$.
- Next, Alice computes $M_3 = M_2 \oplus K_A$ and sends it to Bob.
- Bob may now retrieve the message M.

i. Show that $M = M_3 \oplus K_B$. [*2 marks*]

ii. This protocol is insecure. Show that Eve can retrieve any message intended for Bob. [*5 marks*]

(b) [**ElGamal**]

i. Recall the details of the ElGamal encryption scheme seen in class. [*4 marks*]

ii. Assume you are given an ElGamal public key pk (but not the corresponding private key). Assume you are also given the ciphertexts $c_a = E(pk, m_a)$ and $c_b = E(pk, m_b)$ corresponding to the encryption using ElGamal of messages m_a and m_b under pk respectively. But you are not give m_a nor m_b. Show how you can construct a ciphertext which is a valid ElGamal encryption under the key pk of the message $m_a \cdot m_b \pmod p$. [*7 marks*]

iii. Assume you are given an ElGamal public key PK (but not the corresponding private key) and a ciphertext $c = E(pk, m)$ which is the ElGamal encryption of some unknown message m under pk. You are furthermore given access to an oracle that will decrypt any ciphertext other than c. ElGamal is said to be vulnerable to a chosen ciphertext attack if you can retrieve m. Show that ElGamal is indeed vulnerable to a chosen ciphertext attack. [*7 marks*]

2. [**Memory safety**] Consider the following code

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

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

(a) Depict the stack frame of a call to `vuln` before the execution of the `strncpy` command. You will assume that the code is executed on a 32-bit machine, with the size of the `int` data type being 4 bytes, and the size of the `char` data type being 1 byte. [*5 marks*]

(b) Which of the variables `x`, `y`, and `z` (if any) can be overwritten by `strncpy`? Explain your answer. [*3 marks*]

(c) How many bytes of the saved frame pointer can be overwritten? Explain your answer. [*3 marks*]

(d) How many bytes of the return address pointer can be overwritten? Explain your answer. [*3 marks*]

(e) Assume that the input to `vuln` is controlled by an attacker. Can the attacker launch a successful control hijacking attack? Explain your answer. [*5 marks*]

(f) Recall what is a `return-to-libc` attack. Do stack canaries defeat `return-to-libc` attacks? Explain your answer. [*6 marks*]

3. **[CSRF defenses]**

(a) In class we discussed Cross Site Request Forgery (CSRF) attacks against web sites that only rely on cookies for session management. Recall what a CSRF attack is. [*7 marks*]

(b) A common CSRF defense is to include a token in the DOM of every page (*e.g.* as a hidden form element) in addition to the cookie. An HTTP request is accepted by the server only if it contains both a valid HTTP cookie header and a valid token in the POST parameters. Why does this prevent the attack from question 3a? [*6 marks*]

(c) One approach to choosing a CSRF token is to choose it at random. Suppose a web site chooses the token as a fresh random string for each HTTP response. The server checks that this random string is present in the next HTTP request for that session. Does this prevent CSRF attacks? If so, explain why. If not, describe an attack. [*6 marks*]

(d) Another approach is to choose the token as a fixed random string chosen by the server. That is, the same random string is used as the CSRF token in all HTTP responses from the server. Does this prevent CSRF attacks? If so, explain why. If not, describe an attack. [*6 marks*]