UNIVERSITY OF EDINBURGH

COLLEGE OF SCIENCE AND ENGINEERING

SCHOOL OF INFORMATICS

INFR10067 COMPUTER SECURITY

Monday 6 $\frac{\text{th}}{\text{}}$ August 2018

14:30 to 16:30

INSTRUCTIONS TO CANDIDATES

Answer any TWO of the three questions. If more than two questions are answered, only QUESTION 1 and QUESTION 2 will be marked.

All questions carry equal weight.

CALCULATORS MAY NOT BE USED IN THIS EXAMINATION

Year 3 Courses

Convener: C. Stirling
External Examiners: S.Rogers, A. Donaldson, S. Kalvala

THIS EXAMINATION WILL BE MARKED ANONYMOUSLY

1.
```
# Alice's computer
# Generated by iptables-save v1.6.0
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
-A INPUT -p tcp -m tcp --dport 22 -j ACCEPT
-A INPUT -p udp -m udp --dport 4444 -j DROP
-A INPUT -p udp -m udp --dport 4444 -j ACCEPT
-A INPUT -p tcp -m tcp -s 192.168.1.102 --dport 443 -j ACCEPT
-A INPUT -p tcp -m tcp --dport 80 -j ACCEPT
-A INPUT -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
-A INPUT -j DROP

-A OUTPUT -m conntrack -ctstate RELATED,ESTABLISHED -j ACCEPT
-A OUTPUT -j DROP
COMMIT
```

   (a) Alice's computer has the above firewall policy. Read it and use it to answer the following questions:

      i. If Eve (32.4.110.4) were to run a port scan on Alice using nmap what ports would she see as open? List the numbers of the open ports and explain your reasoning. [*2 marks*]

      ii. Imagine Alice were to try and open the Computer Security class website in a web browser. You use Wireshark and limit the collection to packets to those coming from Alice's computer and with a destination of ports 80 or 443. You can assume that no other computers are communicating with Alice at the time. Describe the traffic you would expect to see coming from Alice's computer and explain your reasoning. [*4 marks*]

      iii. There are several different types of firewalls that operate at different layers of the OSI network stack. Name the type of firewall shown above and explain how you know it is of that type. [*3 marks*]

      iv. Imagine that Eve decides she really does not like Alice having a website on port 80. She decides to launch a SYN Flood attack against Alice. Would this attack work against Alice? Explain your answer. [*3 marks*]

      v. The SYN Flood described in question 1(a)iv has some disadvantages for Eve. Name one of the problems with this attack and suggest a better approach Eve could use. [*3 marks*]

QUESTION CONTINUES ON NEXT PAGE

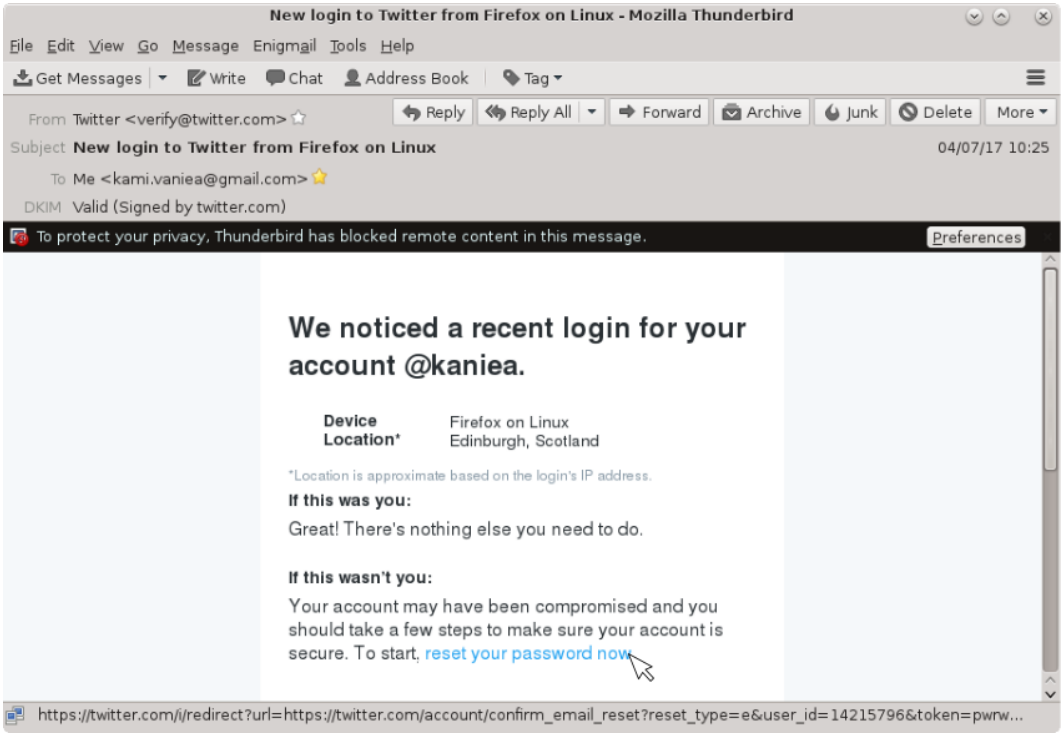

Figure 1: Screenshot of an email the instructor received.

(b) Is the email in Figure 1 legitimate? Explain your answer. [*3 marks*]

(c) Eve discovers a company website where she can log in using the username "admin" and the password "admin". She uses this information to log onto the website and download a large amount of personal customer data.

i. There are five controls in Cyber Essentials. Name the control that if followed would have prevented the above attack. [*1 mark*]

ii. Describe a different future attack that the system will be protected against, if the administrator correctly implements the control you described in Assume that the administrators correctly implement the control you described in question 1(c)i. [*3 marks*]

iii. The website administrator would like to add an invisible second authentication mechanisum to the website in addition to the password to further improve security. Describe how such an approach might be implemented on the site. [*3 marks*]

2. **Coin flipping**

Alice and Bob are responsible for marking the Computer Security exam, but none of them wants to do it. They decide to toss a coin to determine who will be marking the exam. One problem is that one of them is travelling, so they only communicate over the telephone. Alice proposes that she flip the coin and telephone Bob with the result. This proposal is of course not acceptable to Bob since he has no way of knowing whether Alice is telling the truth when she says that the coin landed tails.

In this question we will examine 5 different protocols for distributed coin tossing. Nor Alice nor Bob should be able to cheat and force the coin to show heads or tails. For each of these schemes, say if it is secure or not. If you think it is secure justify your answer. If you think it is insecure give the attack which allows one of the two parties to obtain heads with probability more than 50%.

In the following, we let p be a 2048-bit prime number, g a generator modulo p, and H a secure (in the usual sense) cryptographic hash function know to all parties.

(a) Alice randomly picks a such that $0 < a < p$ and sends to Bob $g^a \mod p$. Bob randomly picks b such that $0 < b < p$ and sends to Alice $g^b \mod p$. Both Alice and Bob can now compute $g^{ab} \mod p$. If $g^{ab} \mod p$ is even, then the coin flip is heads; otherwise, the coin flip is tails. [*5 marks*]

(b) Alice randomly picks a such that $0 < a < p$ and sends to Bob $g^a \mod p$. Bob randomly picks b such that $0 < b < p$ and sends b to Alice. Alice then sends a to Bob who checks that it corresponds to what Alice sent in her first message. If $H(a||b)$ is even, then the coin flip is heads; otherwise, the coin flip is tails. [*5 marks*]

(c) Alice randomly picks a such that $0 < a < 2^{128}$ and sends $H(a)$ to Bob. Then, Bob randomly picks b such that $0 < b < 2^{128}$ and sends $H(b)$ to Alice. Then, Alice reveals a to Bob and Bob checks that it matches what Alice previously sent. Bob also reveals b to Alice and Alice checks that it matches what Bob previously sent. If a and b have the same parity (ie. both parties picked an odd number, or both parties picked an even number) then the coin flip is heads. Otherwise, the coin flip is tails. [*5 marks*]

(d) Alice randomly picks a such that $0 < a < 2^{128}$ and sends $H(a)$ to Bob. Then, Bob randomly picks "even" or "odd" and sends that to Alice. Then, Alice reveals a to Bob who checks that it corresponds to what Alice sent earlier. If Bob did guess a's parity (ie. Bob said "even" and a is even, or Bob said "odd" and a is odd), then the coin toss is heads; otherwise, it is tails. [*5 marks*]

QUESTION CONTINUES ON NEXT PAGE

(e) Alice randomly picks a and k such that $0 < a, k < 2^{128}$. Alice computes $c = a \oplus k$ [the one-time-pad encryption of a, under key k] and sends c to Bob. Then, Bob randomly picks "even" or "odd" and sends that to Alice. Alice then reveals a and k to Bob who checks that it corresponds to what Alice previously sent. If Bob did guess a's parity (ie. Bob said "even" and a is even, or Bob said "odd" and a is odd), then the coin toss is heads; otherwise, it is tail. [*5 marks*]

3. **Memory management**

Initially, you will assume a 32-bit linux machine with the run-time stack growing downwards in memory (from high memory addresses to low ones). That is, the stack frame of the caller function is higher up in memory than the stack frame of the callee function. Consider the following code

```
int foo(char *buffer);

void bar(void);

void vuln() {
     char buffer[4];
     gets(buffer);
     if (foo(buffer))
        bar();
     return 0;
}
```

`gets()` reads characters from standard input until the newline character (`'\n'`) is read. The sequence of characters is stored in the parameter of `gets()`. If any characters are read, `gets()` appends the character `'\0'` to the end of the parameter.

(a) Depict the stack frame of the function `vuln()` just before the call to the `gets()` function. Explain why this code is vulnerable to a buffer overflow. [*5 marks*]

(b) The function `bar()` is located at address `0x00000462`. What string can be given to `foo()` to cause the function `vuln()` to unconditionally transfer control to the function `bar()`? [*5 marks*]

(c) In class we discussed several protection mechanisms against different buffer overflow attacks. Give one that would prevent the above attack. Explain how it works, and why it would thwart the above attack. [*5 marks*]

The goal now will be to understand how the order of the stack frames in memory is relevant to buffer overflows. So, in all the following questions we will assume a 32-bit x86 linux machine but with the stack growing upwards instead (from low memory addresses to higher ones). Consider the following code, in which `vuln()` is invoked with a string of any length supplied by the user. We can assume that `foo()` is safe, and that it simply returns straight away.

QUESTION CONTINUES ON NEXT PAGE

```
void vuln(char *input) {
     char buffer[256];
     strcpy(buffer, input);
     if (!strncmp(buffer, "GET ", 4))
        foo(buffer);
     return;
}
```

The C library function, `char *strcpy(char *dest, char *src)` copies the string pointed to by `src` to `dst`. The C library function, `int strcmp(char *src1, char *src2)` compares the two strings pointed to by `src1` and `src2`.

(a) Explain how you could exploit `vuln()` to execute arbitrary code. Depict the stack to illustrate what locations on the stack you need to corrupt for your attack, and what input string you would need. [*5 marks*]

(b) Where should you place the stack canary, when should it be written on the stack, and when should it be checked, to defend against buffer overflows that overwrite the return address? [*5 marks*]