Practice Problems

Computer Security

Principles and Practice

Fourth Edition

William Stallings

TABLE OF CONTENTS

CHAPTER 1 OVERVIEW

1.1 Which of the following activities might be considered a possible source of threat to a company's network, and why?

a. The daily courier service personnel who drop off and pick up packages.

b. Former employees who left the company because of downsizing.

c. An employee traveling on company business to another city.

d. The building management company where an organization has its offices has decided to install a fire sprinkler system.

CHAPTER 2 CRYPTOGRAPHIC TOOLS

2.1 Define the notion of collision resistance for hash functions.

2.2 Consider an application that requires an encryption and MAC algorithm be implemented on a processor with a small amount of non-volatile memory. The only cryptographic algorithm that the processor can compute is Triple-DES. But you have space for one 168-bit key. Describe how it is possible to both encrypt and MAC using only a single key. Justify your answer and state any assumptions you use.

2.3 We consider a banking application, where messages *m* of the form `fromAccount, toAccount, amount` are sent within the bank network, with the meaning that `amount` dollars should be transferred from `fromAccount` to `toAccount`. Each message consists of three blocks, with each block holding one of the three parameters. Messages are encrypted using AES in Counter mode, i.e.

$$K_j = \mathrm{E}(K, T_j)$$
$$C_j = M_j \oplus K_j$$

Each of the three parts of a message is sixteen characters, i.e. one block, so messages consist of three blocks.

a. The adversary has an account in the bank and can intercept and change messages. Imagine now that he knows the `toAccount` for a particular message $m = C_1C_2C_3$. Explain how he can modify the message so that the amount is transferred to his own account

b. Explain how the use of a MAC would prevent this attack.

c. Above, E(*K*, *M*) denotes using block cipher E with key *K* on message *M*. It is possible to define a cipher using similar ideas, but using a hash function instead. Describe how to do it, including how to decrypt.

CHAPTER 3 USER AUTHENTICATION

3.1 In the authentication protocol below, pw is A's password and J is a key derived from pw. Can an attacker that can eavesdrop messages (but not intercept or spoof messages) obtain pw by off-line password guessing? If you answer no, explain briefly. If you answer yes, describe the attack.

A (has pw)	B (has J)
send [conn] to B	
	generate random challenge R send [R]
compute J from pw compute X ← encrypt(R) with key J send [X] to B	
	compute Y ← decrypt(X) with key J if Y = R then A is authenticated

3.2 The chart below shows an authentication protocol, followed by data exchange, followed by disconnection. Only an initial part of the authentication protocol is shown; here, pw is A's password, J is a key derived from pw, and L is a high-quality key. Assume an attacker that can (1) eavesdrop messages and (2) intercept and spoof messages sent by A (but not those sent by B). Complete the authentication protocol (i.e., supply the part indicated by the "** …. * *") so that in spite of this attacker

- B authenticates A,
- this authentication is not vulnerable to off-line password guessing, and
- A and B establish a session key S (for encrypting data) such that after A and B disconnect and forget S, even if the attacker learns pw, the attacker cannot decrypt the data exchanged.

	A (has pw)	B (has J, L)
	send [conn] to B	X ← encrypt(L) with key J send [X]
	compute J from pw L' ← decrypt(X) with key J	
*		
*		
*		
*		
*		
*		
*		
*		
	←A and B exchange data →	
	←A and B disconnect →	

3.3 Consider an Intrusion Detection System with a False Positive Rate of 0.001 and a False Negative Rate of 0.09.

a. If there are 100,000,000 legitimate transactions (connections) a day, how many false alarms will occur?

b. If there are 1000 hacking attempts (connections) per day, how many true alarms will be given?

c. How many hacking attempts will go unnoticed?

CHAPTER 4 ACCESS CONTROL

4.1 Early Intel processors (e.g., the 8086) did not provide hardware support for dual-mode operation (i.e., support for a separate user mode and kernel mode). As a result, most of the systems implemented on these processors did not support multi-user operation. List and explain one potential problem associated with supporting multi-user operation without hardware support for dual-mode operation

Chapter 6 Malicious Software

6.1 Explain the following terms: Bot Net; Easter Egg; Logic Bomb.

6.2 Look at the following code snippet. You may assume that escape() argument is always non-null and points to a ’\0’- terminated string. What's wrong with this code (from a security point of view)?

```
/*Escapes all newlines in the input string, replacing them
    with"\n".*/
/* Requires: p != NULL; p is a valid ’\0’-terminated string */
void escape(char *p)
{
while (*p != ’\0’)
switch (*p)
{
case ’\n’:
memcpy(p+2, p+1, strlen(p));
*p++ = ’\\’; *p++ = ’n’;
break;
default:
p++;
}
}
```

CHAPTER 7 DENIAL OF SERVICE

7.1 The software company NikSoft is selling a new defense against DDoS attacks. Their software looks at the source IP address on all incoming packets, and if it finds any IP address that accounts for more than 1% of traffic over the last hour, it installs an entry in the router that blocks all packets from that address for the next 24 hours. Their marketing folks are claiming that this will stop all DDoS attacks cold in the water. Is this a good solution to the problem?

CHAPTER 8 INTRUSION DETECTION

8.1 Name some of the ways by which hackers compromise computers without code breaking.

8.2 What is a “Null session” problem?

8.3 How can an intrusion detection system actively respond to an attack?

8.4 Consider the following login protocol.

```
user knows password P
user knows Hash function H(.) and has a mobile calculator
user gives login name N to machine
machine generates random number R
machine gives R to user
user computes X := Hash(P) XOR Hash(R)
user gives X to machine
machine uses N to obtain P from password table
machine computes Y := Hash(P) XOR Hash(R)
if X=Y then machine allows login
```

a. Explain what is wrong with it and how can it be broken.
b. Show a simple way to strengthen this protocol against your attack.

CHAPTER 9 FIREWALLS

9.1 Can a packet filter block all incoming email containing the phrase “Make money fast”? If yes, show a packet filtering ruleset that provides this functionality; if no, explain why a (stateless) packet filter cannot do it.

9.2 List and explain three network threats that a firewall does not protect against.

9.3 You’ve been asked to protect a site with a firewall. There are no inbound services. The only outbound service is Web browsing, which of course requires some form of DNS name resolution. There is a lot of concern about people going to improper sites; there is also a desire to filter all web content to remove active content. Describe the best firewall configuration for this site. Justify the purpose of each element.

9.4 Which is generally safer between a firewall with a “default deny” policy or a firewall with a “default allow “policy? Why?

CHAPTER 10 BUFFER OVERFLOW

10.1 Consider the following C code snippet.

```
/* Escapes all newlines in the input string, replacing them with "\n". */
/* Requires: p != NULL; p is a valid ’\0’-terminated string */
void escape(char *p)
{
      while (*p != ’\0’)
      switch (*p)
      {
            case ’\n’:
                  memcpy(p+2, p+1, strlen(p));
                  *p++ = ’\\’; *p++ = ’n’;
                  break;
            default:
                  p++;
      }
}
```

You may assume that escape()'s argument is always non-null and points to a ’\0’-terminated string. What's wrong with this code?

10.2 Consider the following piece of code. What could cause a buffer overflow? Rewrite it to make it safe.

```
void main(int argc, char**argv)
{
    charbuf[256];
    sscanf(argv[0],"%s", buf);
}
```

10.3 Consider the following piece of code. What could cause a buffer overflow? Rewrite it to make it safe.

```
int main(int argc, char *argv[])
{
      char filename[MAXPATHLEN];
      if (argc==1)
            sprintf(filename, "/tmp/xxx%d", getpid());
      else
            sprintf(filename, "/tmp/%s", argv[1]);
      if ((fd = open(filename, O_RDWR|O_EXCL|O_CREAT, 0644) )< 0)
      {
            perror(filename);
            exit(1);
      }
}
```

10.4 Consider the following piece of code. What could cause a buffer overflow? Rewrite it to make it safe.

```
int check_authentication(char *password)
{
     int auth_flag = 0;
     char password_buffer[16];
     strcpy(password_buffer, password);
     if(strcmp(password_buffer, "brillig") == 0)
          auth_flag = 1;
     if(strcmp(password_buffer, "outgrabe") == 0)
          auth_flag = 1;
     return auth_flag;
}

int main(int argc, char *argv[])
{
     if(argc < 2)
     {
          printf("Usage: %s <password>\n", argv[0]);
          exit(0);
     }
     if(check_authentication(argv[1]))
     {
          printf("\n-=-=-=-=-=-=-=-=-=-=-=-=-=-\n");
          printf(" Access Granted.\n");
          printf("-=-=-=-=-=-=-=-=-=-=-=-=-=-\n");
     }
     else
     {
          printf("\nAccess Denied.\n");
     }
}
```

CHAPTER 11 OTHER SOFTWARE SECURITY ISSUES

11.1 a. Race conditions are a common problem in operating system protection mechanisms. An easy example involves a Time-of-check to Time of Use (TOCTOU) race based on Unix symbolic links, which contain a path that is resolved at access time. For example, the UNIX print utility “lpr” runs setuid root and has the ability to read any file on the system and send it to the printer – an operation that requires root privileges. To prevent a user from printing a file she doesn’t own, it checks the ownership of a file before printing it:

```
if (access(argv[1], R OK) != 0)
exit(1);
fd = open(argv[1], O RDONLY);
send_to_printer(fd)
```

Explain how a user can use two programs in combination with symbolic links to get lpr to print a file to which the user does not have permission to open.

b. To actually mount a practical attack, timing is critical. The attacker must make sure the second program is able to run exactly between the two system calls in the above code. How can the attacker guarantee this?

Chapter 20 Symmetric Encryption and Message Confidentiality

20.1 Assuming you can do 2^{20} encryptions per second and the key size is 40 bits, how long would a brute force attack take? Give a scenario where this would be practical and another where it wouldn't. What happens if you double the key size?

20.2 As you know, DES is insecure because of its short key length (56 bits). An improvement, proposed by Rivest, is DESX. DESX has key length 120 bits, seen as a pair $K = (k_1, k_2)$, where k_1 is 56 bits and k_2 is 64 bits. The encryption of a one-block message m is

DESX(K, m) = $k_2 \oplus$ DES(k_1, (m $\oplus$ k_2))

a. Explain how decryption is done.

b. Explain why the inner XOR is necessary, i.e. explain an attack against

DESX'(K, m) = $k_2 \oplus$ DES(k_1, m)

that is much better than brute force.

20.3 Briefly describe the Shift Rows and Byte Substitution layers of AES. Explain why we can apply them in either order with the same result.

20.4 Consider a sensor X that periodically sends a 64-octet measurement to a receiver Y. One day the administrator decides that X should encrypt the measurement data using DES in CBC mode. How many octets does X now send for each measurement? Explain your answer.

CHAPTER 21 PUBLIC-KEY CRYPTOGRAPHY AND MESSAGE AUTHENTICATION

21.1 Public-key algorithms are usually used for encrypting short messages. But if we need to encrypt a longer message we can split it into blocks, use RSA for each block and use a block cipher mode. Which of the two modes, CBC and Counter, would you recommend in such a situation?

21.2 Many hash functions are constructed from a simpler building block, called a compression function. Describe this general construction.

CHAPTER 22 INTERNET SECURITY PROTOCOLS AND STANDARDS

22.1 What is the way by which one can download normal email or web pages so that the content is hidden?

22.2 Explain the terms "proof of submission" and "non-repudiation" in an electronic mail system. Explain the importance of non-repudiation in an e-commerce system.

22.3 Will IPsec make firewalls obsolete?

22.4 An attacker is intent on disrupting the communication by inserting bogus packets into the communications. Discuss whether such an attack would succeed in systems protected by IPsec. Discuss whether such an attack would succeed in systems protected by SSL.

22.5 Some website designers decide that they don't want to use SSL. To prevent a user's password from being sniffed over the network, they arrive at a clever idea. Instead of sending the user's password over the Internet, they create some Javascript that will hash the user's password and send the hash instead.
a. Is this scheme secure?
b. Can this scheme be improved to make it more secure?

CHAPTER 23 INTERNET AUTHENTICATION APPLICATIONS

23.1 Consider a Web service that allows people to sign their web pages. The service does this by appending, hidden inside a special HTML tag at the bottom of an otherwise normal web page, the author's name, the date, and a digital signature (which contains the author's name and date signed by the author's RSA private key). The web page itself is unencrypted, but the signature can be validated by downloading a list of all registered users of the Web service and each user's public key) to retrieve the author's public key.

Explain why this gives a completely false sense of security, by outlining two different ways that you could make it appear that Linus Torvalds has posted a web page saying “Open source is for losers; I've decided to go work for SCO”. The definition of “different” is that each attack has a unique fix.

For each of the attacks you list, give a countermeasure that the author/viewer could take to protect themselves against that attack.

CHAPTER 24 WIRELESS NETWORK SECURITY

24.1 What do you call a fake hotspot used by hackers to prey on mobile workers, and how does it work?

24.2 Briefly describe three distinct types of denial-of-service (DOS) attacks that may be performed on IEEE802.11 wireless networks. Each of the three attacks should be at a different layer of the protocol stack.

CHAPTER 27 TRUSTED COMPUTING

27.1 State True/False and also give justification for each statement.

a. Access control matrices can represent anything that is represented by access control lists.

b. Consider data that is stored over time in a mandatory access control based system. The contents of files containing highly classified ("top secret") information are necessarily more trustworthy than material stored in files marked unclassified