**Câu 1. Nguyên tắc an toàn bảo mật**

1. Trong hầu hết các rạp chiếu phim, bạn sẽ phải đến quầy bán vé để mua vé cho suất chiếu của bộ phim muốn xem. Sau đó, bạn cần xuất trình vé này cho nhân viên soát vé ở trước cửa phòng chiếu. Hãy cho biết 2 nguyên tắc an toàn bảo mật nào đã được áp dụng? Với mỗi nguyên tắc, hãy giải thích bằng 1-2 câu để cho thấy rạp phim đã áp dụng nguyên tắc đó.
2. Khi kiểm tra vé của bạn, nhân viênát vé phải xé cuống vé. Tại sao chủ rạp phim yêu cầu anh ta phải hành động như vậy?
3. Rạp phim nâng cấp hệ thống bán vé với một chip RFID được gắn trong vé. Con chip này lưu trữ giá trị ID của vé. Tại cánh cửa phòng chiếu, một máy quét giao tiếp với con chip này để đọc giá trị ID của vé và gửi thông tin về máy chủ để kiểm tra. Nếu phát hiện vé giả, máy quét phát âm thanh báo động. Nguyên tắc an toàn bảo mật nào đã được áp dụng? Hãy giải thích bằng 1-2 câu để cho thấy rạp phim đã áp dụng nguyên tắc này.

**Câu 2. Các chế độ mật mã đối xứng**

1. Sử dụng chế độ mật mã nào sau đây có thể chống lại được tấn công CPA?(Chọn tất cả đáp án đúng)

|  |  |
| --- | --- |
|  | Chế độ mã ECB |
|  | Chế độ mã CBC sử dụng giá trị IV từ một bộ đếmz |
|  | Chế độ mã CBC sử dụng giá trị IV tạo ra từ thao tác truy cập ổ đĩa cứng |
|  | Chế độ mã CTR sử dụng giá trị IV từ một bộ đếm |
|  | Chế độ mã CTR sử dụng giá trị IV là kích thước của thông điệp |

1. Bob thiết kế chế độ mã new-CBC như sau:

* Khởi tạo giá trị IV có kích thước n bit theo ngẫu nhiên phân bố đều.
* Khối m1 có kích thước n bit được mã hóa: c1 = E(k, m1) Å IV
* Mỗi khối mi (với i > 1) có kích thước n bit được mã hóa: ci = E(k, mi) Å ci-1
* Giá trị IV được đính kèm với bản mã

Diagram, schematic

Description automatically generated

Hãy vẽ sơ đồ giải mã

1. Chế độ mã new-CBC có an toàn trước tấn công CPA không? Hãy chứng minh câu trả lời.

**Câu 3. Xác thực thông điệp quảng bá**

Giả sử server cần gửi thông điệp tới tất cả client C1, C2, .., Cn. Mỗi client khi nhận được thông điệp này cần xác minh được rằng thông điệp do chính server gửi. Giả sử rằng đã có một kênh an toàn được sử dụng để phân phối khóa giữa các bên.

a. Giả sử server và tất cả client chia sẻ một khóa đối xứng ngẫu nhiên cho từng phiên quảng bá. Server sử dụng khóa này để tạo mã MAC và gắn vào mỗi bản tin được gửi đi. Client có thể tin tưởng vào thông điệp nhận được là do server gửi và nội dung thông điệp toàn vẹn hay không? Giải thích.

b. Giả sử server chia sẻ một khóa ngẫu nhiên với mỗi client cho từng phiên quảng bá. Server sử dụng khóa này để tạo mã MAC và gắn vào mỗi bản tin được gửi đi. Client có thể tin tưởng vào thông điệp nhận được là do server gửi hay không? Giải thích.

c. Giả sử client nhận được một thông điệp từ server có chứa phần mềm độc hại. Client có thể khởi kiện server hay không? Giải thích.

**Câu 4. Chữ ký số dựa trên RSA**

Alice sử dụng chữ ký số RSA xây dựng theo 2 sơ đồ sau:

**Sơ đồ đơn giản:** Chữ ký số sig = E(KR, m)

**Sơ đồ tiêu chuẩn:** Chữ ký số sig = E(KR, H (m))

Trong đó:

KR = (d, n) là khóa cá nhân của Alice được giữ an toàn bí mật

KU = (e, n) là khóa công khai của Alice đã được công bố trên website tin cậy

H(m): Hàm băm mật mã an toàn

Bản tin được Alice gửi cho Bob là (m, sig).

a. Nếu Alice sử dụng chữ ký số đơn giản, bằng cách nào Bob có thể kiểm tra tính xác thực của bản tin.

b. David muốn tấn công vào tính xác thực của bản tin dùng chữ ký số đơn giản. Bằng cách nào David có thể tạo ra bản tin (m, sig) đánh lừa được Bob là bản tin này do Alice gửi.

c. Cách thức tấn công trên của David có tác dụng không nếu Alice sử dụng chữ ký số tiêu chuẩn. Hãy giải thích.

**Câu 6.** **Xác thực danh tính bằng mật khẩu**

Thông tin tài khoản người dùng được lưu trong cơ sở dữ liệu của server gồm định danh (userID) và giá trị băm của mật khẩu Hash(password). Giao thức xác thực giữa server và client như sau:

(1) Client à Server: userID || password

(2) Server băm mật khẩu nhận được từ yêu cầu của client và so sánh với giá trị đã lưu trong CSDL. Nếu trùng khớp thì xác thực thành công

a. Hãy mô tả 2 điểm yếu của hệ thống xác thực trên.

b. Mô tả giải pháp để sữa chữa các điểm yếu đã chỉ ra.

**Câu 6. Phân quyền (1 điểm)**

Một hệ thống kiểm soát truy cập theo mô hình Bibaphân loại mức độ quan trọng cho tài liệu và người dùng như sau:

|  |  |  |
| --- | --- | --- |
| Mức độ bí mật | Người dùng | Tài liệu |
| Crucial | An | Hồ sơ dự án |
| Very Important | Bình, Cường | Hồ sơ nhân viên |
| Important | Duy, Giang | Báo cáo tài chính  Báo cáo kinh doanh |

Biểu diễn quyền đọc(read) và ghi(write) của người dùng trên hệ thống này bằng ma trận điều khiển truy cập

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Hồ sơ dự án | Hồ sơ nhân viên | Báo cáo tài chính | Báo cáo kinh doanh |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

**Câu 7. Lỗ hổng phần mềm**

Một số hàm chương trình được viết như sau:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41 | typedef struct product{  char name[256];  short quantity;  } product;  /\* Order products of the store  Return total quantity of products \*/  int order(product list[])  {  char tmp\_name[256];  int n\_products = 0;  while(1)  {  printf(“Product: ”);  gets\_s(tmp\_name, 255);  if(strlen(tmp\_name == 0)) break;  strcpy(list[n\_products].name, tmp\_name);  printf(“Quantity: ”);  scanf(“%d”, &list[n\_products].quantity);  n\_products++;  }  return n\_products;  }  /\* Send shopping cart to server  Return number of transferred bytes, -1 if error \*/  int sendCart(int socket, product list[], int n)  {  char buff[4096], str\_quantity;  int ret, len;  for (int i = 0; i < n, i++){  len = strlen(list[i].name);  strncat(buff, list[i].name, len);  strncat(buff, “ ”, 1);  sprintf(str\_quantity, “%d\n”, list[i].quantity);  len = strlen(str\_quantity);  strncat(buff, str\_quantity, len);  }  ret = send(socket, buff, strlen(buff), 0);  return ret;  } |

1. Hãy chỉ ra 2 lỗ hổng phần mềm trong đoạn chương trình trên dẫn đến kẻ tấn công có thể khai thác để chèn mã độc thực thi.
2. Viết lại mã nguồn cần thiết để sửa các lỗ hổng trên

**Phụ lục: Một số hàm C/C++**

char \*gets\_s(char \*buffer, size\_t n)

Gets a line from the **stdin** stream.

**Parameters**

*buffer:* Storage location for input string.

*n:* The size of the buffer.

**Return Value:** Returns *buffer* if successful. A **NULL** pointer indicates an error.

int sprintf(char \*buffer, const char \*format, [arguments] ...);

Write formatted data to a string.

### Parameters

buffer: Storage location for output

format: Format-control string

arguments: Optional arguments

**Return Value:** The number of bytes stored in buffer, not counting the terminating null character.

char \*strncat(char \*strDest, const char \*strSource, size\_t count);

The **strncat** function concatenates two string.

### Parameters

strDest: Null-terminated destination string.

strSource: Null-terminated source string.

count: Number of characters to append.

## Return Value: Returns a pointer to the destination string. No return value indicates an error.

int send(int *socket*, char \**buffer*, size\_t *length*, int *flags*);

The **send** function sends data on a connected socket.

### Parameters

*socket:* Specifies the socket file descriptor.

*buffer:* Points to the buffer containing the message to send.

*length:* Specifies the length of the message in bytes.

*flags:* Specifies the type of message transmission.

## Return Value

Upon successful completion, *send*() shall return the number of bytes sent. Otherwise, -1 shall be returned and *errno* set to indicate the error.