KTH Datavetenskap
och kommunikation

DD2395 Computer Security
Exam 2016-01-16, 9.00 –12.00

This is a closed-book exam, no material (books, articles, laptops, phones, or any other electronic devices, etc.) permitted. Please answer in English if you can, only answer in Swedish if you must. Each question is annotated with the corresponding level (A, C, E); there are 8 E questions, 4 C questions and 2 A questions. Grading is according to the following preliminary scheme:

- Fx: correctly answer at least 5 E questions
- E: 6 E questions
- D: 6 E questions and 2 C questions
- C: 6 E questions and 3 C questions
- B: 6 E questions, 3 C questions and 1 A question
- A: 7 E questions, 3 C questions, 2 A questions
  or 6 E questions, 4 C questions, 2 A questions
  and 2 on 3 Bonus points from the labs

This is a preliminary draft of the exam solutions.

1 CIA. [E]

Provide two examples of confidentiality loss: one with low impact and one with high impact

Solution (i) One of my colleagues discovers my choice in the department anonymous survey used to select the fika time. (ii) The evil-party obtain access to my choices in a e-voting.

2 Cryptography.

2a **[E]** Describe the mathematical to attack the RSA algorithm. That is, which mathematical problem must be solved by an attacker to break RSA without brute-forcing all possible keys.

Solution Factorization of a large number into two prime numbers.

2b **[C]** What is Electronic Codebook Mode (ECB)? Invent and describe a scenario where it is not secure to use ECB. Explain why it is not secure to use ECB in such scenario. Which mode of operation can you use (securely) in your example? Explain why.

Solution ECB is a block mode of operation. Assume E be an encryption function and D be the corresponding decryption function, both being able to process blocks of x bits. A message $M = B_1, \ldots, B_n$ (consisting of n blocks of x-bits) is encrypted as $E(B_1), \ldots, E(B_n) = C_1, \ldots, C_n$. The cyphertex is the decrypted as $D(C_1), \ldots, D(C_n) = M$.

This mode of operation has two weakness: (i) if a block is repeated in the message, then the corresponding cyphertex is also repeated and (ii) it is sufficient to reorder the cyphertexes to obtain the encryption of a message whose blocks are reordered.

I exploit the second vulnerability. Two banks b_1 and b_2 exchange transactions, by b_1 sending the following message to b_2. The first block $B_1 = sender_1, receiver_1, \ldots, sender_k, receiver_k$ contains the list transaction headers ($sender_i$ wants to send money to $receiver_i$). Then, the block $B_{i+1} = amount_i$ contains the amount of money that each transaction should transfer. (p.s. this is a fictional scenario and I assume that the resulting blocks have all the same size).

A man-in-the middle can take the message $E(B_1), E(B_2), E(B_3) \ldots, E(B_{k+1})$, reorder the cyphertex as $E(B_1), E(B_3), E(B_2) \ldots, E(B_{k+1})$ and then causing the transfer of the $amount_2$ (instead of $amount_1$) from $sender_1$ to $receiver_1$.

The two banks can use Cipher Clock Chaining Mode. In this case, the blocks can not be reordered, without altering the value that is decrypted be the receiver. The bank b_1 will deliver $C_1 = E(IV \oplus B_1), C_2 = E(C_1 \oplus B_2), C_3 = E(C_2 \oplus B_3) \ldots$. If the attacker reorders the message as $C_1, C_3, C_2, \ldots$, the receiver will decrypt
(1) $V_1 = D(C_1) \oplus IV, V_2 = D(C_3) \oplus C_1, \ldots$
(2) $V_1 = D(E(IV \oplus B_1)) \oplus IV, V_2 = D(E(C_2 \oplus B_3)) \oplus C_1, \ldots$
(3) $V_1 = IV \oplus B_1 \oplus IV, V_2 = C_2 \oplus B_3 \oplus C_1, \ldots$
(4) $V_1 = B_1, V_2 = C_2 \oplus B_3 \oplus C_1, \ldots$
Namely, V_2 will result in a non-compliant message, enabling b_2 to discover that the blocks have been reordered.

2c **[E]** Define the requirements for a secure Hash Function

Solution Let H be the hash function

1. H can be applied to input of arbitrary size
2. H produces outputs of fixed size
3. $H(x)$ is relatively easy to compute
4. one-way: given h, it is difficult to find x such that $H(x) = h$
5. weak collision resistant: given x, it is difficult to find $y \neq x$ such that $H(x) = H(y)$
6. strong collision resistant: it is difficult to find x and $y \neq x$ such that $H(x) = H(y)$

An answer is considered correct if it presents at least requirements 4, 5 and 6.

2d **[A]** Let $M = B_1, \ldots, B_n$ be a message consisting of n blocks of 1 MB each. Let H be the function defined as $H(M) = SHA(B_1\ xor\ \ldots\ xor\ B_n)$, where SHA is the Secure Hash Algorithm and xor is bitwise xor. Is H a secure Hash Function? For each requirement, motivate if it holds or disprove it.

If the function is not a secure hash function, invent a scenario where using H can result in a high impact integrity loss. Include in your description how the function is used and the attack that can be performed by a treat agent.

If the function is a secure hash function, explain the benefit of using H instead of SHA.

Solution It is not a secure hash function:

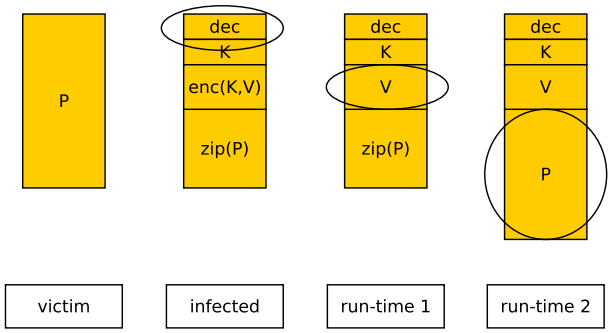
- requirement 4 hold: to find a M such that $H(M) = h$ requires to solve the problem of finding B such that $SHA(B) = h$.
- requirement 5 doesn't hold: let $x = B_1, B_2$, then $H(B_2, B_1) = H(x)$.
- requirement 6 doesn't hold: $H(0, 1) = H(1, 0)$.

3 Firewall. [E]

Can I use a packet filter firewall to block the HTTP traffic that carries files bigger than 5MB while allowing the other requests/responses? If yes, briefly explain how I should configure the firewall. If not, explain which kind of firewall can be used and why. (p.s. an HTTP response is transmitted via TCP using usually several packets.)

Solution No. Since the HTTP response is delivered via several TCP packets, a stateless firewall can not keep track of either TCP connections or the single HTTP responses. A simple solution is to use an application proxy firewall; e.g. an HTTP proxy that sits between the clients and the servers and has specific understanding of the HTTP protocol.

4 **Malware [E]** Describe (also include a figure) the behavior of an Encrypted virus.



Solution An Encrypted virus is a virus that encrypts its payload to make difficult to a anti-virus engine to discover its presence. The behavior of an Encrypted virus is the following

- Infection of a victim executable
  1. the virus generates a new encryption key (K)
  2. the virus encrypts its payload (V)
  3. the virus copies the cyphertext (E(K,V)) into the victim (P)
  4. the virus copies a small bootstrap code (dec) and the decryption key
- Execution of the virus
  1. the bootstrap code (dec) is executed
  2. it decrypts in memory the payload (E(K,V))
  3. it executes (jumps to) the decrypted payload (V)
  4. when the payload terminates, it executes the victim (P)

5 **DOS [E]** Give an example of a denial of service attack (include the mechanism used by the attacker, the resources consumed and possible countermeasure).

Solution An example is the SYN-spoofing attack. The attacker A targets a victim V. The attacker uses the IPs of hosts that are not available on the network (or that are not able answer to IP packets) and generates a large volume of SYN packets, with destination V and source the spoofed IPs. These packets request V to open a TCP connection. For each SYN packet, V adds the request to a local table (needed to keep track of the TCP connection variables, like sequence numbers) and deliver a SYN-ACK packet to the spoofed source. Since the spoofed source is non existent (or not able to answer to IP packets), V does not receive a reset packet.
If the attacker is fast enough, it can exhaust the memory allocated for the TCP connection table of the victim, making impossible for V to receive licit requests of opening new TCP connection. One of the possible countermeasure is to randomly drop pending TCP connections from the table whenever the number of entries in the table exceed a predefined threshold.

6 **Multilevel security [E]** Describe the Chinese Wall Model (including the elements of the models and the security rules). Which top level security property is guaranteed by this security model?

Solution The goal of the Chinese Wall Model is to ensure that information can not flow between two corporations being in conflict of interest.
The "objects" (i.e. unit of information) are grouped into "datasets". Each dataset represents a corporation. Moreover, each dataset belongs to one or more conflict of interest class (CI).
There are two main rules to respect:

- (ss-rule) a subject can read an object if
  - the object is in a dataset that has been already accessed by the subject or
  - the object belongs to a CI that has never been accessed by the subject
- (*-rule) a subject can write an object O if
  - the subject can read the object and
  - the subject can not read objects outside the dataset of O.

7 Network security The server S provides a news aggregation service. Users can log-in using a password, register for one or more topics and receive updates whenever the server discover a new website related to the subscribed topics. Hereafter we assume that each host has an IP that is statically assigned and that never changes. The clients and the server communicate using the following protocol over UDP:

- a user log-in by sending to S a UDP datagram containing $E_{PU}(user, pwd)$, where E represents the encryption using an asymmetric algorithm, PU is the public key of S, usr and pwd are the login and password of the user
- the server can accept the authentication (by replying with a UDP datagram containing the payload OK) or reject it (by replying with a UDP datagram containing the payload $fail$)
- a user can subscribe to a topic by sending to S a UDP datagram containing $REGISTER; topic$
- the server accepts the registration only if the IP of the sender has previously performed an authentication. S stores this information in a local table.
- whenever the server discovers a new website related to a subscribed topics, the server accesses the local table, fetches the corresponding subscribers and delivers to each of them a UDP datagram $NEW; topic; url$. To reduce the amount of information received by the user, the server delivers the notification only after checking that the published website does not contain spam.

The system should guarantee that users receive only non-spam websites and only for the topic they registered.

7a **[C]** Describe a security threat of the system that can be used by an attacker to cause to a denial of service of a host on the network. Include in your description (1) which is the victim, (2) which resource is consumed, (3) how the attacker can exploit the vulnerability and (4) possible assumptions that are needed to make the attack possible.

Describe an additional security threat (not necessarily a DOS) of the system.

Solution The system can be used to attack an host V, even if V is neither a user nor the server. We assume that the server S has high speed internet connection, while the victim has a low speed one. We also assume that the attacker has a internet connection that is better that the one of the victim. Finally, We assume that the size of the payload $fail$ is at least the same of the payload $E_{PU}(user, pwd)$.

The attacker generates high volume of authentication packets $E_{PU}(user, pwd)$. The attacker can generate random *user* and *pwd* and uses the IP address of V as source of the UDP packet. The server, will answer with a high volume of $fail$ packets, all targeting the victim IP, thus saturating his bandwidth.

An additional threat depends on the fact that the messages $NEW; topic; url$ are plaintexts over UDP. An attacker (assuming that the attacker knows to which topics a user is interested in) can spoof the server IP and generate similar packets, thus persuading a user to visit forged urls (spam).

An additional reflection DOS is the following. We assume that the attacker knows the IP of a previously authenticated user. The attacker deliver the messages $REGISTER; topic$ to the server, using the IP of V as source of the UDP packets. The attacker repeat the procedure for a huge number of different topics. Now, for each packet delivered by the attacker, the server can potentially deliver a large number of packets to the victim: one for each website that is related to the topic.

7b **[C]** Describe some countermeasures that reduces the two security threats. (e.g. changing the protocols (including how), adding firewalls (including their type and configuration), adding IDS). Describe why these countermeasure are effective.

Solution A simple solution to the first problem is to change the behavior of the server. The server only accepts up to one request per second from an IP, and every further request is dropped. This automatically limit the amount of traffic that the server can deliver to a spoofed IP in response to a request (there can still be a DOS by registering an IP to a high volume of news).

A solution to the second problem can be deliver the packets $NEW; topic; url$ signed with the private key of the server. The server delivers $NEW; topic; url; E_{PR}(H(NEW; topic; url))$, where H is a secure hash function and PR is the server private key. An attacker can not forge a similar authentication code without knowing the server private key (the attacker can still send repetition of messages generated by the server).

This is a simple and minimal solution that only targets the vulnerability of the previous question. In general, you should establish a session key (during the authentication) and use that for the subsequent messages. Even better, you should not use a custom protocol, but use one whose security properties have been deeply investigated (e.g. TLS).

8 Buffer overflow

A system uses the following protocol (over TCP). After establishing the connection, the client sends a first message N, consisting of four bytes. N represents the number of bytes that the client will deliver subsequently to the server. Then, the client sends several messages to the server, each one consisting of 256 bytes. Finally, it sends a message consisting of only one byte: the NULL character. After receiving this character the server performs some activities using the received data.

The following code is a fragment of the server source code (assume that the programming language is C, `sockfd` is the previously established socket and `read(s,p,n)` is a function that copies n bytes from the socket s into the pointer p).

```
void function(int sockfd) {
  unsigned int n;
  read(sockfd,&n,4);
  char data[n];
  unsigned int i = 0;
  while (1) {
    read(sockfd,data+(i*256),256);
    if (data[i*256] == '\0')
      break;
    i++;
  }
  // ...
}

void nuke_world() {
  // ...
}
```

8a **[E]** Identify (and explain the presence) of buffer overflows.

Solution line `read(sockfd,data+(i*256),256)` can cause a buffer overflow. The problem is that the server blindly trust the information delivered by the client. The client can send 257 as first messages and then do not obey to this constraint, by delivering several UDP datagrams of 256 bytes. Moreover, the attacker can not deliver the null terminator.

8b **[C]** Fix the code.

Solution A simple solution is to change the line `while (1) {` into `while (i < n/256) {`. Additionally, the allocation should take into account the size of each read, so the line `char data[n];` should be changed into `char data[ceil(n/256)*256];`

8c **[A]** Explain how an attacker can force the server to invoke the function `nuke_world` (i.e. a function left in the code but never used). Include a description of the message that can be sent by an attacker to exploit the vulnerability and describe the additional information about the server's code/binary/state that you need to build such message.

Solution The buffer `data` is allocated into the stack of the victim. In the stack, above this variable, the compiler places the other local variable `n`, the previous stack frame pointer and the return address of the function, the memory address to which `function` has to return after termination and the parameter `sockfd`. If `read(sockfd,data+(i*256),256)` overflows the buffer `data`, it can also override the return pointer, forcing `function` to jump to an arbitrary address in memory after its termination.

If the attacker sends a first message containing the value 257, the server allocates the buffer `data` to be 257 bytes. Let assume that the stack, while executing `function`, is

```
data       data+257  data+261   data+265
  |           |         |           |
  +-----------------------------------+
  |           |n       |frame_ptr  |return |
  +-----------------------------------+
```

and that the address of `nuke_world` is `0x11223344`.

The attacker can send three UDP datagrams. The first datagram consists of 256 dummy bytes (which fill the buffer `data`). The first nine bytes of the second datagram are also dummy bytes and override the last byte of `data`, the local variable n and the frame pointer. The bytes from 10th to 13th of the second datagram are `0x11223344`, which is written the return address. Finally, the third datagram contains just the null terminator.