Exam 2018, questions

Computer Security and Forensics (University of Sheffield)

Data Provided:
None

DEPARTMENT OF COMPUTER SCIENCE Spring Semester 2017-2018

COMPUTER SECURITY AND FORENSICS 2.0 hours

Answer ALL FOUR QUESTIONS. Write clearly in the sense of logic, language, and readability. The clarity of your arguments and explanations affects your grade.

Questions 1 and 2 carry a weight of 20%, questions 3 and 4 carry a weight of 30%. Figures in square brackets indicate the percentage of available marks allocated to each part of a question.

THIS PAGE IS BLANK.

1. Security Fundamentals & Access Control

a) Briefly explain the concepts *identification*, *authentication*, and *authorisation*. [20%]

b) For each of the following multi-factor authentication systems, explain briefly if it is a good or bad multi-factor authentication system:

(i) A *fingerprint* and *voice recognition*. [5%]

(ii) A *GPS signal* and a *password*. [5%]

c) Instead of using a login based on username and password, many modern websites allow users to log-in using services such as Google, Facebook, or Github. This mechanism is called *single sign-on*.
Briefly explain one advantage and one disadvantage of a single sign-on solution. [10%]

d) Recall the Access Control Matrix Model discussed in the lecture and consider the following configuration:

- The set of subjects S is defined as follows: $S = \{\text{Alice}, \text{Bob}, \text{Charlie}, \text{Eve}\}$
- The set of objects O is defined as follows: $O = \{\text{File1}, \text{File2}, \text{File3}, \text{File4}\}$
- The set of actions A is defined as follows: $A = \{\text{read}, \text{write}, \text{execute}, \text{append}\}$
- The informal access control policy is defined as follows:
  - *Alice* is allowed to *read* all files and she is allowed to *append* to File3.
  - *Bob* can *read* and *write* File4 and *append* to File2.
  - *Charlie* is allowed to *write* to the files File1 and File2 and he is allowed to *read* File1.
  - File2 can be *executed* by all subjects.
  - File1 can be *read* by Eve.

(i) Specify the Access Control Matrix for this scenario. [20%]

(ii) Briefly explain a scenario where two subjects that collaborate can violate the integrity or the confidentially of a file that is not possible without the two subjects working together. [20%]

(iii) Briefly explain if Access Control Matrix Models scale well (i.e., assume a scenario with thousands of subjects and/or objects). Compare them, in this aspect, with role-based access control (RBAC) [20%]

2. Cryptography & PKIs

a) (i) Briefly explain the concept of revocation lists.
[**Hint:** What information is stored in a revocation list and who maintains certification lists?] [10%]

(ii) Briefly explain a problem of revocation lists. [10%]

b) (i) Consider a Cesar Cipher, as discussed in the lecture, with the key 5. Recall that we encode the letters by their position in the alphabet (e.g., the letter "a" is represented by the number 1, spaces are not encoded).
Encrypt the text

```
fight for your privacy
```

[15%]

(ii) Does encrypting a text twice using Caesar Cipher improve the security? Briefly explain your answer. [15%]

c) Consider the RSA crypto scheme with the following configuration:

- Alice's public key is $(n_a, e_a) = (33, 7)$, her private key is $d_a = 3$. Alice uses this key for both signing and encryption/decryption.
- Bob's public key is $(n_b, e_b) = (65, 7)$, his private key is $d_b = 7$. Bob uses this key for both signing and encryption/decryption.
- Eve's public key is $(n_b, e_b) = (77, 13)$, her private key is $d_b = 37$. Eve uses this key for both signing and encryption/decryption.

Recall that we encode the letters by their position in the alphabet (e.g., the letter "a" is represented by the number 1, spaces are not encoded).

Alice sends to Bob the following *encrypted* message:

$$9, 1, 52, 38, 1, 45, 60$$

(i) Explain how the attacker Eve can obtain the clear text of this cipher?
[**Hint:** Recall that their keys are used for both signing and encryption/decryption.] [15%]

(ii) Demonstrate the attack by computing the plain text of the message that Alice sends to Bob. Briefly explain the necessary steps.
[**Hint:** If you did not answer 2(c)(i), show instead how Bob can obtain the plain text. Briefly explain the necessary steps.] [15%]

d) Can symmetric cryptography be used for implementing digital signatures? Briefly explain your answer. [20%]

3. **Security Protocols**

a) Consider the following protocol:

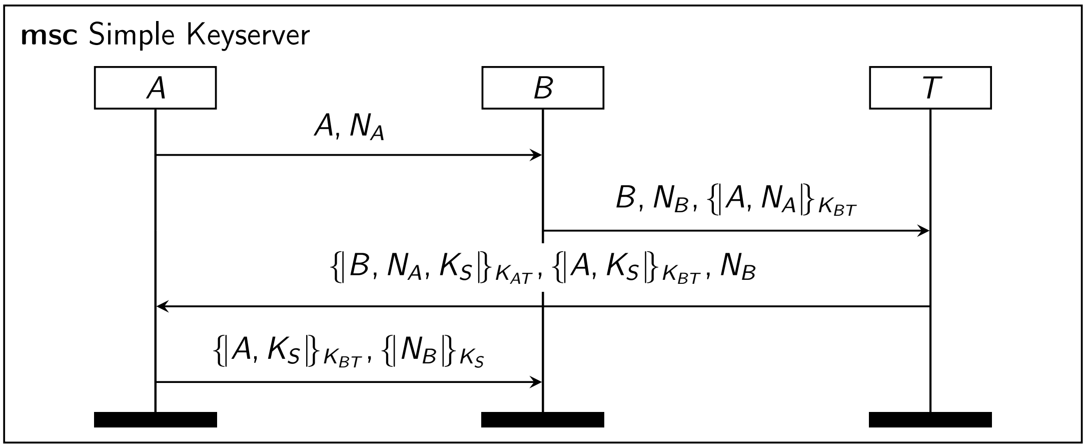
$A \longrightarrow B: \ \{N_A, A\}_{\mathsf{pk}(B)}$
$B \longrightarrow A: \ \{N_A, N_B, A\}_{\mathsf{pk}(A)}$
$A \longrightarrow B: \ \{N_B\}_{\mathsf{pk}(B)}$

Briefly explain if this protocol is secure or not.

- If it is secure, give an informal justification why it is secure.
- If an attack is possible, explain this attack (e.g., by drawing a message sequence chart), i.e., how does it work and why it is possible.

[20%]

b) Consider the following protocol in which A and B use a trusted third party T to perform mutual authentication and establish a session key K_S . Assume that initially A and T share the symmetric key K_{AT} and B and T share the symmetric key K_{BT} . A and B generate Nonces N_A and N_B, respectively. There are four steps in the protocol, as illustrated in the following message sequence chart:


msc Simple Keyserver
A
B
T
A, N_A
$B, N_B, \{\|A, N_A\|\}_{K_{BT}}$
$\{\|B, N_A, K_S\|\}_{K_{AT}}, \{\|A, K_S\|\}_{K_{BT}}, N_B$
$\{\|A, K_S\|\}_{K_{BT}}, \{\|N_B\|\}_{K_S}$

Consider the message in step 4 (from A to B). Briefly explain why A encrypts the Nonce N_B with K_S. [15%]

QUESTION CONTINUED ON THE NEXT PAGE

c) The Dolev-Yao closure $\mathcal{DY}(M)$ consists of *seven* deductive rules. Three of them are:

$$\frac{}{m \in \mathcal{DY}(M)}\text{ Axiom } (m \in M) \qquad \frac{s \in \mathcal{DY}(M)}{t \in \mathcal{DY}(M)}\text{ Algebra } (s \approx t)$$

$$\frac{\langle m_1, m_2 \rangle \in \mathcal{DY}(M)}{m_i \in \mathcal{DY}(M)}\text{ Proj}_i$$

(i) Define the missing *four* rules of the Dolev-Yao closure *formally* and give a brief informal description of what action of the intruder they formalise.
[**Hint:** if you cannot describe the rules formally, explain informally the different actions a Dolev-Yao intruder can use for inferring new messages from an initial set of messages (this might need two or three short sentences per rule).] [20%]

(ii) Consider the following intruder knowledge:

$$M = \big\{ \{\!|m|\!\}_{g(n_1)}, Axiom\{n_1\}_{\mathrm{inv}(\mathrm{pk}(a))}, \{n_2\}_{\mathrm{pk}(i)}, \mathrm{pk}(a), \mathrm{pk}(i),$$
$$\mathrm{inv}(\mathrm{pk}(i)), \{\!|secret|\!\}_{\mathrm{pk}(b)}, \{\{\!|secret|\!\}_m\}_{\mathrm{inv}(\mathrm{pk}(b))} \big\}$$

where g is a public function (i.e., $g \in \Sigma_P$).
Prove formally that the intruder can learn the message "*secret*".
[**Hint:** if you cannot recall the formal proof notation, give a detailed informal argument.] [25%]

d) Consider the following two key exchange protocols:

- Protocol 1 (using regular session keys):

$$\begin{array}{lll} B \longrightarrow A: & B, N_B \\ A \longrightarrow S: & A, B, N_A, N_B \\ S \longrightarrow A: & \{\!|K_{AB}, B, N_A|\!\}_{\mathrm{sk}(A,S)}, \{\!|K_{AB}, A, N_B|\!\}_{\mathrm{sk}(B,S)} \\ A \longrightarrow B: & \{\!|K_{AB}, A, N_B|\!\}_{\mathrm{sk}(B,S)}\mathrm{sk}(B,S) \end{array}$$

- Protocol 2 (using Diffie-Hellman-based session keys):

$$\begin{array}{lll} B \longrightarrow A: & B, A, \{\!|B, A, \exp(g, Y)|\!\}_{\mathrm{sk}(B,S)} \\ A \longrightarrow S: & \{\!|A, B, \exp(g, X)|\!\}_{\mathrm{sk}(A,S)}, \{\!|B, A, \exp(g, Y)|\!\}_{\mathrm{sk}(B,S)} \\ S \longrightarrow A: & \{\!|A, B, \exp(g, X), \exp(g, Y)|\!\}_{\mathrm{sk}(A,S)}, \{\!|B, A, \exp(g, Y), \exp(g, X)|\!\}_{\mathrm{sk}(B,S)} \\ A \longrightarrow B: & \{\!|B, A, \exp(g, Y), \exp(g, X)|\!\}_{\mathrm{sk}(B,S)} \end{array}$$

Explain briefly which of the two protocols is more secure?
[**Hint:** consider an attacker that has control over the server S, i.e., who can learn all information that is transmitted as well as all information the server knows or creates.] [20%]

4. **Application Security**

a) Consider a web application that uses the following client-side protection to prevent users from creating new "admin" accounts:

```
<select name="user[role]" id="user_role">

<option disabled="disabled" value="admin">admin</option>
<option disabled="disabled" value="lecturer">lecturer</option>
<option value="student">student</option>
</select>
```

(i) Explain briefly how an attacker can circumvent this check and create a new admin account. [10%]

(ii) Explain briefly how a programmer can prevent this type of attack. [10%]

b) Consider the following vulnerability in OpenSSL (CVE-2014-0160):

> The TLS and DTLS implementations in OpenSSL do not properly handle Heartbeat Extension packets, which allows remote attackers to obtain sensitive information from process memory via crafted packets that trigger a buffer over-read.
> A successful attack requires only sending a specially crafted message to a web server running OpenSSL. The attacker constructs a malformed "heartbeat request" with a large field length and small payload size. The vulnerable server does not validate that the length of the payload against the provided field length and will return up to 64 kB of server memory to the attacker. It is likely that this memory was previously utilized by OpenSSL. Data returned may contain sensitive information such as encryption keys or user names and passwords that could be used by the attacker to launch further attacks.

What CVSS v2 Base Vector (AV:?/AC:?/Au:?/C:?/I:?/A:?) would you assign to this vulnerability? Explain your decision for each component of the CVSS v2 Base Vector. In your answer, you will need to replace the '?' in the CVSS v2 Base Vector. [25%]

c) Consider the following Java program

```
String mname = request.getParameter("month");
String uid = session.getCurrentUserId();
PreparedStatement pstmt = conn.prepareStatement
             ("SELECT username, startdate, transaction, amount
               FROM transaction_history
               WHERE user=" + uid + " AND month=" + mname");
pstmt.execute();
pstmt.close();
```

(i) What vulnerability does this program have? Briefly explain the vulnerability in general and, in particular, why this program is vulnerable. [15%]

(ii) Rewrite this program to fix the vulnerability. [15%]

QUESTION CONTINUED ON THE NEXT PAGE

d) Consider the following Java program:

```
 1  String name      = request.getParameter("name");
 2  String uid       = session.getCurrentUserId();
 3  String secret    = "Cee0phoo";
 4  String output    = uid;
 5  PrintWriter out = resp.getWriter();
 6
 7  if(nodesLength > 0){
 8    output = name;
 9  }
10  if(nodesLength < 0){
11    output = output + name;
12  }
13  for(int i=0; i < nodesLength; i++){
14    output = "Hello Universe";
15  }
16  if(name.equals(secret){
17    out.println(""+Files.readAllLines(Paths.get("/etc/passwd"),
18                                      StandardCharsets.UTF_8));
19  }
20  out.println(output); // prints on a web page
```

The variable nodesLength is of type int and can take any integer value.

A static analysis tools generates the following four findings (potential vulnerabilities):

- Finding 1:
  - Line 2: `String uid = session.getCurrentUserId()`
  - Line 4: `String output = uid`
  - Line 20: `out.println(output)`
- Finding 2:
  - Line 1: `String name = request.getParameter("name")`
  - Line 8: `output = name`
  - Line 20: `out.println(output)`
- Finding 3:
  - Line 1: `String name = request.getParameter("name")`
  - Line 2: `String uid = session.getCurrentUserId()`
  - Line 4: `String output = uid`
  - Line 11: `output = output + name`
  - Line 20: `out.println(output)`
- Finding 4:
  - Line 1: `String name = request.getParameter("name")`
  - Line 8: `output = name`
  - Line 11: `output = output + name`
  - Line 20: `out.println(output)`

(i) Explain briefly all vulnerabilities of this program. [10%]

(ii) Explain the four findings of the SAST tool and name one example for a false positive, false negative, and true positive.

[15%]

END OF QUESTION PAPER