COIT20262 - Advanced Network Security, Term 2, 2018

Assignment 1 Questions

Due date:	5pm Friday 24 August 2018 (Week 6)	**ASSESSMENT**
Weighting:	40%	**1**
Length:	N/A	

Instructions

Attempt all questions.

This is an individual assignment, and it is expected students answer the questions themselves. Discussion of approaches to solving questions is allowed (and encouraged), however each student should develop and write-up their own answers. See CQUniversity resources on Referencing and Plagiarism. Guidelines for this assignment include:

- Do not exchange files (reports, captures, diagrams) with other students.
- Complete tasks with `virtnet` yourself – do not use results from another student.
- Draw your own diagrams. Do not use diagrams from other sources (Internet, textbooks) or from other students.
- Write your own explanations. In some cases, students may arrive at the same numerical answer, however their explanation of the answer should always be their own.
- Do not copy text from websites or textbooks. During research you should read and understand what others have written, and then write in your own words.
- Perform the tasks using the correct values listed in the question and using the correct file names.

File Names and Parameters

Where you see *[StudentID]* in the text, replace it with your actual student ID. If your student ID contains a letter (e.g. "s1234567"), make sure the letter is in lowercase.

Where you see *[FirstName]* in the text, replace it with your actual first name. If you do not have a first name, then use your last name. Do NOT include any spaces or other non-alphabetical characters (e.g. "-").

Marking Scheme

A separate spreadsheet lists the detailed marking criteria.

Question 1. Cookie Stealing Attack

For this question you must use `virtnet` (as used in the workshops) to perform a cookie stealing attack. This assumes you have already setup and are familiar with `virtnet`. See Moodle and workshop instructions for information on setting up and using `virtnet`, deploying the website, and performing the attack. The tasks and sub-questions are grouped into multiple phases. You must complete all phases, in order.

Phase 1: Setup

1. Create topology 7 in virtnet.
2. Add a new normal student user to the MyUni grading system. The user must have:
   a. Username: *[StudentID]*
   b. Password: *[FirstName]*
3. Add a new malicious student user to the MyUni grading system. The user must have:
   a. Username: 12345678
   b. Password: *[StudentID]*
4. Add a grade for the normal student user for unit/course ‘coit20262’ with a grade of what you expect to receive this term, e.g. HD, D, C, P or F.
5. Change the title of the MyUni website by editing `header_footer.php` and changing the `<title>Grades</title>` line to:

   ```
   <title>Grades:[StudentID]</title>
   ```

6. Change the domain of the MyUni website to www.[StudentID].edu by editing the `/etc/hosts` files.
7. Test that the existing users and new student can access the grading website.

The roles of nodes in topology 7 are:

- `node1`: Web browser (lynx) of normal student user.
- `node2`: Web browser (lynx) of malicious student user.
- `node3`: Capture of packets with `tcpdump`.
- `node4`: MyUni grading website.
- `node5`: not used in this question.

Phase 2: Capture Cookies

8. Start capturing on `node3` using `tcpdump`.
9. The normal student user must do the following on `node1`:
   a. Visit the MyUni grading website, e.g. as below or with any options:
      `lynx http://www.[StudentID].edu/grades/`
   b. Follow the “Login” link and login
   c. Follow the “View grades” link and enter their username and ‘coit20262’ to view the course/unit grade, and submit.
   d. Follow the “Logout” link.
   e. Exit lynx by pressing q for quit.

10. Stop capturing on node3. Note that it is important that the start of the TCP connection (i.e. 3 way handshake), as well as all HTTP requests/responses are included in the capture.

Save the capture file as `normal-student.pcap`.

Phase 3: Masquerade Attack

Using information from the capture in part 2, the malicious student user must now perform a cookie stealing attack to masquerade as the normal student user. Although the capture may have recorded the normal student users' password, you MUST NOT use it in the cookie stealing attack (e.g. assume the password was encrypted). Your cookie stealing attack must only use the cookie information (not the password).

11. Setup for the cookie stealing attack on node2.
12. Start capturing on node 3 using `tcpdump`.
13. The malicious student user must do the following on node2:
    a. Visit the MyUni grading website, e.g.
    `lynx http://www.[StudentID].edu/grades/`
    b. Follow the "View grades" link and enter the username of the normal user, leaving the course/unit field empty (see you see all grades), and submit.
    c. Follow the "Logout" link.
    d. Exit lynx by pressing q for quit.
14. Stop capturing on node3. Note that it is important that the start of the TCP connection (i.e. 3 way handshake), as well as all HTTP requests/responses are included in the capture.

Save the capture file as `malicious-student.pcap`.

Phase 4: Analysis

Answer the following sub-questions regarding the previous phases and cookie stealing attack.

(a) Submit `normal-student.pcap`.

(b) Submit `malicious-student.pcap`.

(c) Draw a message sequence diagram that illustrates all the HTTP messages for the normal student user viewing the grades (i.e. the HTTP messages from `normal-student.pcap` from step 7 above). Do not draw any packets generated by other applications or protocols, such as ARP, DNS or SSH, and do not draw TCP connection setup or ACKS. Only draw HTTP messages. A message sequence diagram uses vertical lines to represent events that happen at a computer over time (time is increasing as the line goes down). Addresses of the computers/software are given at the top of the vertical lines. Horizontal or sloped arrows are used to show messages (packets) being sent between computers. Each arrow should be labelled with the protocol, packet type and important information of the message. Examples of message sequence diagrams are given in workshops. Note that you do not need to show the packet times, and the diagram does not have to be to scale.

(d) Based on your captures only, identify the following information. If the information is found in multiple packets, give the first packet from the capture. For example, if the information is found in packet numbers 3, 5 and 7, you would give the packet number as 3.

   a. Packet number from `normal-student.pcap` that contains the normal students' password
   b. Packet number from `normal-student.pcap` in which the server originally sends the cookie to the browser
   c. Last 4 HEX digits of the `id_hash` in the cookie (give the value of the last 4 digits, not the packet number)
   d. Packet number from `malicious-student.pcap` that contains the normal students grade for coit20262.
   e. Packet number from `malicious-student.pcap` in which the client originally sends the stolen cookie

(e) Explain how the `id_hash` is calculated, giving both the equation/algorithm for calculating it, as well as a description of the values used in calculating it (for example, where do the values come from? How are they set?).

(f) Explain how the `id_hash` provides security on the context that it is used in the grading web application.

(g) Explain a weakness or vulnerability of how the `id_hash` is calculated or used. For example, how could the security it provides be broken?

(h) In this question, the username and password of the normal student user are sent as plaintext from browser to server. This is an obvious weakness, as an attacker that intercepts the packets between browser and server immediately learns the password. A possible solution is for the client browser to calculate a hash of the password using JavaScript, and sending the hash of the password to the server, instead of the actual password. Discuss the strengths or weaknesses of such a scheme with respect to preventing an attacker from logging in using the normal student users' password.

(i) In this question, the malicious student performing the cookie stealing attack uses lynx as a web browser. Explore how to edit or create cookies in common graphical web browsers (Firefox, Chrome, IE, Edge or Safari). Give a brief explanation of what you need to do to modify/create cookies (e.g. which options of the browser, or what software needs to be installed) and take a screenshot of a cookie you modified or created. The cookie in the screenshot MUST include your *[StudentID]* (e.g. put your *[StudentID]* in any field of the cookie).

Question 2. Cryptography

For this question you must use `openssl` to perform a set of cryptographic operations. When performing cryptographic operations you must be very careful, as a small mistake (such as a typo) may mean the result is an insecure system. Read the instructions carefully, understand the examples, and where possible, test your approach (e.g. if you encrypt a file, test it by decrypting it and comparing the original to the decrypted). It is recommended you use `virtnet` to perform the operations.

The tasks and sub-questions are grouped into multiple phases. You must complete all phases.

Phase 1: Download

Normally in public key cryptography you generate your own public/private key pair. However in this assignment to simplify the tasks, the Unit Coordinator has generated a key pair for you. Your key pair will be available to you on Moodle to download, with filename:

- *`[StudentID]`*`-keypair.pem`

In addition to your key pair, a number of files will be available to all students on Moodle to download. Each file starting with *[StudentID]* must be downloaded by you. You may also need to download files with other student's IDs (see the next phase).

The download URL will be published on Moodle.

Phase 2: Read the Messages

The files for download have been created by another student, denoted as the sender:

1. Sender student created a message to you *`[StudentID]`*`-message`*`[N]`*`.txt`, where *`[N]`* is an integer, e.g. 1, 2, 3, …
2. The sender signed the message to produce *`[StudentID]`*`-message`*`[N]`*`.sgn`.
3. The sender wrote their student ID into a text file *`[StudentID]`*`-sender`*`[N]`*`.txt`.
4. The sender signed the sender file to produce *`[StudentID]`*`-sender`*`[N]`*`.sgn`.
5. The sender used openssl to generate a random 256-bit secret key, in Hex, and saved it in *`[StudentID]`*`-key`*`[N]`*`.txt`.
6. The sender generated a random Initialisation Value (IV), in Hex, and saved it in *`[StudentID]`*`-iv`*`[N]`*`.txt`.
7. The sender encrypted the message using symmetric key encryption, the random secret key, and the random IV, producing *`[StudentID]`*`-message`*`[N]`*`.enc`.
8. The sender encrypted the random secret key file using public key encryption, producing *`[StudentID]`*`-key`*`[N]`*`.enc`.

The sender then sends to you the necessary files from above.

Note that the files were actually created by the Unit Coordinator, but in this assignment you can assume they were created by a student. The "sending" of files to you is performed by the sender (Unit Coordinator) uploading them to Moodle, and you downloading them from Moodle.

Your task is, for every message, decrypt and verify the files. Be careful: there may have been attacks on some messages!

The algorithms used in this question are:

- Public key: RSA, 2048 bit
- Symmetric key: AES-256-CBC
- Hash: SHA256

Phase 3: Report Your Results

After downloading, decrypting and verifying all messages, you need to create a summary of the results for each message. The summary must be in a text file called `*[StudentID]*-summary*[N]*.txt`. The summary must contain exactly two lines, of the format:

```
ResultType
Message
```

where `ResultType` is one of the following strings:

- `Success` – means all files successfully decrypted and successfully verified.
- `FailDecryptKey` – means the decryption of secret key was unsuccessful.
- `FailDecryptMessage` – means the decryption of message was unsuccessful.
- `FailVerifySender` – means the verification of sender file was unsuccessful.
- `FailVerifyMessage` – means the verification of message was unsuccessful.

If `ResultType` is `Success`, then include the contents of the message on the next line. If `ResultType` is another value, then include "`NoMessage`" on the next line. Examples of possible summary files are:

Example 1:

```
Success
12345678-3-hello
```

Example 2:

```
FailDecryptMessage
NoMessage
```

Example 3:

```
FailVerifySender
NoMessage
```

You must sign each summary file, producing `*[StudentID]*-summary*[N]*.sgn`.

Phase 4: Analysis

(a) Submit all summary text files, e.g. `[StudentID]-summary1.txt`, `[StudentID]-summary2.txt`, `[StudentID]-summary3.txt`, ...

(b) Submit all summary signature files, e.g. `[StudentID]-summary1.sgn`, `[StudentID]-summary2.sgn`, `[StudentID]-summary3.sgn`, ...

(c) The sender generated a random 256-bit secret key to be used for encryption. Consider if the sender instead used the following approach: generate a random password of 12 uppercase or lowercase English letters (the password only contains letters; no numbers or other characters), and then apply SHA256 on that password, using the hash value as the encryption key. Discussion the security issues with such an approach of generating a secret key for AES-256-CBC encryption.

(d) The sender encrypted the random secret key, but not the IV. Discuss the security issues with not encrypting the IV.

Question 3. Ransomware Research

Ransomware attacks are increasingly publicised. In addition it is estimated there are many more ransomware attacks not being made public, e.g. companies and users paying a ransom but not disclosing the attack. The prevalence of ransomware, and the impact it has on organisations, has led to the discussion of ransomware insurance. Your task is to study what is ransomware, what are the challenges and possible countermeasures, and report on it in an easy-to-understand manner. You must write a short report on ransomware, covering the following sections.

Overview of Ransomware

Approximately ½ page of text explaining what is ransomware and how it works. Assume the audience of this section is the general public (non-technical). You must include real examples of ransomware and/or ransomware attacks.

Technical Details of Ransomware

Approximately ½ to 1 page of text explaining the technical aspects of ransomware, including:

- What are the common methods of infection?
- What are common payloads?
- What cryptographic techniques are commonly used?
- What technologies are used to obtain ransoms?
- Why are some ransomware very hard to break?

Assume the audience of this section is technical, i.e. have similar background on network security as you. You should refer to techniques and concepts covered in the unit, and give sufficient technical detail to demonstrate you understand the issues.

Recommendations

List and explain 4 recommendations for end-users and/or organisations to avoid ransomware and/or handle ransomware infections.

There is no minimum/maximum length of the report. As a guide 1 to 2 pages of text (not including pictures) may be appropriate. Do NOT include pictures or tables in the report. Use text only. While you may use numbered lists and dot points, the report cannot entirely be lists. References are not necessary (although the normal rules of academic integrity are expected).