Document Version: 1.0
Submission Guidelines

- **Deadline:** Assignment including creation and sharing of Panopto video recording is due on Monday 16 th September 2019, 08:00 AM.
- **Submission Files:**

  1. A report in PDF file format of maximum 6 pages as a reference. The appendices are excluded from the page count.

  Notes:

  1. Do not submit a compression of multiple files. Such submissions may risk losing partial or complete assignment marks.
  2. A handwritten document is **not** acceptable and will **not** be marked even if converted and submitted electronically.

- **Group Assignment:** This is a group assignment with maximum 2 members per group. You must form your group in the same tutorial you are enrolled. You may complete the assignment individually however the scope will not be reduced. You may form a group with a student in another tutorial if there are odd number of students in the tutorial however this must be communicated with the tutors of both tutorials and the lecturer and you must attend the interview as a group.
- **Submission Platform:** Electronic submission via Moodle for the report, interview video, python les and CORE emulator conguration les. The compressed le of the containers must be submitted using another method. The submission approach that will be followed will be announced closer to submission deadline. Only one submission per group is needed.
- **Filename Format:** Name your files for different assignment tasks as follows. consider that SID is the Student ID of the group member that will make the submission on behalf of the group.
  Submission via moodle:

  1. report_SID.pdf
  2. secure_modbus_server_SID.py
  3. secure_modbus_client_SID.py
  4. possible security parameter files (eg. files created by openssl)
  5. wireshark_pcap_SID.pcap
  6. core_SID.imn
  7. containers_SID.zip (the containers names must be modbusServer and modbusClient)

- **Required Student Information:** Please do not forget to include the name and student id of both group members within the report PDF file content.
- **Note:** You must strictly follow the provided file name format or **5 Marks penalty** will be applied per incorrect filename.
- Python code should be written for python version 3 and work on the provided VM without any additional library installment apart from the modbus libraries (if is deemed necessary)
- **Late Submission Policy:** Submit a special consideration form (available on moodle) to formally request a late submission.

- **Late Submission Penalty:** A late submitted assignment without prior approval will receive a late penalty of 100% deduction per day (including Saturday and Sunday) or part thereof, after the due date and time.
- **Academic Integrity/Plagiarism:** It is an academic requirement that your submitted work be original. Zero marks will be awarded for the whole submission if there is any evidence of copying, collaboration, pasting from websites, or copying from textbooks.
  Note: Plagiarism policy applies to all assessments.
- **Grading Procedure:**
  - To receive a grade for the assignment you must create a video recording of **maximum 30 minutes** where you will demonstrate and explain your work using **Panopto platform** (further instructions will be provided via moodle).
  - You must create the recording and share it with your tutor via Panopto platform by the due date specified.
  - Both group members must participate during the demonstration and the face of the person who is explaining a task must be recorded.
  - If you have any privacy concern regarding the Panopto platform then you need to raise it with your tutor and organize an interview at least a week before the deadline for the demonstration.
  - You can use the report and any other notes you have prepared beforehand to help you explain and demonstrate your work.
  - Both group members will receive the same mark for the assignment. You must bring up any issues within the group as early as possible to the attention of your tutor and must be prepared to complete the assignment individually if the conflicts cannot be resolved.
- **IT Use Policy**: Your submission must comply with Monash University's IT Use Policy.

Marks

- This assignment is worth **30%** of the total unit marks.
- The assignment is marked out of **60** nominal marks.

Assignment Activities

1. **[30 Marks]** This exercise is based on the very popular MODBUS protocol that is widely used in industrial networks to report a Programmable Logic Controller (PLC) results to a SCADA system. The protocol has a simple structure and was designed primarily to provide fast, almost real time response and safety. There are a lot of information that can be found over the internet on how MODBUS works. We are focusing on the MODBUS version that runs over TCP. There are two district types of entities in MODBUS, MODBUS server and the MODBUS client (or MODBUS slave and master respectively). Indicatively, some websites that you can find more information on MODBUS are the following:

   `https://en.wikipedia.org/wiki/Modbus`
   `http://www.modbus.org/`
   `https://www.modbustools.com/modbus.html`

   Some videos on youtube that might help you:
   `https://www.youtube.com/watch?v=txi2p5_OjKU`
   `https://www.youtube.com/watch?v=JBGaInI-TG4`
   Most importantly you can find a full implementation of the Modbus protocol in:
   `https://pypi.org/project/uModbus/`

   (a) Perform a small analysis of the MODBUS protocol and report the status of the protocol in terms of security. Highlight possible security issues and describe an attack model on the protocol. Note, that the protocol was not designed to be secure

   (b) We need to provide some enhancements in order to increase the MODBUS security status. Such enhancements must enforce in MODBUS the security principles of device authentication, data confidentiality and data integrity. To achieve that, design and implement in python, a simple protocol that will be able to support the above principles by specifying the following:

      i. An appropriate key agreement scheme to decide a session key between Server and Client. The session key extraction mechanism must include the use of a salt value

      ii. The message integrity and encryption/decryption schemes that will be used in order to secure the communication channel
      Note: You will need to specify the appropriate security mechanisms and generate the correct keys/credentials for them

   (c) After implementing a secure MODBUS Client and Server as two autonomous python programs that can be executed in a single machine, two containers need to be created in the laboratory VM. One container will refer to the secure modbus server and the other to the secure modbus client.

   Use the core network emulation in order to emulate a scenario where the Modbus devices communicate. For example, you can create a network consisting of a single router with two interfaces each connected to one of the defined virtual bridges and then associate the containers with the bridges.

   (d) Provide a Wireshark capture pcap file where the functionality of the presented protocol is demonstrated. Describe and Analyze the pcap file entries in your report to justify the protocol correct usage.

   Notes:

   (a) For all the cryptographic primitives use the cryptography module of python pyca (`https://cryptography.io`). If a cryptographic primitive is not supported by the installed openssl version on the VM you can use an alternative primitive that achieves the same goal.

(b) For the umodbus library you must follow the installation instructions of the website. You will probably need internet access on the VM and/or the containers for installing **pip**. Be careful to install it for python 3.

(c) Your choices of authentication method, key exchange, and symmetric encryption methods will affect your grade in this task. Choose algorithms that would be considered secure

(d) The authentication method does not rely on external services (for simplicity). Not Trusted Third Party should be needed.

(e) You can use the example code for the Modbus Server and client as a starting point. Assume that you are performing the Modbus actions described in the provided library client/server example.

(f) You can use configuration files for both client and server to feed any required information for the server and client such as security parameters etc. Python provides easy to use and powerful functions for reading and parsing files or strings. Few examples are:

i. YAML where the content can be read directly to python data structures.

ii. Simplified INI file where options are specified as keyword=value one per line for which you can use the split() function to separate the keyword from the value. Similarly you can use keyword value with one option per line (space as separator as well as any other character that will not appear as part of the keywords or values).

iii. For simplicity you can use os.urandom() function directly for random values whenever needed in your protocol. You can also use the python random library.

iv. You can mimic a simplified version of protocols you learn in the subject regarding negotiation, authentication, key exchange, key derivation, and encryption used in your protocol.

v. To start you can use the VM as both client and server and once you completed the protocol test it in two separate containers.

2. **[30 Marks]** Let us assume that we have the network that is provided in your individualized Core Network Emulator configuration file (use only one of the two members file). The network includes a factory organization that has two structures. One of them is the actual factory (in a physically remote location) and the other is the factory central facility. In the remote location there is an industrial network with PLCs and modbus sensors that are collecting data and sending them to a SCADA Entity installed in the central factory facility. Also, the factory has a web-site that is accessible to all users (in the central facility) and a local webserver in the remote factory premises for factory personnel only (that are in the central factory network or the remote factory premises)

(a) **[15 Marks]** In this task, you will try to provide a security mechanism for modbus that is different than the previous one. This time you need to use VPN. Thus, you will configure a VPN between the remote factory facility and the central factory of acme.sec. For this you will use two containers one for the modbus server and one for the modbus client that need to be assigned to the appropriate virtual interfaces of the CORE emulator configuration file.

i. You need to create a site-to-site IPsec VPN using strongswan extension of CORE between the two gateway nodes of the network (ie. bast (factory) and osiris (central facility).

ii. The VPN gateways must use public key authentication and Fully Qualified Domain Names (FQDN) that match the CN (Common Name) field in their certificates.

iii. Using the above presented solution, what kind of security principles (or objectives) are achieved? Does the VPN solution solves all Modbus security problems? Justify your answer in the report and the interview.

Note: The DNS records are created for this task.

(b) **[15 Marks]** In this task, you will work with the original configuration file that is provided to you and the attached modbus containers on this emulated network. You will need to configure appropriate firewall rules on the bast and osiris gateways to protect the network appropriately. More specifically:

i. The modbus communication between the client and the server should be possible. Note that typically Modbus uses a specific port.

ii. The Acme central website must accessible to all users

iii. The Acme remote factory website must be accessible only to the central facility users.

iv. The firewall on bast must protect the factory network against any other unauthorized communication. This includes the gateway itself.

v. The firewall on osiris must protect the central facility networks and the DMZ against any other unauthorized communication. This includes the gateway itself. Note: Use the firewall service in security section of services on bast and osiris for this task.