Asean Factori 4.0
Across South East Asian Nations: From Automation and Control Training to the Overall Roll-out of Industry 4.0
Erasmus + Project, 609854-EPP-1-2019-1-FR-EPPKA2-CBHE-JP

# TP8: Cybersecurity - Attacks and Protection

**Operational objectives :**

- To become aware of the different attack vectors and to understand at what levels the different attacks occur
- Understand the different possible objectives of an attack
- Be able to execute an attack script in Python and analyse the attack
- Be able to protect yourself from different types of attacks

**Prerequisites :**

- Master the basic operations of Control Expert
- Be able to launch Vijeo Designer in simulation mode
- Understand the basic structure of a Modbus frame and its main functions
- Have understood the main principles of SNi40 configuration

**The problematic posed:**

To become aware of the different vectors and levels of existing cyber attacks, to understand through the use of several attack scripts the different possible objectives of an attack on an industrial network, and to be able to protect oneself from these various attacks.

**Resources :**

- **Manufacturer documentation**
  - Schneider Electric
    - ➢ website
    - ➢ protocol-modbus.pdf
  - Stormshield :
    - ➢ SNS - User and Configuration Manual
- **Specific documentation**
  Architectures Cybersec model.pptx
  Applications made available for the realization of this TP :
  - M580 application (Control Expert): cybersec_M580
  - HMI application (Vijeo Designer): cybersec_IHM
  - Default SNi40 Firewall configuration file (SNI40-TP2-0.na)
  - 3 attack scripts (Modbus) written in Python (V2.7) : m580_read_memory.py, m580_dos.py, random_speed.py (to be launched in the command prompt window)
- **Software provided,**
  **to be installed on the work PC (console) for the realization of this TP:**
  - Control Expert (Schneider Electric) : Programming of Schneider Electric M340, M580, …
  - Vijeo Designer V6.2 SP8 : Design of Magelis HMI applications (execution including in Simulation mode on the Workstation)
  - Web Gate Client (Schneider Electric) : complement to Vijeo Designer [option] (remote client of the Magelis HMI)
    (remote client of the Magelis HMI, running in an Internet Browser)
  - Internet Explorer : Microsoft Internet Browser
  - Angry IP Scanner (angryip.org): check for accessible IP addresses in a given range [option]
  - Wireshark (Wireshark Foundation): observation of Ethernet frame details
  - WinSCP/Putty: manipulation of Custom Patterns files on SNi40
  - Python code execution environment (V2.7.15)

| **Evaluation Criteria:** | 😄 | 😐 | ☹ |
|---|---|---|---|
| Awareness of the different vectors and levels of attack | | | |
| Understand the different possible objectives of an attack | | | |
| Be able to execute a Python attack script and analyze the attack | | | |
| Be able to protect yourself from different types of attacks | | | |
| Autonomy - Quality of work/restitution | | | |

| **Time spent:** | 2 h | Objective(s): | | Comment(s) : |
|---|---|---|---|---|
| **Assessment :** | / 20 | Reached(s) | Not reached | |

**TP8 - Attacks and Protections**

**Objective:** The objective of the following section is to understand the different attack vectors used and the existing threats.

1. Using Appendix 1 "The 4 main attack vectors", explain the different attack vectors, giving, if possible, an example of an attack using each vector. Classify these vectors in order of ease of exploitation.
2. Give at least 3 types of malicious software, indicating, if possible, an example of known software for each type.

---

**Context:** 3 malicious scripts (written in Python) were received by email and were executed on the target system connected to the M580. The objective of this section is to analyse these 3 attack scripts.

3. Run the script "m580_read_memory.py" from the command line via a Python 2.7 interpreter and interpret the result. Open the file "m580_read_memory.py" with a text editor and analyse the contents of the file. Explain briefly the process of the flaw exploited and what the exploitation of this flaw implies (You may use the document provided "protocol-modbus.pdf").

   **N.B.**: It is possible that the script returns an error indicating a loss of connection with the PLC. If this is the case, check the accessibility of the PLC using its IP (which must correspond to the one entered in the script), wait a few seconds and then run the script again.

4. Run the script "m580_dos.py" from the command line using a Python 2.7 interpreter and interpret the result. Open the file "m580_dos.py" with a text editor and analyse the contents of the file. Explain briefly the procedure of the flaw exploited and what the exploitation of this flaw implies (You may use the document provided "protocol-modbus.pdf").
5. Run the "random_speed.py" script from the command line using a Python 2.7 interpreter and interpret the attack. Open the file "random_speed.py" with a text editor and analyse the contents of the file. Explain briefly the procedure used in this script and what the execution of such a script involves (You may use the document provided, "protocol-modbus.pdf").

---

**Objective:** The objective of the following section is to secure the network at different levels using the SNi40

6. Explain very briefly the model provided in Annex 2. Justify your answer by determining at which level(s) an attack could potentially be more effective and therefore at which level(s) a protection system could be more effective (Note: this does not imply that protection at other levels is optional: cybersecurity is implemented at all levels!

7. Using the SNi40 configuration, establish examples of protection at the application layer to stop one or more of the attacks previously studied (check that the attack(s) are indeed stopped).
8. Using the SNi40 configuration, establish an example of protection at the network layer to stop the attacks previously discussed.
9. Still using the SNi40, establish an example of protection at the link layer to stop the attacks previously studied.

**Note:**

This exercise will be performed on a phase 5 architecture with a separate industrial network (via the NOC) and separate firewall network (at the SNi40). The "SNi40 - Splitted Networks.na" configuration of the SNi40 is provided and can be set up before the start of the test.

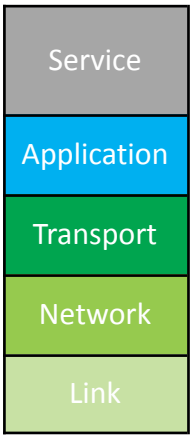
The M580 and HMI programs have been designed to make it easy to see when attack scripts are being executed. These same attack scripts are kept simple. Of course, in real attack scenarios, attacks are often much more complex.

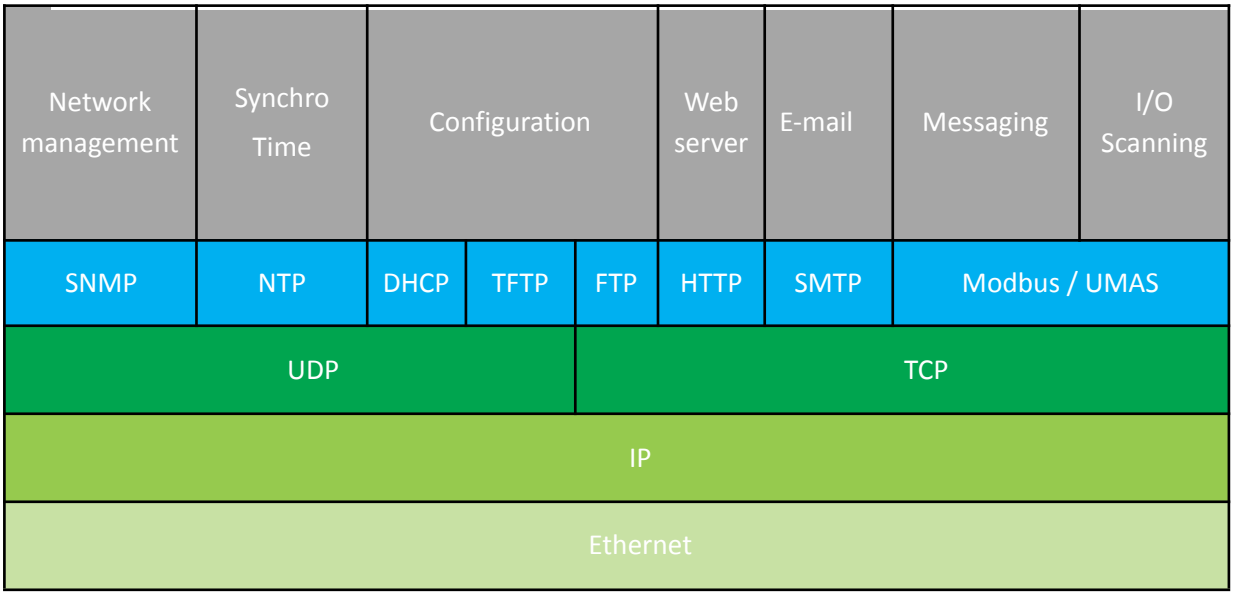
In order to correctly exploit the "m580_read_memory.py" and "m580_dos.py" attacks, firmware files for the M580, including the exploited flaws, are provided: "M580_BMEP581020_SV2.70.ldx" and "M580_BMEP582040_SV2.70.ldx" depending on the M580 version. The teacher is strongly recommended to check the correct functioning of the test with the firmware version contained in the M580 and to downgrade if necessary, before running the test. If the M580 has a firmware version higher than version 2.70, it is more than likely that the flaws are no longer exploitable because they have been corrected.

It is also important to remember that this course is intended to teach good practice and knowledge in terms of cybersecurity within industrial networks and not to learn how to create malicious software: the scripts and scenarios are therefore very simple. In practice, the attacks are much more complex. For example, the scripts used in this tutorial were provided in the form of a Python script but could very well have been implemented, in other forms (shellcode ...), in seemingly innocuous files such as a PDF, an image, or others ... by exploiting flaws in the software responsible for reading these documents to execute.

## Appendix 1: The 4 main attack vectors

## Appendix 2: TCP/IP Model




**Details of expected operations**

1. **Attack vectors**

As highlighted in Annex 1, the 4 main attack vectors are the following (ranked in order of exploitability):

- **Human vector**: the most widely used attack vector, exploiting or targeting an individual. It is the vector used in particular by attacks such as phishing emails, or "president scams".

- **Software vector:** a vector that most often uses a flaw in software that can be found at different levels: basic software, operating system, kernel, or even firmware. In general, the lower the level of the software, the more complex it is to exploit its flaws: for example, exploiting a flaw in a CPU firmware (Spectre and Meltdown attacks on Intel microprocessors) requires significant knowledge and a very thorough understanding of said microprocessors.
- **Network vector:** vector targeting communication protocols within a network and/or exploiting the network configuration. This is one of the vectors exploited by the Heartbleed attack targeting the TLS (Transport Layer Security) protocol used in particular for secure communication between a web server and a user.
- **Physical vector:** attack vector consisting in attacking the system from a physical point of view. This vector is for example used by the Stuxnet attack which started with the physical introduction of a USB key on a machine. Moreover, a physical attack can be much more complex, for example by exploiting physical measurements (calculation time, CPU temperature, etc.) in order to deduce sensitive information (e.g. TEMPEST type attacks use the electromagnetic radiation of a machine to deduce confidential information).

Regarding the exploitability of each vector, this order can be justified by pointing out that it is much easier to send a phishing email to an employee (human vector) asking him to connect to a malicious site with his professional credentials than to break into the company's premises (physical vector) (assuming it is difficult to access) to insert a USB key on a machine.

Furthermore, the vulnerabilities exploited by the network vector are (most of the time) related to the software enabling communications and configurations in the network. Acting (in general) on a larger number of machines, this attack vector exposes the attack even more to discovery. For these reasons, the network vector is usually more difficult to exploit than the software vector and requires more knowledge (systems and networks).

Finally, it should be noted that today, because of (or rather thanks to) the strengthening of cybersecurity, there are hardly any attacks that are content to exploit only one attack vector: attacks exploit different attack vectors at different levels. For example, the notorious WannaCry ransomware exploits both the network vector (the Windows flaw that allowed its rapid spread) and the human vector (obviously only makes money if the user decides to pay the ransom).

### 2. Types of malware

There are different types of malware, including :

- **Viruses**: (most commonly) malicious software that runs directly or through a document or other software (often illegally obtained) in order to retrieve/alter information or damage the target system. (e.g. Brain)
- **Worms:** software capable of spreading autonomously by duplicating itself from machine to machine, like the virus it can have different objectives such as acting on data. (ex: Stuxnet)
- **Trojan horse:** software that looks legitimate but has malicious features that are used to install other malicious software without the user's knowledge, often by opening a backdoor. (e.g. Zeus, Havex, ...)
- **Rootkit:** surely the most formidable malware, not necessarily because of its actions on the system, but because its main objective is to remain stealthy by hiding from possible security software (antivirus etc...) and to maintain access to the machine. It can also be used to hide other malware. (ex: XCP)

- **Adware:** malicious software that is annoying because its purpose is to display advertising and/or collect data that can be used for the same purpose, but generally not very dangerous.
- **Ransomware:** malicious software that usually encrypts user data in order to demand money for the recovery of data. (ex: WannaCry)

Note that some malware today is broken down into different parts, each of which can be considered a different type of malware: for example, malware can consist of installing a rootkit, to hide itself and act as a Trojan horse opening a backdoor for virus execution. The same software can also act as a worm and duplicate itself.

On the other hand, some malware have capabilities that make them very difficult to detect: polymorphism, …

**3. Attack: Reading of sensitive information**

Before running the script, Vijeo Designer can be launched in simulation mode in order to have access to the GUI from the computer.

Run the python script "**m580_read_memory.py**", which then asks for the IP address of the M580 PLC, as well as the port on which the PLC communicates through the Modbus protocol. Once this information has been entered (you can also press "Enter" without entering the information to use the default information), the attack script is launched.

Note that during the attack, the GUI displays a "You have been hacked!" window indicating that the attack is in progress:

**FOTO**

If the script was executed correctly, the following result is obtained:

**FOTO**

You will notice that the script displays the names of the different SNMP communities: set, get and trap.

SNMP is a network management protocol. A device with SNMP functionality has an SNMP directory in which a lot of information about the system and its connectivity to the network is stored. The device can then receive different requests through different identifiers:

- The "set" community allows the modification of variables contained in the SNMP directory, it is a read/write access.
- The "get" community authorises the reading of variables contained in the SNMP directory, it is a read only access.
- The "trap" community which allows you to manage alert messages.

Depending on the SNMP version used, these different community names may allow direct access to information without the use of an additional password: the simple possession of the community name is then a considerable advantage.

Normally, the attack returns the default community names as shown in the example: private, public and alert. In order to verify that the attack returns the real community names established on the M580, it is possible to modify these names in the PLC program using Unity :

**FOTO**

Save the changes made, regenerate the project and then transfer the program to the M580 controller.

Running the attack script again gives a new result:

**FOTO**

The script then returns the new community names that have been filled in, indicating that the script has successfully retrieved the information and thus the effectiveness of the attack.

N.B.: Remember to restore the names to their original value.

When we open the python program of the attack, we can observe the following main functions:

- set_screen: this function is only used to change the screen displayed on the HMI, in particular to display the "You have been hacked!" screen which is provided in the HMI program.
- init_download: initiates the download of the M580 strategy program.
- download: download the M580 strategy program in blocks.
- parse_strategy: formats the retrieved content and after analysis returns the part concerning SNMP information.
- parse_snmp: formats the retrieved SNMP information and, after extraction, returns sensitive information about SNMP community names.

Finally, this attack allows sensitive information to be recovered through normal use of the Modbus protocol on the M580, which leaks information via the procedure for downloading the strategy program.

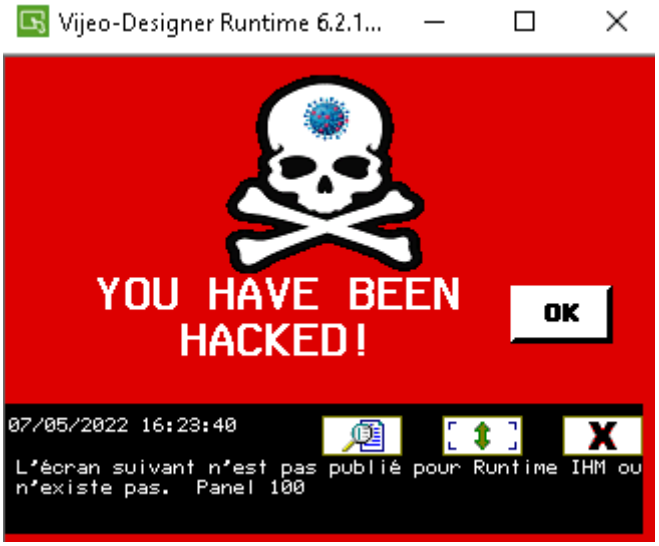
As it does not intervene in the system, this attack is passive, as it merely retrieves information. This type of attack can remain quite discreet, and will only be detected by analysing the packets in the network very efficiently. Nevertheless, the attack does not prevent a more active threat from being carried out as a result of the information retrieval.

### 4. Attack: Denial of Service

Before running the script, Vijeo Designer can be launched in simulation mode in order to have access to the GUI from the computer.

Run the python script "**m580_dos.py**", which then asks for the IP address of the M580 PLC, as well as the port on which the PLC communicates through the Modbus protocol. Once this information has been entered (you can also press "Enter" without entering the information to use the default information), the attack script is launched.

When the attack is launched, the HMI will display the "You have been hacked!" screen, and then after a while the HMI loses the connection to the M580 PLC.




On the console side, the correct execution of the attack gives the following result:

**FOTO**

# Error

If we look at the information displayed by the attack script, we can see that the script, after sending its payload, verifies that the M580 becomes unavailable, which is indeed the case, and confirms that the Denial Of Service (DOS) attack was successful.

When looking at the control information on the M580, we can see that the M580 has gone into an error state, as well as other components that were driven by the M580 (in reality the other components have not been affected, they are entering an error state because the M580 is not responding). Concerning the Magelis HMI, it has also lost access to the M580.

An attempt can be made to connect to the M580 using Unity, via USB or Ethernet, without success: all communications with the M580 have been cut off and the M580 has entered an error state from which it cannot recover.

To restore the M580's operation, it is necessary to turn off the power (using the main switch) and then turn the power back on: the M580 will then restart and be accessible again.

However, even though it has been powered up again, the PLC is still unavailable, even though it is no longer in its error state. The PLC program has also been stopped, so it is necessary to connect to the PLC, using Unity (via USB or Ethernet), and restart the PLC program. Once this has been done, communication with the PLC is restored.

When the python program for the attack is opened with a text editor, the following functions used for the attack can be found:

- set_screen: this function is only used to change the screen displayed on the HMI, in particular to display the "You have been hacked!" screen which is provided in the HMI program.
- dos: This function sends a specially crafted packet using UMAS function code 0x28, corresponding to a request to read a physical address from the M580 memory, with block number 0x0030. Receipt of such a packet will cause the M580 to enter an error state.

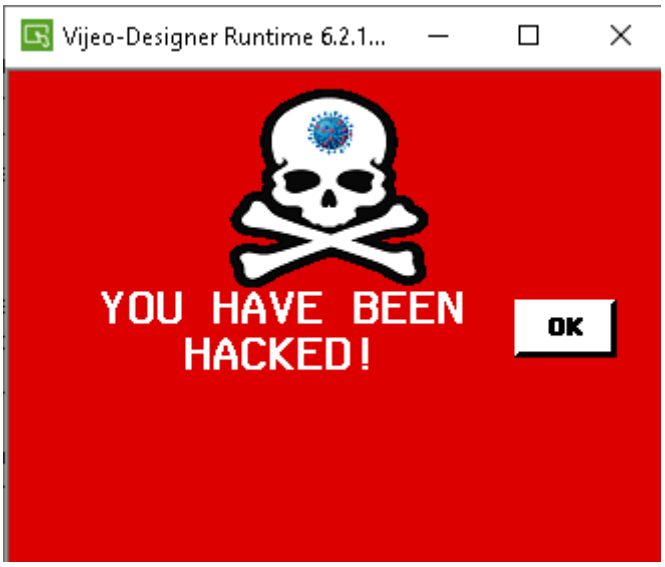
Unlike the previous attack, this is an active attack and can have serious consequences: it causes an immediate shutdown of the PLC, which can lead to a halt in productivity and even material and/or physical damage. Furthermore, physical intervention is required to bring the M580 back into service.

### 5. Attack: Takeover

Before running the script, Vijeo Designer can be launched in simulation mode in order to have access to the GUI from the computer.

Run the python script "**random_speed.py**", which then asks for the IP address of the M580 PLC, as well as the port on which the PLC communicates through the Modbus protocol. Once this information has been entered (you can also press "Enter" without entering the information to use the default information), the attack script is launched.

When the attack is launched, the HMI will display the screen "You have been hacked! :




While the attack is in progress, the drive will change speed every 3 seconds for about 30 seconds, the console will give a result similar to the following:

```
C:\Python27>random_speed.py
Automate IP address ? (Default: 192.168.0.1)

Modbus Port ? (Default: 502)


[+] Connecting to 192.168.0.1:502...
[+] Connected to 192.168.0.1:502.
[+] Setting speed to 50...
[+] Speed set to 50.
[+] Setting screen to 404...
[+] Screen set to 404.
[+] Starting speed controller...
[+] Speed controller started.
[+] Setting speed to 171...
[+] Speed set to 171.
[+] Setting speed to 79...
[+] Speed set to 79.
[+] Setting speed to 153...
[+] Speed set to 153.
[+] Setting speed to 169...
[+] Speed set to 169.
[+] Setting speed to 45...
[+] Speed set to 45.
[+] Setting speed to 30...
[+] Speed set to 30.
[+] Setting speed to 162...
[+] Speed set to 162.
[+] Setting speed to 11...
[+] Speed set to 11.
[+] Setting speed to 24...
[+] Speed set to 24.
[+] Setting speed to 174...
[+] Speed set to 174.
[+] Setting screen to 1...
[+] Screen set to 1.
[+] Stopping speed controller...
[+] Speed controller stopped.
[+] Closed connection to 192.168.0.1:502.
```

When the python program for the attack is opened with a text editor, the following functions used for the attack can be found:

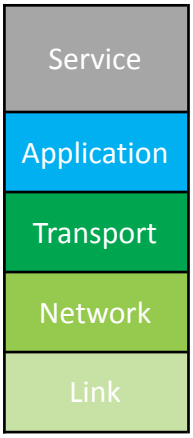
- set_screen: this function is only used to change the screen displayed on the HMI, in particular to display the "You have been hacked!" screen which is provided in the HMI program.
- start: This function sends a Modbus packet to start the drive.
- set_speed: This function sends a Modbus packet to change the value of the word in memory in the M580 that is used to set the drive speed.
- stop: This function sends a Modbus packet to stop the drive.

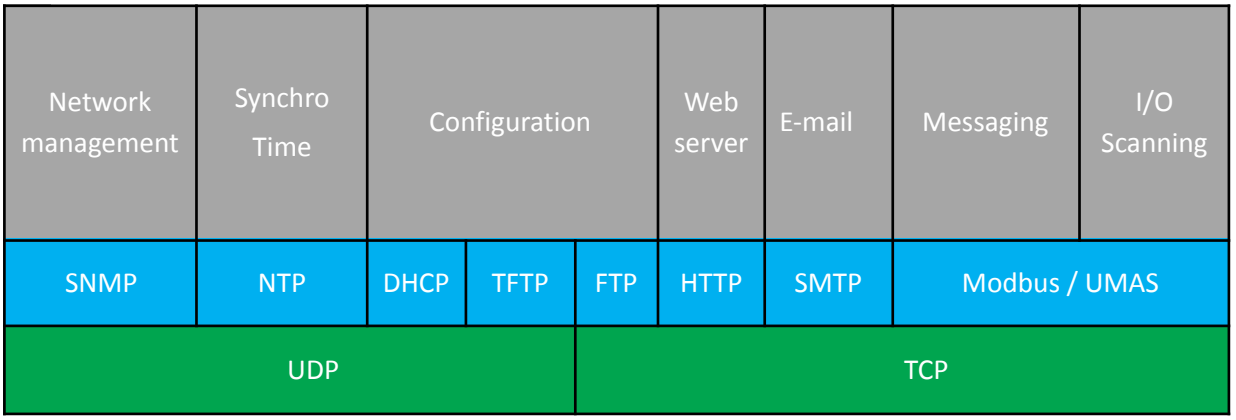
It should be noted that for this attack script, no vulnerability is used to achieve the result. It is therefore an attack that could be carried out either by a person with direct access to the network (it should be remembered that the majority of attacks today come from the inside), or by a person with access to the network through corrupted equipment (e.g. an administrator PC).

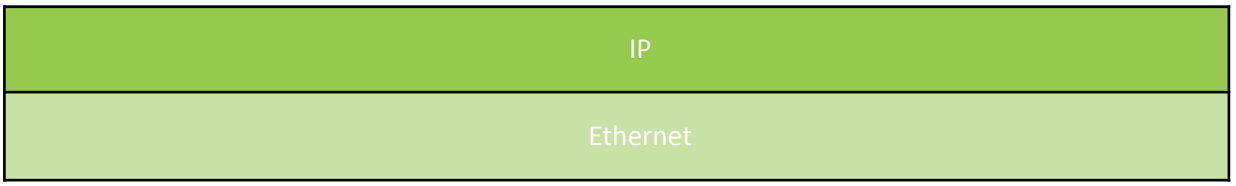
This attack shows that with a bad network configuration, a bad security set up, or even through a bad manipulation of an administrator (opening a fraudulent email attachment, hence the interest in informing the staff of good cybersecurity practices), an attacker can take control (partially or totally) of the industrial network.

**Important**: The attacks used in this tutorial are only simple examples, exploiting vulnerabilities or not. It is important to be aware that some attacks can be much more complex and exploit several vulnerabilities, or even exploit vulnerabilities accessible through other vulnerabilities (this is the case, for example, of the script in an email attachment executed by an administrator, this attack takes advantage of the administrator access through a human error)

## 6. TCP/IP model






The model provided in Appendix 2 corresponds to a TCP/IP network model. This model, which is based on the OSI model, is broken down into different interlocking layers. There are 4 important layers ordered from the lowest to the highest layer:

- Link: this is the lowest level layer (because here we are not considering the physical layer corresponding to Ethernet cables, and physical aspects allowing data transmission). It mainly concerns the communication and identification of machines between them, using the Ethernet protocol and in particular via MAC addresses.
- Network: this is the layer that simplifies communication between machines using the IP protocol: within a local network, each IP will be associated with a MAC address, and the use of a gateway, which is often a manageable switch or router, allows communication from the local network to other networks (e.g. Internet).
- Transport: this is the layer that designates the method by which data will be transmitted. Today, there are two major transport methods: TCP, which is mainly used, and which is "connection" oriented (two machines establish a connection, identify themselves and "agree" on certain parameters before communicating) and UDP, which is not "connection" oriented (apart from the IP address, there is no information about the sender of a packet).
- Application: This is the layer for application protocols that allow two applications (of the same nature) to communicate according to a well-defined protocol.

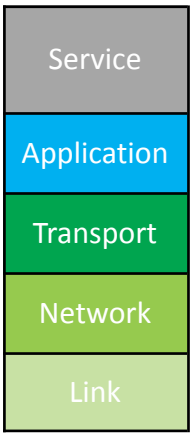
As the service layer does not exist, it only gives information about the use of application protocols.

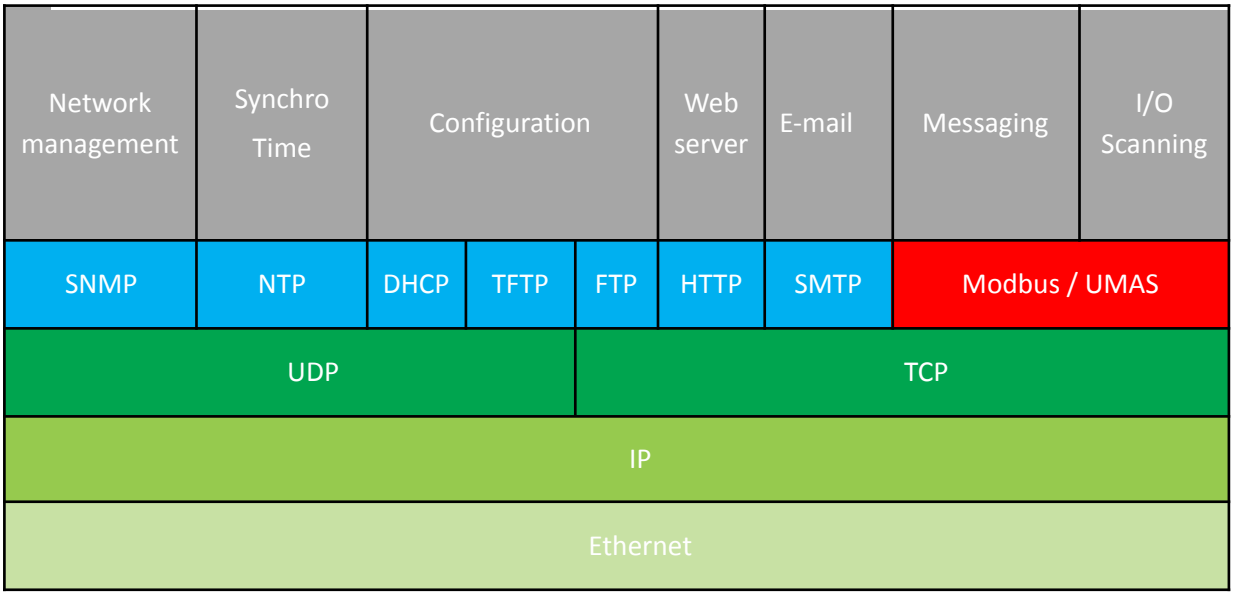
A TCP/IP packet can then be broken down using these same layers, and each part of the packet will be analysed sequentially from the lowest layer to the highest.

As a result, an attack on the lowest layers will potentially have more impact: for example, a possible attack on the network layer would be to usurp the IP address of a trusted machine (IP spoofing), and in this case the attacker's machine would have all the authorisations granted to the trusted machine, particularly in the higher layers. (Note that there are possible protections to avoid IP spoofing, which are notably implemented on the SNi40).

Thus, protection on a lower layer will (in general) be more effective than protection on a higher layer. However, as stated in the question, cybersecurity must be implemented at all levels, so it is necessary to establish as much protection as possible at all layers.

**7. Application Protection: Modbus / UMAS**

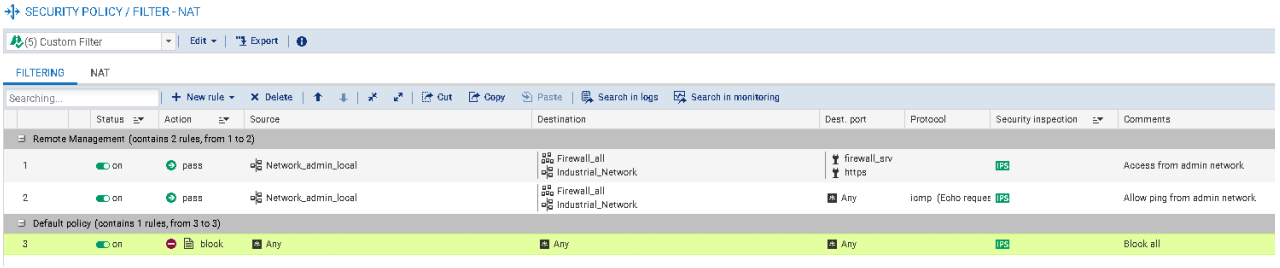


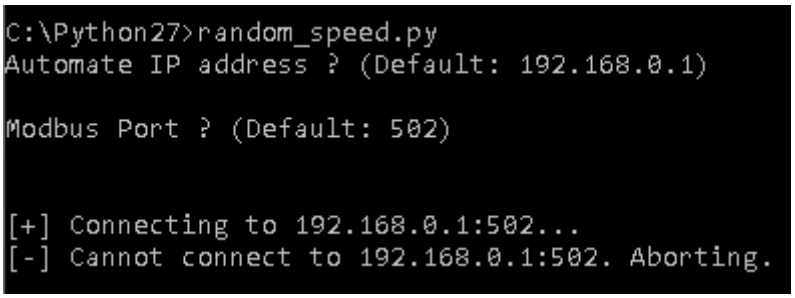

An application protection on the Modbus / UMAS protocol takes place at the Application layer, on the Modbus / UMAS protocol.

A first possible application protection is to completely disable the Modbus / UMAS protocol on the network. To do this, go to Configuration > Security Policy > Filtering and NAT, then check that the security policy is set to (5) Custom Filter. In the first rule, delete the Modbus protocol in the

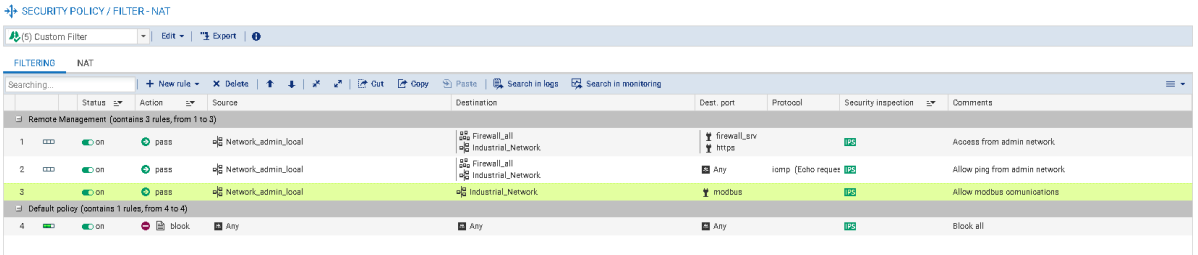
Destination Port column, then save and apply the new security policy:



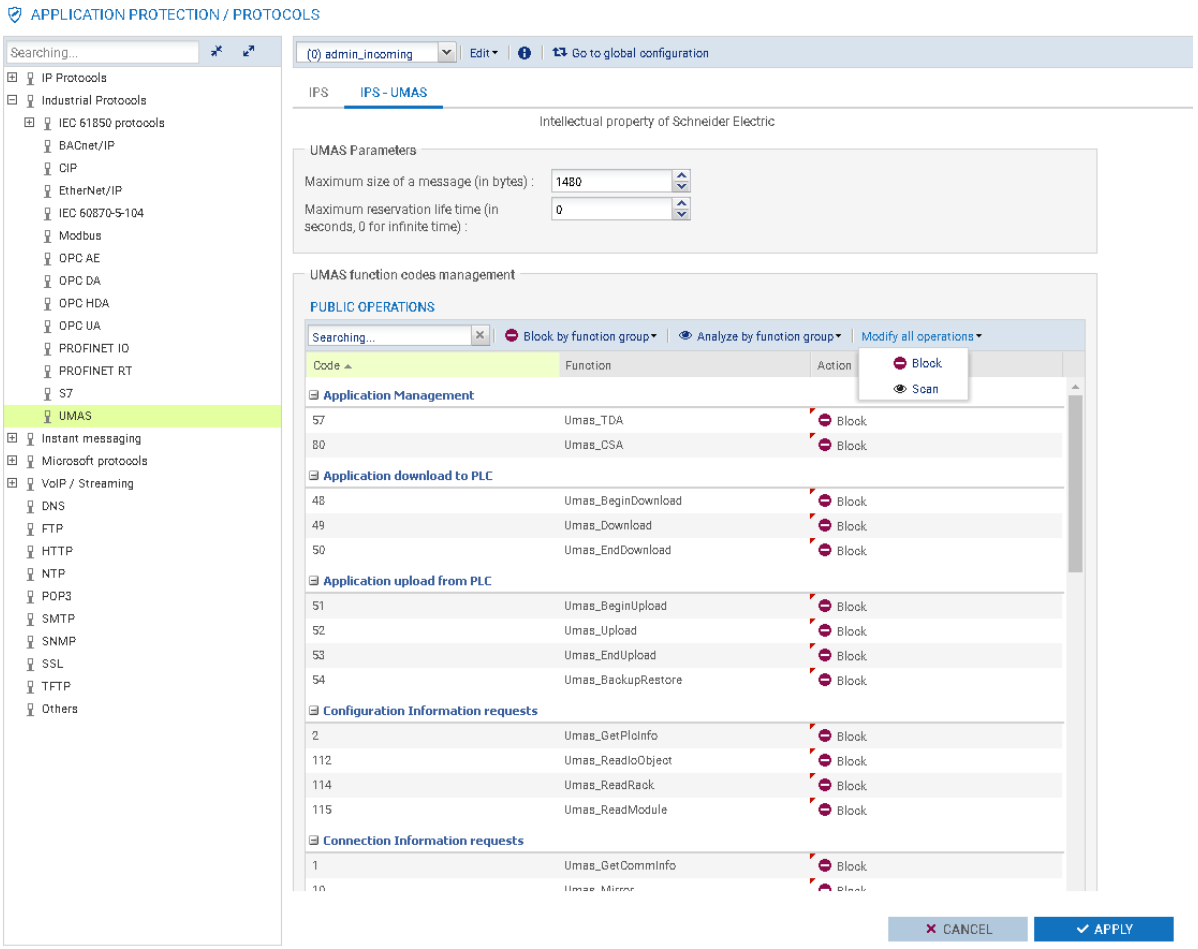

Then, at the command line, run one of the attack scripts, for example "random_speed.py". After a few seconds, you will get a result similar to the following:




This protection is very radical, and given that we wish to use Modbus / UMAS to control the industrial network, it is not interesting. In the following, we will restore the modbus protocol with a new rule:




If we look at the attack file "**m580_read_memory.py**" or "**m580_dos.py**", we can see that these attacks exploit the Modbus function **code 0x5a (90)** corresponding to the UMAS extension: Schneider-Electric's proprietary protocol. If you do not wish to use this protocol, you can go to **Configuration > Application Protection > Protocols**, and the **Industrial Protocols > UMAS** tab. In the "**Public Operations**" section of this tab, to the right of "**Analyse by function group**", there is a chevron to access the "**Modify all operations**" option, which can be used to block all operations:

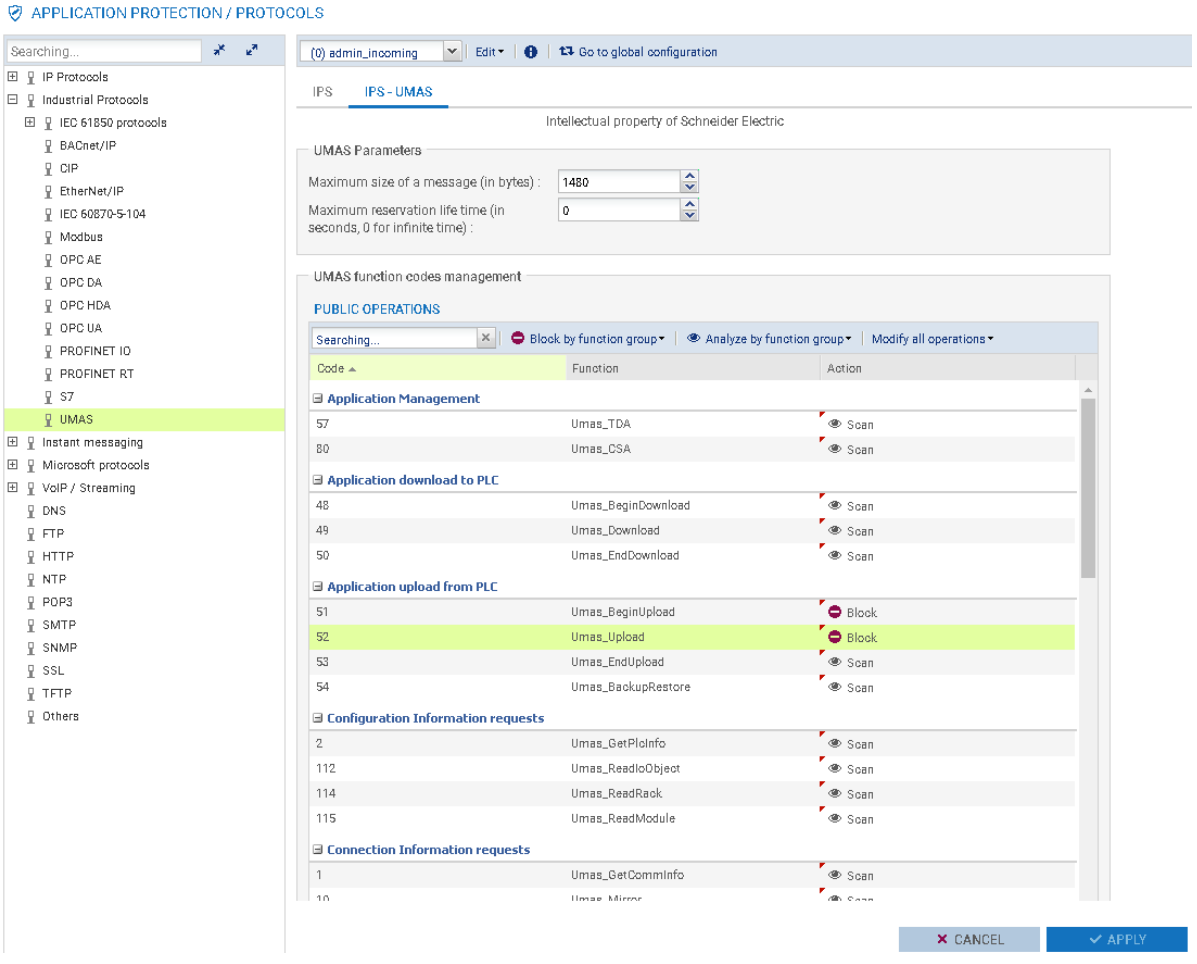


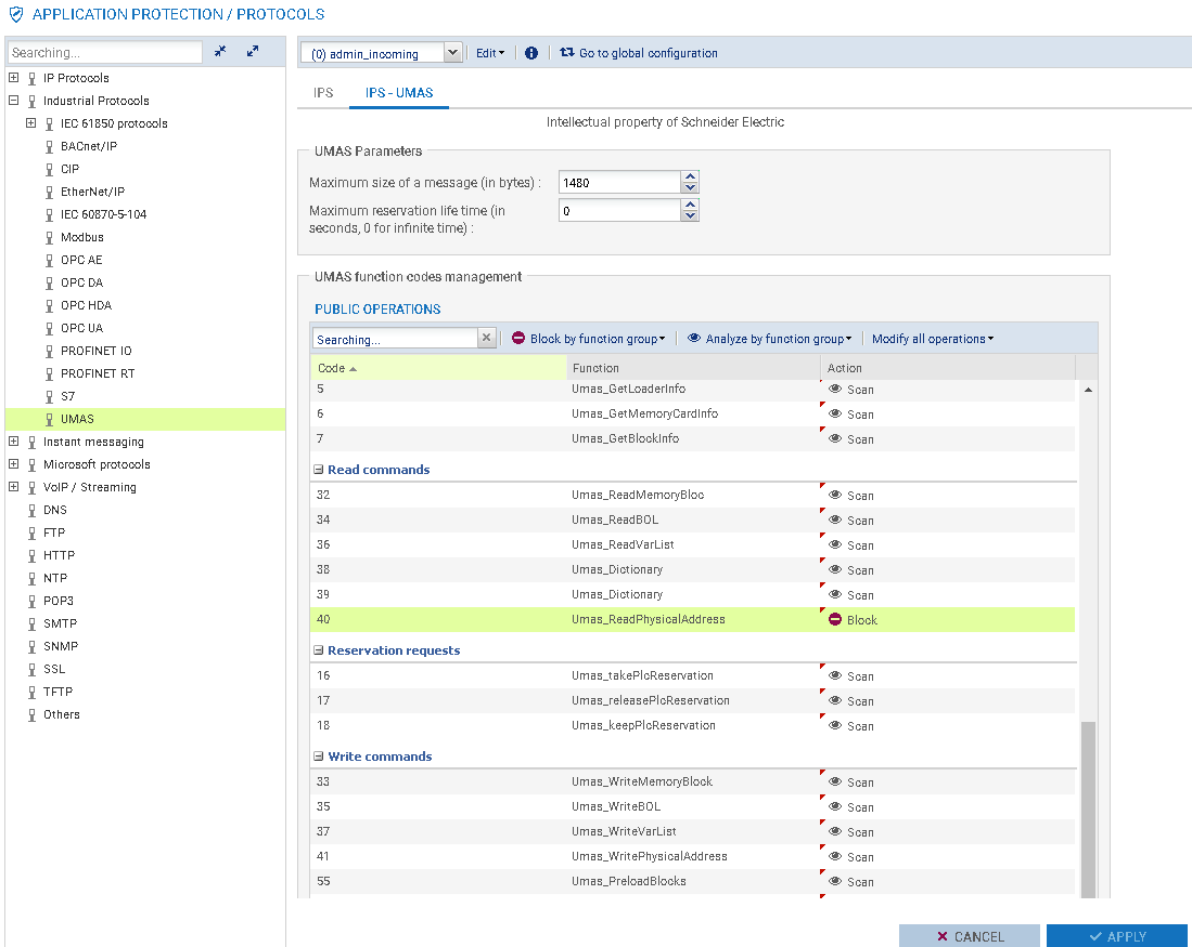

We can then apply the protection, and by running one of the two attack scripts, we get the following result:

```
C:\Python27>m580_dos.py
Automate IP address ? (Default: 192.168.0.1)

Modbus Port ? (Default: 502)


[+] Connecting to 192.168.0.1:502...
[-] Cannot connect to 192.168.0.1:502. Aborting.
```

The attack fails, as the packet is not transmitted, and we see that the automaton is still functional.

Nevertheless, it may be necessary to use the UMAS protocol, or at least part of it. Reset the UMAS protocol permissions by the same procedure by selecting "**Analyse**". Then, as you can see, the "**m580_read_memory.py**" script uses the UMAS function **codes 0x33 (51)** and **0x34 (52)** to download the PLC strategy program and retrieve the sensitive information. The "**m580_dos.py**" program uses the UMAS function **code 0x28 (40)** to read a physical address in the PLC memory. A possible protection (in case these function codes are not used) would be to block only these function codes:






Thus, when running the two attack scripts again, they fail:

```
C:\Python27>m580_dos.py
Automate IP address ? (Default: 192.168.0.1)

Modbus Port ? (Default: 502)


[+] Connecting to 192.168.0.1:502...
[-] Cannot connect to 192.168.0.1:502. Aborting.

C:\Python27>m580_dos.py
Automate IP address ? (Default: 192.168.0.1)

Modbus Port ? (Default: 502)


[+] Connecting to 192.168.0.1:502...
[+] Connected to 192.168.0.1:502.
[+] Setting screen to 404...
[+] Screen set to 404.
[+] Sending DOS payload...
[+] DOS payload sent.
[+] Checking M580 availability...
[-] M580 still available, DOS attack failed.
[+] Setting screen to 1...
Traceback (most recent call last):
  File "C:\Python27\m580_dos.py", line 104, in <module>
    set_screen(conn, 1) # write 1 to not display Hacking page
  File "C:\Python27\m580_dos.py", line 68, in set_screen
    sock.sendall(b'\x00\x00\x00\x00\x00\x06\x01\x06\x00\x12' + to_bytes(screen, 2))
  File "C:\Python27\lib\socket.py", line 228, in meth
    return getattr(self._sock,name)(*args)
socket.error: [Errno 10054] An existing connection was forcibly closed by the remote host
```

**FOTO**

It can also be seen from the SNi40 dashboard that these blocked packets are notified:

**FOTO**

If these function codes are needed, then it is possible to proceed differently. First of all, restore the authorisations of these function codes by setting them to "Analyse". Go to Configuration > Application Protection > Applications and Protections and in the "Search" field, enter "modbus":

**FOTO**

We should then see a number of application protection signatures, including one named CVE-2018-7853 which blocks the "m580_dos.py" attack (this code can be obtained in the flaw report referenced in the source header of the attack file).

N.B.: The signatures must be up to date in order to have access to these signatures. Moreover, at the time of writing this tutorial, no signature allowing to block the "m580_read_memory.py" attack is available, but we could possibly look for a signature named CVE-2018-7848, which if it exists concerns the attack in question.

Set this signature to "Prohibit", then apply the changes. When running the "m580_dos.py" script, we see that it fails:

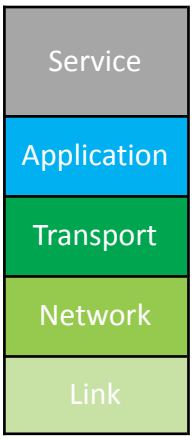
**FOTO**

On the SNi40 dashboard, the blocking of the package was also well notified:

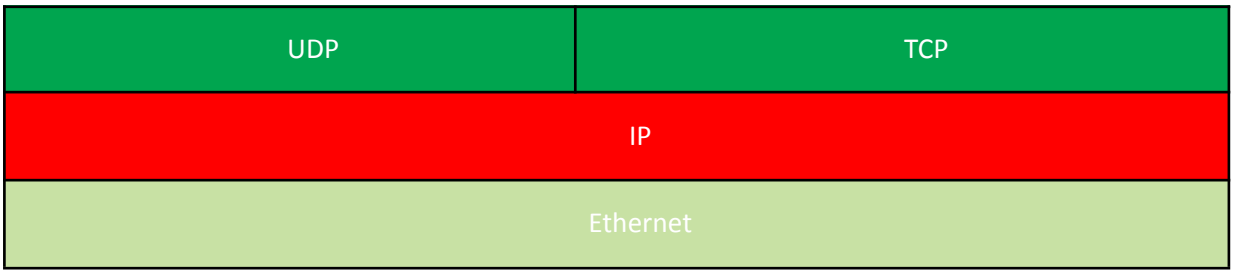
**FOTO**

**N.B.:** One could also define custom signatures as seen in another tutorial. However, to do so, the attack must be analysed in detail (or a vulnerability report must be used), which can be complex and time-consuming: this will therefore not be done in this tutorial.

## 8. Network protection: IP filtering




| Network management | Synchro Time | Configuration | | | Web server | E-mail | Messaging | I/O Scanning |
|---|---|---|---|---|---|---|---|---|
| SNMP | NTP | DHCP | TFTP | FTP | HTTP | SMTP | Modbus / UMAS | |




Protection by IP filtering takes place at the Network layer, on the IP protocol.

In order to set up IP address filtering, we will first define the administrator machine within the network in the SNi40 configuration. To do this, go to Configuration > Objects > Network objects. In this section, add a new machine named "PC_Admin" and specify the IP address of the machine 172.16.112.200 :

**FOTO**

Then go to the Configuration > Security Policy > Filtering and NAT section and change the source of the previously created rule regarding modbus requests to "PC_Admin", then save and apply the new policy:

**FOTO**

Using Vijeo Designer, we can then check that we still have access to the equipment:

**FOTO**

Change its IP configuration in order to obtain the IP address 172.16.112.201 :

**FOTO**

From now on, if one of the attack scripts is launched, communication to the equipment will be refused after a few moments, showing the effectiveness of the protection:

**FOTO**

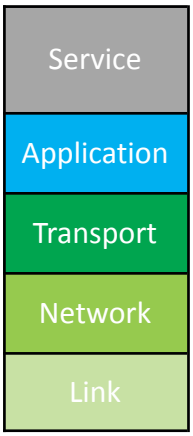
With the help of Vijeo Designer, one can also notice that the connection is also refused. As we have already seen, this protection is much more restrictive because it no longer concerns the modbus protocol of the Application layer but the IP protocol of the Network layer:

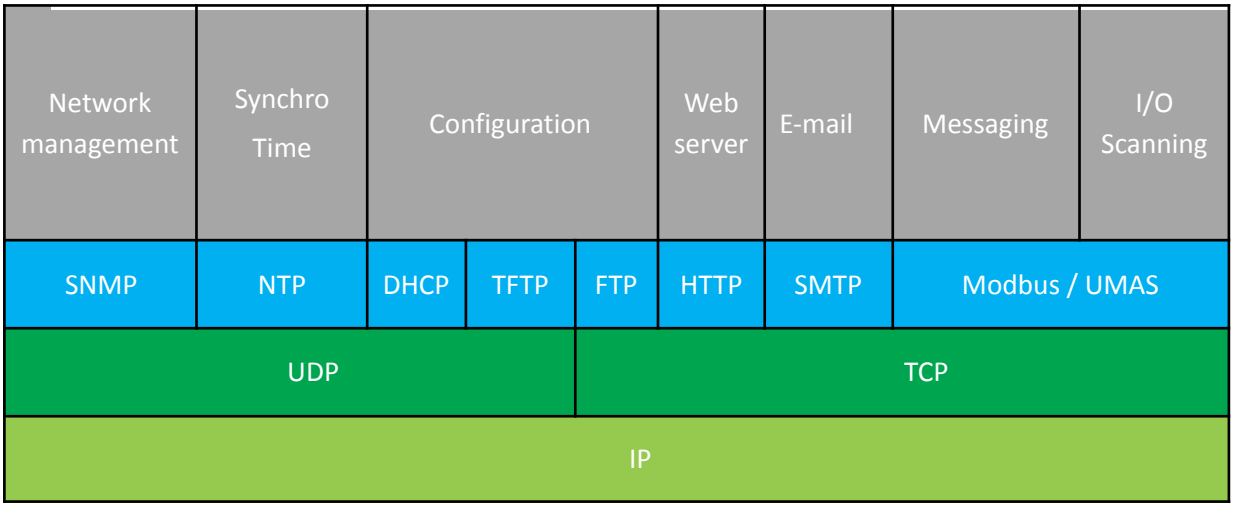
**FOTO**

Restore the old IP configuration for the rest of the course:

**FOTO**

### 9. Link protection: MAC filtering



# Ethernet

MAC filtering protection occurs at the Link layer, on the Ethernet protocol.

On the command line, use the command "ipconfig /all" to find the MAC address of your PC:

**FOTO**

Go to the Configuration > Objects > Network Objects tab and in the "Machines" object category, find the "PC_Admin" machine which should be at the end of the list.

Edit the machine to add its MAC address, then apply the changes:

**FOTO**

Check that the connection to the PLC is working properly using Vijeo Designer :

**FOTO**

There are several possibilities to test the protection in place:

- Use another PC (with a different MAC address) connected to the same port as the previous one, and with the same IP configuration.
- Change its MAC address on the interface through which the current PC is connected to the SNi40.

The second method will be used here to simplify handling and also to show that a MAC address, like an IP address, can be modified quite easily.

Go to the properties of your network interface (here Ethernet port), and click on "Configure..." :

**FOTO**

In the "Advanced" tab, change the "Locally administered address" property to the MAC address 12:34:56:78:9A:BC (without the colon):

**FOTO**

From the command line, still with the "ipconfig /all" command, we can check that the MAC address change has been recorded and we can also check that the routes are still well defined (especially for the 192.168.0.0/24 subnet):

**FOTO**

Now, using Vijeo Designer, we can see that the connection cannot be established with the :

**FOTO**

N.B.: Note also the loss of connection with the SNi40. Since the IP address 172.16.112.200 is used and is linked to the real MAC address of the PC, the SNi40 refuses any packet coming from the machine that does not have the right MAC address.

N.B.: Remember to remove the filtering rule and the "PC_Admin" machine from the firewall in order to remove the IP and MAC address filtering and to avoid any problems of access by other people that may result from this filtering.

**Important**: Through these manipulations, one will notice the simplicity with which one can change IP address or MAC address. These last two protections are therefore not unstoppable, hence the interest in applying maximum security at ALL levels. Furthermore, it should be noted that an attacker with the necessary knowledge will be able to create packets and define every bit of the packet. Attacks that impersonate another machine, by taking its IP, MAC address, or any other identifying information on a network, are called spoofing attacks. However, there are strategies to avoid these attacks, some of which are available in the firewall.