

# Analysis and deployment of a Network Access Control wired solution in an enterprise environment

AUTHOR: MARCO BASSO  
SUPERVISOR: PROF. ANDREA ZANELLA  
CO-SUPERVISOR: GIULIO TRIANI  
CO-SUPERVISOR: ANDREA CAVAZZINI

UNIVERSITY OF PADOVA

DEPARTMENT OF INFORMATION ENGINEERING

MASTER DEGREE IN  
INFORMATION AND COMMUNICATION TECHNOLOGY  
FOR INTERNET AND MULTIMEDIA -  
TELECOMMUNICATIONS

OCTOBER 5, 2022

ACADEMIC YEAR 2021/2022





## **Acknowledgement**

I would like to thank my family, Paolo and Maurizia, for pushing and supporting me during my studies.

I would like to thank Prof. Andrea Zanella for supporting and giving me the opportunity to develop this master's degree thesis.

Thanks also to Raffaele, Roberto, Giulio, Andrea, Albino, Paolo and Matteo for always helping me throughout the internship period in which I developed this project. Also thanks to all InfoCamere colleagues for always helping me with professionalism.

Last but not least, I would like to thank all the professors and friends I met during this wonderful journey that was the university.



# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Network introduction</b>	<b>3</b>
2.1	Internet protocol suite . . . . .	3
2.2	L2 switching and L3 routing concepts . . . . .	7
2.3	Network devices . . . . .	8
2.4	Network architectures . . . . .	9
2.5	Network security background . . . . .	10
2.5.1	Network security threats . . . . .	10
2.5.2	Threat actor tools . . . . .	11
2.5.3	Endpoint security attacks . . . . .	12
2.6	Security tools . . . . .	13
2.6.1	Security devices . . . . .	13
2.6.2	Security configurations . . . . .	15
<b>3</b>	<b>Survey on NAC technology</b>	<b>19</b>
3.1	NAC functionalities . . . . .	19
3.1.1	Node detection or Profiling . . . . .	20
3.1.2	Authentication . . . . .	21
3.1.3	Authorization . . . . .	22
3.1.4	Posture . . . . .	23
3.1.5	Quarantine and Remediation . . . . .	24
3.2	NAC architecture . . . . .	25
3.2.1	NAC Terminal . . . . .	26
3.2.2	NAD . . . . .	27
3.2.3	NAC Server . . . . .	27
3.3	Pre and Post admission . . . . .	27
3.4	In-line or Out-of-band . . . . .	28
<b>4</b>	<b>Case study</b>	<b>29</b>
4.1	Cisco ISE . . . . .	29
4.1.1	Personas . . . . .	31
4.1.2	Deployment architecture . . . . .	33
4.1.3	ISE GUI . . . . .	39
4.2	InfoCamere . . . . .	42

4.2.1	IC NAC solution . . . . .	44
4.3	Wireshark software . . . . .	47
<b>5</b>	<b>AAA applied to NAC</b>	<b>49</b>
5.1	Authentication . . . . .	50
5.1.1	RADIUS . . . . .	50
5.1.2	TACACS+ . . . . .	57
5.1.3	802.1X . . . . .	60
5.1.4	MAB . . . . .	66
5.1.5	EAP . . . . .	68
5.1.6	TLS tunnel . . . . .	80
5.1.7	Authentication flow . . . . .	85
5.2	Authorization . . . . .	99
5.3	Accounting . . . . .	102
5.4	AAA service set-up . . . . .	102
5.4.1	ISE set-up . . . . .	103
5.4.2	NAD set-up . . . . .	108
5.4.3	Supplicant set-up . . . . .	110
<b>6</b>	<b>Profiling</b>	<b>113</b>
6.1	CoA . . . . .	116
6.1.1	CoA messages . . . . .	116
6.2	Probes . . . . .	119
6.2.1	RADIUS . . . . .	122
6.2.2	DHCP . . . . .	127
6.2.3	HTTP . . . . .	129
6.2.4	SNMP . . . . .	130
6.3	Profiling policies . . . . .	135
6.4	Profiling service set-up . . . . .	137
6.4.1	ISE set-up . . . . .	137
6.4.2	NAD set-up . . . . .	140
6.5	Profiling related problem . . . . .	141
<b>7</b>	<b>Posture</b>	<b>143</b>
7.1	Posture architecture . . . . .	143
7.1.1	Temporal Agent . . . . .	143
7.1.2	Agentless . . . . .	144
7.1.3	AnyConnect Agent . . . . .	145
7.1.4	Stealth AnyConnect Agent . . . . .	146
7.2	ISE posture flow . . . . .	147
7.2.1	Posture conditions . . . . .	148
7.2.2	Posture remediations . . . . .	150
7.2.3	Posture requirements . . . . .	151
7.2.4	Posture policy . . . . .	152
7.2.5	Client provisioning . . . . .	153
7.2.6	Access policy . . . . .	156

---

7.3	Posture service set-up . . . . .	158
7.3.1	NAD set-up . . . . .	158
7.3.2	ISE set-up . . . . .	159
<b>8</b>	<b>Solution analysis</b>	<b>163</b>
8.1	Short-list comparison . . . . .	164
8.2	InfoCamere case study analysis . . . . .	167
8.2.1	Value, effort and risk analyses . . . . .	167
8.2.2	Feasibility analysis . . . . .	170
8.2.3	Future work . . . . .	171
8.3	Conclusion . . . . .	176



# List of Figures

2.1	ISO/OSI (left) and TCP/IP (right) protocols suite. . . . .	5
2.2	Encapsulation process. . . . .	6
3.1	NAC flow. . . . .	20
3.2	Assessment types. . . . .	24
3.3	NAC architecture. . . . .	26
4.1	Standalone deployment. . . . .	34
4.2	Cisco ISE deployment in a distributed architecture. . . . .	35
4.3	Small deployment scenario. . . . .	37
4.4	Maximum RADIUS scaling for SNS 3500/3600 series. . . . .	38
4.5	Medium deployment scenario. . . . .	38
4.6	Large deployment scenario. . . . .	39
4.7	ISE GUI. . . . .	40
4.8	Policy set section. . . . .	41
4.9	Authentication and Authorization policy of a specific set. . . . .	41
4.10	Profiling interface. . . . .	42
4.11	Radius logging interface. . . . .	42
4.12	Detailed radius logging interface. . . . .	43
4.13	ClearPass interface. . . . .	43
4.14	InfoCamere's logo. . . . .	44
4.15	NAC architecture in test environment. . . . .	46
4.16	NAC architecture in production environment. . . . .	46
4.17	Virtualized ISE node. . . . .	47
4.18	Wireshark's interface. . . . .	48
5.1	RADIUS centralized architecture. . . . .	50
5.2	Authentication method. . . . .	51
5.3	RADIUS packet. . . . .	51
5.4	Radius AVP. . . . .	54
5.5	AVP/TLV fields in a RADIUS packet. . . . .	54
5.6	Encapsulation of a RADIUS message. . . . .	54
5.7	RADIUS and TACACS+ flow comparison. . . . .	58
5.8	TACACS+ packet. . . . .	59
5.9	TACACS+ authentication process. . . . .	60
5.10	Controlled and uncontrolled port states. . . . .	61

---

5.11	Effect of MAC states. . . . .	62
5.12	802.1X PDU. . . . .	63
5.13	EAPoL packet captured in IC scenario. . . . .	64
5.14	Encapsulation of a 802.1X message. . . . .	64
5.15	PAE MAC destination address example. . . . .	64
5.16	PAE MAC destination address in test environment. . . . .	65
5.17	Service-Type based on the vendor. . . . .	66
5.18	MAB authentication process concept. . . . .	67
5.19	MAB authentication process in a real scenario. . . . .	68
5.20	Authentication flowchart. . . . .	69
5.21	Components of 802.1X. . . . .	69
5.22	EAPoR packet captured in IC context. . . . .	71
5.23	Components of 802.1X with EAPoR. . . . .	71
5.24	EAP packet. . . . .	71
5.25	802.1X authentication process at glance. . . . .	73
5.26	EAP-MS-CHAPv2 packet. . . . .	77
5.27	EAP tunneled method. . . . .	78
5.28	EAP tunneled method encapsulation. . . . .	78
5.29	EAP tunneled method encapsulation in IC real scenario. . . . .	78
5.30	Cipher suite example. . . . .	82
5.31	TLS 1.2 initial handshake. . . . .	84
5.32	TCP three-way handshake and TLS 1.2 initial handshake. . . . .	85
5.33	Packet capture of TLS 1.2 initial handshake. . . . .	85
5.34	IC GPO template. . . . .	86
5.35	Machine authentication report example. . . . .	87
5.36	EAP-MS-CHAPv2 packet. . . . .	90
5.37	EAP-MS-CHAPv2 session keys. . . . .	91
5.38	Authentication process in IC scenario. . . . .	92
5.39	RADIUS termination session message. . . . .	93
5.40	Traffic monitoring architecture in test environment. . . . .	95
5.41	IP phone configuration download sample. . . . .	96
5.42	EAP-MD5 authentication flow for IP phone. . . . .	97
5.43	IP phone authentication flow. . . . .	97
5.44	NAK packet example. . . . .	98
5.45	Authorization policy example. . . . .	99
5.46	RADIUS Access-Accept packet including dACL and VLAN asso- ciation. . . . .	100
5.47	RADIUS Accounting-Request packet including dACL request in- formation. . . . .	101
5.48	RADIUS Access-Accept packet including ACEs. . . . .	102
5.49	Authorization profile download in ISE interface. . . . .	102
5.50	Network device configuration interface. . . . .	103
5.51	AAA flow between policies conditions. . . . .	104
5.52	Policy sets configuration. . . . .	104
5.53	802.1X authentication policies. . . . .	105



---

5.54	MAB authentication policies. . . . .	105
5.55	Authorization profile interface. . . . .	106
5.56	Example of policy conditions. . . . .	106
5.57	802.1X authorization policies. . . . .	107
5.58	MAB authorization policies. . . . .	107
5.59	Windows 802.1X supplicant configuration. . . . .	111
5.60	Windows 802.1X supplicant configuration. . . . .	111
5.61	Windows 802.1X supplicant configuration. . . . .	112
6.1	ISE node settings. . . . .	114
6.2	Profiling policies interface. . . . .	114
6.3	CoA work flow. . . . .	120
6.4	CoA-Request packet sample. . . . .	120
6.5	Profiling concept flow. . . . .	121
6.6	Profiling configuration interface. . . . .	121
6.7	Profiling probes configuration. . . . .	122
6.8	ISE RADIUS probe settings. . . . .	123
6.9	Device sensor architecture. . . . .	124
6.10	Endpoint attributes collected. . . . .	126
6.11	Wrong packet content. . . . .	126
6.12	Correct packet content. . . . .	127
6.13	General SNMP MIB tree structure. . . . .	132
6.14	ISE SNMP probe settings. . . . .	132
6.15	ISE SNMPTRAP probe settings. . . . .	133
6.16	ISE SNMPQUERY probe settings. . . . .	135
6.17	Example of profiling policy. . . . .	136
6.18	802.1X profiling policy set. . . . .	138
6.19	MAB profiling policy set. . . . .	138
6.20	Yealink device profiling policy set. . . . .	139
6.21	Yealink device profiling conditions. . . . .	139
6.22	Yealink device profiling flow. . . . .	140
6.23	AAA and profiling flow for Polycom-VVX201. . . . .	142
7.1	AnyConnect interface. . . . .	146
7.2	Generic posture assessment flow. . . . .	147
7.3	Posture process flow of a compliant device. . . . .	147
7.4	Posture configuration flow through the ISE GUI. . . . .	148
7.5	Posture update interface. . . . .	148
7.6	Posture conditions interface. . . . .	149
7.7	Custom firewall condition configuration interface. . . . .	150
7.8	Custom firewall remediation configuration interface. . . . .	151
7.9	Custom patch management remediation configuration interface. . . . .	152
7.10	Posture requirements interface. . . . .	152
7.11	Posture policy interface. . . . .	153
7.12	Cisco AnyConnect download page. . . . .	154
7.13	Client provisioning resources interface. . . . .	154

7.14	Posture profile interface. . . . .	155
7.15	Posture AnyConnect configuration interface. . . . .	156
7.16	Client provisioning policy interface. . . . .	156
7.17	AnyConnect files installed within the workstation. . . . .	156
7.18	Posture assessment flow. . . . .	158
7.19	Web redirection for remediation authorization profiles. . . . .	160
7.20	RADIUS AVP pair for posture redirection. . . . .	161
7.21	Authorization policy for posture assessment. . . . .	162
7.22	Posture assessment flow with remediation. . . . .	162
8.1	Final implementation schema. . . . .	163
8.2	DTLS configuration interface. . . . .	172
8.3	EAP-TEAP configuration in ISE. . . . .	173
8.4	EAP-TEAP configuration within policy set. . . . .	174
8.5	Private VLAN diagram. . . . .	175

# List of Tables

2.1	Protocol type. . . . .	4
4.1	ISE deployment terminology. . . . .	30
5.1	Main RADIUS attributes. . . . .	57
6.1	CoA VSA examples. . . . .	118
6.2	Main RADIUS attributes. . . . .	128
8.1	NAC comparison between Cisco ISE and Aruba ClearPass. . . . .	166



# Chapter 1

## Introduction

The topic of IT security is nowadays a fundamental aspect when the time comes to design the IT infrastructure in an enterprise environment. Cyberattacks, the number of which is constantly growing, are aimed at exploiting unknown or not yet fixed vulnerabilities in order to steal or damage sensitive data. A very important aspect related to cyberattacks is the fact that they can be initialized both from outside the defense perimeter and from inside. There is therefore a need to restrict and limit access to one's own network. This issue acquires considerable relevance especially in those business environments in which the IT infrastructure has a large-scale dimension. In fact, environments of this size will host a large number of resources, data and services whose access must be guaranteed only to those who are authenticated and authorized. The reality of InfoCamere falls into this scenario.

To overcome this problem, various technologies have been presented on the market that allow the definition of a defense perimeter, not only towards the outside (and therefore everything that is not within the company environment) but also towards the internal. The solution called Network Access Control (NAC) has the purpose of restricting access to the network only to authenticated and authorized persons. After a study of the protocols used, the NAC solution currently implemented in the company (Cisco ISE) is presented. The main features that distinguish NAC are therefore analyzed, namely AAA, profiling and postures. While the former authenticates users, the second one determines which device the user attempts to log in with. Furthermore the state of health of the device is determined by means of the posture function. The information determined by this 3 features can be used jointly to define the access authorization level of a user/device. Subsequently, a comparison is reported between the 2 most adopted solutions in the enterprise environment, i.e., Cisco ISE and Aruba ClearPass.

Once this (theoretical) comparison has been made, the feasibility indices for the migration of the designed solution into the production environment of InfoCamere are analyzed. At the end of this analysis the results obtained are reported and discussed.

In the last part of this work, new features are described and proposed to improve the services illustrated. The proposals are also accompanied by brief

comments on their feasibility of implementation.

# Chapter 2

## Network introduction

In order to fully understand Network Access Control (**NAC**) technology (and the various standards on which it is based) it is necessary to briefly introduce the basic networking concepts. Therefore, starting from the description of the protocol suite in Section 2.1, the routing concepts are then reported in Section 2.2. The various types of network devices that make up a generic IT architecture are then presented in Section 2.3. The latter assume an important role when defining NAC architecture.

### 2.1 Internet protocol suite

To be able to communicate over a network, each device must abide by the same set of rules. These rules are known as protocols, and they have many functions in a network. Network protocols define a common format and set of rules for exchanging messages between devices. Protocols are implemented both by intermediary and end devices in software, hardware, or both. Each network protocol has its own function, format and rules for communications. The different types of network protocols are listed in Table 2.1).

In many cases, protocols must be able to work with other protocols so that the online experience gives the user everything he needs for network communications. Protocol suites are designed to work with each other seamlessly. A protocol suite is a group of inter-related protocols necessary to perform a communication function. One of the best ways to visualize how the protocols within a suite interact is to consider the interaction as a stack. A protocol stack shows how the individual protocols within a suite are implemented. Protocols are displayed in terms of layers, where the lower one provide services to higher layers. The lower layers of the stack are concerned with moving data over the network and providing services to the upper layers, which are focused on the content of the message being sent. Specific functions are therefore associated with each layer. Using a model implemented as a stack divided into layers makes the modification of a protocol belonging to a certain layer much more streamlined and intuitive. The first model defined between the 70s and 80s by the International Organization for Standardization (**ISO**) was the Open Systems Interconnection (**OSI**)

Protocol type	Description
Network Communications Protocols	Protocols enable two or more devices to communicate over several networks. The Ethernet family of technologies involves a variety of protocols such as IP, Transmission Control Protocol ( <b>TCP</b> ), HyperText Transfer Protocol ( <b>HTTP</b> ), and many more.
Network Security Protocols	Protocols secure data to provide authentication, data integrity, and data encryption. Examples of secure protocols include Secure Shell ( <b>SSH</b> ), Secure Sockets Layer ( <b>SSL</b> ) and Transport Layer Security ( <b>TLS</b> ).
Routing Protocols	Protocols enable routers to exchange route information, compare path information, and select the best path to the destination network. Examples of routing protocols include Open Shortest Path First ( <b>OSPF</b> ) and Border Gateway Protocol ( <b>BGP</b> ).
Service Discovery Protocols	Protocols used for the automatic detection of devices or services. Examples of service discovery protocols include Dynamic Host Configuration Protocol ( <b>DHCP</b> ) which discovers services for IP address allocation, and Domain Name System ( <b>DNS</b> ) which is used to perform name-to-IP address translation.

Table 2.1: Protocol type.

model. This model defines 7 layers organized in a hierarchical structure. At the same time, the concepts of encapsulation, Protocol Data Unit (**PDU**), Protocol Control Information (**PCI**) and Service Data Unit (**SDU**) were defined. Since the '70s, there have been different protocol suites, developed both from standard organisation and different vendors. The Transmission Control Protocol/Internet Protocol (**TCP/IP**) suite, which is one of the several paradigm presented, represents nowadays a standard “*de facto*”. An important difference between the ISO/OSI suite and the TCP/IP suite is that the latter provides for the definition of 4 layers instead of 7. The comparison between the 2 stacks is shown in Figure 2.1).

As for the ISO/OSI suite, starting from the highest layer, there is the application layer whose task is to keep the process to process communication active [1]. This layer is closest to the final user and is made up of Application Programming Interface (**API**). It therefore provides application layer functions to the user. This layer includes DHCP, DNS, HTTP, File Transfer Protocol (**FTP**) and Simple Network Management Protocol (**SNMP**) services. The presentation



layer provides a representation of the data exchanged between the application layer services. The session layer provides services to the upper layer to initialize a communication channel and to manage data exchange. These 3 layers just defined are therefore more related to the final application rather than to the actual communication process. The transport layer defines services to segment, transfer, re-transfer and reassemble the data for communications between end devices. The protocols that distinguish this layer are TCP and User Datagram Protocol (**UDP**). In addition to these, there is also a third one created by Google that is being implemented more and more, namely QUIC. The main difference between TCP and UDP is that the former is connection-oriented, while the latter is connectionless. In fact, during a TCP connection, before the actual data exchange takes place, control information is exchanged. This first phase is called “*handshake*”. A connection is thus established over which data can then be transmitted. Furthermore, whenever the source sends data, the destination will reply with a special acknowledgment (**ACK**) message if it arrived correctly or with a **NACK** if it arrived corrupted (or didn't arrived). In this second case, a re-transmission by the source takes place. This layer is also entrusted with the function of controlling network congestion.

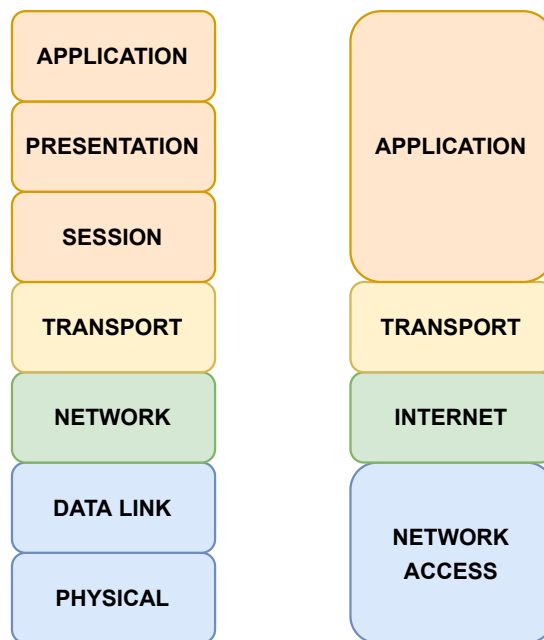


Figure 2.1: ISO/OSI (left) and TCP/IP (right) protocols suite.

The network layer provides services to exchange the individual pieces of data over the network between identified end devices. To be reached, all devices must have an associated identifier. This identifier corresponds to the IP address. It is a string formed by 4 octets (**IPv4**), with which it is possible to identify 4.294.967.296 nodes. Since the number of devices able to connect to the network gradually increases, the standards have moved on to another addressing method (**IPv6**) with which it is possible to define a much higher number of devices. An

important function that is provided at this layer is the routing service. Examples of routing protocols are BGP, OSPF, Routing Information Protocol (**RIP**), and Intermediate System to Intermediate System (**IS-IS**).

The remaining layers instead deal with the transmission on the physical medium. In particular, the data link layer is divided into 2 sub-layers: Medium Access Control (**MAC**) and Link Logical Control (**LLC**). The first one deals with the management of access to the transmission medium, while the second one deals with the control of errors occurred during transmission. At this layer another type of addressing is introduced, namely MAC address (composed by 6 bytes). This uniquely ID, identifies the network card installed inside the device. Unlike the IP address, this information is used to exchange messages locally or within the same network. Finally, the physical layer manages the actual transmission on the physical medium, whether it is wireless or wired (ethernet, optical fiber, etc.). As regards the TCP/IP model, some of the layers have been compressed into a single one. The 3 layers closest to the end user have been grouped into a single one called application layer, while the 2 layer related more to the management of physical data transmission have been grouped into a single layer called network access layer. The intermediate layers, from which the TCP/IP model takes its name, have remained unchanged [2].

When a data is generated at the application layer, in order to be correctly transmitted, it needs to be associated with control information. So, the SDU (payload) is associated with information contained in the PCI (header). Once these 2 parts are associated with each other, the layer  $N$  message is formed (i.e., the PDU). This PDU is then passed to layer  $N - 1$ , which will associate the control information relating to its layer ( $N - 1$ ) to it. This process is iterated up to layer 2. At layer 1, the message will then be transmitted through the transmission medium. The process just reported is presented in Figure 2.2). At the different layers, the PDU is associated with a different name. From a generic term (message) the literature pass to a more specific one. At layer 2 a message is equivalent to a frame. At layer 3 the term packet is used, while at layer 4 the term segment is used.

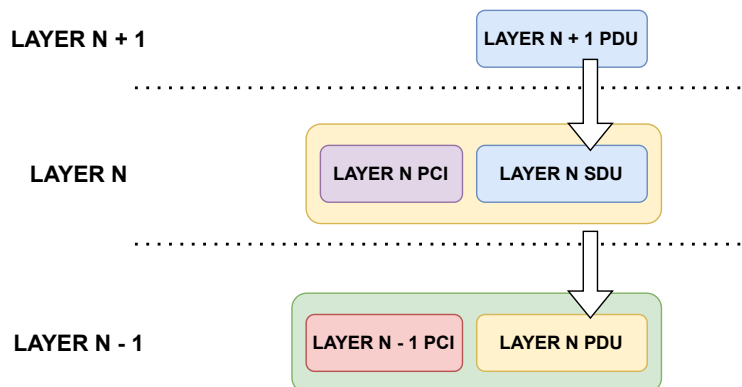


Figure 2.2: Encapsulation process.

## 2.2 L2 switching and L3 routing concepts

To ensure that the end devices can communicate, it is necessary to define protocols capable of directing messages from the source to the destination. The concept of routing is therefore born from this need. Message routing can take place in 2 different contexts: within the local network or outside.

If the sender and recipient of the message are within the same network (i.e., same network part relating to the IP address), it will be exchanged using the information contained within the frame (layer 2 information). The MAC addresses of the devices are therefore used. An example of a device capable of reading this information and capable of directing messages to the correct destination is the switch. This device and the other one will be discussed in Section 2.3. The message is then generated by the sender (PC, smartphone, printer, etc.) and is transmitted on the transmission medium in use. This message will be received by an intermediary device (e.g., switch) which, by reading the control information contained within the header (in this case relating to the frame), will forward the message to the output port where the recipient device is connected.

If, on the other hand, the sender and the recipient reside in different networks, it is necessary to use the control information contained within the packet (i.e., layer 3 information). In this scenario, L3 device (e.g., router), based on its routing table, will redirect the message to the correct destination. In both cases, in order to access control information, network devices must apply the concept of encapsulating and decapsulating. For example, in order to read the source and destination IP addresses, the router must first extract the packet from the entire frame. Once the information has been read, it is up to him to correctly recreate the original message by wrapping the packet within the entire frame.

To choose which interface to send the message on, both layer 2 and 3 devices use auxiliary tables. For layer 2 devices, these tables are called “*forwarding tables*”. Within these one, entries formed by the Interface ID/MAC address association are inserted. One or more layer 2 addresses can correspond to each interface of the device. When the switch has to decide on which port to send the message, it compares the destination MAC address with those present in the table. Once the match between the 2 addresses has occurred, the message is sent on the port associated with it. If, on the other hand, no match occurs, the message is sent to all ports, except the one from which it came. The message sent in this scenario is called “*unknown unicast*”.

For layer 3 devices, on the other hand, the auxiliary table used is called “*routing table*”. This table is similar to the forwarding one, with the substantial difference that there are IP addresses rather than MAC addresses stored within it. The routing table can be updated in 2 different ways: statically or dynamically. As for the first, the network administrator, after connecting to the device management interface, insert a static route defined by himself. This route cannot be changed in the future, except by the operator himself. The second way instead is linked to all those protocols that define dynamic routing. Examples of these have been mentioned above (e.g., RIP, OSPF, etc.). The layer 3 devices, through

complex algorithms, define the routes based on the information exchanged with the other layer 3 devices. This implementation makes it possible to address any problems that may arise within the infrastructure (e.g., a link failure). However, such implementations must be configured by an operator, and unlike static routing, they require much higher computational power. The computational power required therefore increases exponentially with respect to the number of nodes that make up the network.

## 2.3 Network devices

To satisfy the high and diversified number of functions to be implemented in the networking world, it is necessary to use multiple devices. Commonly, each device is associated with the functions it guarantees and the layers at which it operates. Starting from the physical layer can be found devices such as hubs or repeaters. While the former are used in wired contexts, the latter are used for wireless connectivity. Although they operate with different transmission media, their function is common. In fact their main task is that of, given a signal as input, regenerate its physical characteristics (i.e., amplitude, power, etc.) and then transmit it towards an output interface. Nowadays this type of devices, without any computational power capacity, are no longer used. Regarding layer 2 devices can be found bridges and switches. As explained above, these types of devices are able to process the message up to layer 2 (frame). There are different types of switches that can be differentiated by capacity and number of interfaces installed. Also, nowadays, each port on a switch defines a broadcast domain. The most common devices of layer 3 are routers. It is important to mention that some types of switches have implemented layer 3 functionality. These are referred to as L3 switches. Routers allow connectivity between different networks. While switches are commonly referred to as access devices, meaning they have end devices connected to themselves, routers are edge devices. An end device that therefore wants to communicate with a device belonging to another network, must necessarily send the message towards the router (which in this case acts as the default gateway). Moving on to layer 4 can be found the Firewall (**FW**). They are able to inspect messages (segments) up to layer 4. An important example of application is that of controlling TCP connections between a private network and the Internet. A firewall is able to control the handshake phase and in case of anomalies, it is able to terminate this phase thus terminating the connection. As in the case of other network devices, there are many types of FW. One of these are the so called Next Generation Firewall (**NGFW**). The latter are able to inspect messages up to layer 7, and therefore understand from which application the message was generated. In the enterprise environment, these devices take on considerable importance, and in fact they are the devices that are installed along the perimeter of the campus network.

Furthermore, in the last 10 years, the concepts of virtualized devices have been introduced. Therefore, instead of having specific physical machines, general

purpose machines are used in which multiple virtual machines are installed that take on the roles of network devices.

## 2.4 Network architectures

The devices described in Section 2.3 can be arranged in multiple ways, giving rise to different network architectures. Below are reported the 2 types of architectures that are of interest for the case study presented. The most common is Local Area Network (**LAN**). This term identifies a local network created within a private context (e.g., company, college, etc.). This network allows communication between different devices located within the same environment. In a business reality, where the number of devices is high, the network is often divided into subnetworks (also known as subnet), in such a way as to logically segment the traffic, creating more broadcast domains. For example, if the company is divided into floors, it is common to associate a subnet, also called Virtual Local Area Network (**VLAN**), to each floor. Therefore, being separate networks, it is necessary to pass through a layer 3 device (e.g., router or L3 switch) to ensure that they can communicate. A type of local network, but extensive in the number of devices belonging to it, is also known as campus network.

A company that offers services to third parties, manages a particular area of the network called Data Center (**DC**). This area is the main definitive area for requests from users. Since it contains sensitive data, it's protected by devices such as Intrusion Prevention System (**IPS**), Intrusion Detection System (**IDS**), NGFW and anti-**DDOS** devices (Distributed Denial of Services). These devices offer a wide range of security services, including:

- SSL/TLS encrypted traffic inspection;
- Virus, malware and trojan scanner;
- Data Loss Prevention (**DLP**).

The network that interconnect several LANs distributed along a wide geographical area is defined as Wide Area Network (**WAN**). Covering a large geographic area, it is often referred to as a geographic network. Almost always, this type of network is managed by an Internet Service Provider (**ISP**). The company therefore stipulates contracts with one or more ISP (to guarantee redundant services), with which several remote offices are interconnected. So, when an employee belonging to a certain remote office wants to access a service hosted at the headquarter, he will do nothing but send messages that will first pass over the WAN (in which the L3 devices will perform the routing action), to then arrive at the central campus network.

## 2.5 Network security background

This section provides a general overview of the subject of cybersecurity. The main methodologies of attacks are therefore presented with the respective tools used by malicious users.

### 2.5.1 Network security threats

The purpose for which a network is protected by security devices is to ensure robustness against cyberattacks. There are many types of tools and attacks with which a malicious user can threaten an infrastructure [3]. Some types of such attacks are described below.

**Eavesdropping Attack** This represents one of the first types of attacks ever used. This type of attack is also known as Man In The Middle Attack (**MITM**). In this scenario, the malicious user positions himself between a client (target) and another device (such as a server). Through simple tools it is able to intercept (or sniff) the traffic between the two network nodes without them noticing. These types of software tools, very often, are downloadable for free.

**Modification Attack** Exploiting the previous type of attack, the malicious user has the capability, not only to identify the type of traffic, but also to modify it. An example of the use of such an attack would be that in which a user requests generic information from a server. The attacker, by intercepting the message, modifies the request in such a way as to make it appear that the requested information is sensitive (e.g., user credential). At this point the server would respond, causing the credentials to be compromised.

**Compromised-Key Attack** A fundamental problem that the 2 types of previous attacks have in common is that of encryption. In fact, almost all communications today, before being established from the source, are somehow encrypted. This encryption (which is mentioned in Chapter 5) allows the 2 extreme nodes of the communication not to expose the data exchanged between them. It is however possible, always through simple software, to sniff the traffic to understand the type of data flow (but not the data encrypted). The encryption used can be of two types: symmetric or asymmetric. While in the first method, the 2 nodes use the same key to encrypt and decrypt data, in the second one each node has a pair of keys. This type of attack involves the subtraction, by a malicious user, of the secret key of one of the two nodes. This way the encrypted data can be easily decrypted. A stolen key is therefore defined as compromised key.

**Denial of Service Attack** With this attack, the hacker tries to make a service or resources in general unavailable by continuously sending messages to a certain node in the network. The latter in fact, receiving a large number of messages, runs out of resources to satisfy them, making the physical machine unavailable.

In this scenario, therefore, other users who wish to make legitimate requests cannot be satisfied. In the previous section, a particular type of machine was mentioned, namely the anti-DDOS device. A DDOS attack, being a distributed DOS, causes false requests to be performed not only by a single device but by a set of nodes. These nodes are controlled by a central entity, which at the respective start command, begin to flood the target.

### 2.5.2 Threat actor tools

To carry out cyberattacks, malicious users almost always need additional software [4]. Some of these are shown below.

**Network Scanning** These types of software allow the attacker to understand how the network infrastructure is implemented. In fact, before carrying out a possible attack, it is important to know which logical resources make up the target network. Some useful software are Nmap, SuperScan, and NetScanTools. These therefore have the purpose of sending probes to understand, for example, the addressing used and the services accessible by scanning the open ports. Often these methods fall under the type of reconnaissance attacks.

**Packet Sniffers** These software have the purpose of intercepting the traffic and therefore allow sniffing the conversations between 2 or more nodes. Some of the best known are Wireshark (later also used for NAC analyses), Tcpdump and Ettercap. In the case of encrypted traffic, through the implementation of particular profiles containing the compromised keys, it is possible to decrypt it, thus referring to the clear data.

**Packet Crafting Tools** If a malicious user want to make an attack towards an enterprise environment, he has to deal with the perimeter security systems implemented. One of these are the FW and NGFW. These software are therefore used, to understand if they have any vulnerabilities (perhaps caused by a solution not well implemented). By creating malicious packets, the attacker tries to understand if inside these machines there are any exploits to be used. It is important to remember how these software can also be used in the test phase, and therefore by the company itself, to test the effectiveness of the security perimeter implemented. Some examples of such software are Hping, Yersinia and Nemesis.

**Wireless Hacking Tools** In an enterprise environment it is commonly used to implement the network infrastructure also using wireless technology. In particular, the 2 most common types of networks within a similar environment are guests and intranets. While the former are subject to special policies and therefore users belonging to that type of network have limited access, the latter are logically equivalent to wired infrastructures. These too must therefore be adequately protected. Examples of these software are Firesheep and KisMAC.

### 2.5.3 Endpoint security attacks

While the previous sections focus on the types of attacks performed from the external point of view with respect to the logical infrastructure of the network, it is necessary to introduce some types of threats that make NAC necessary. Often the attacks with the greatest destructive potential are those carried out from within the company. For example, let's consider physical attacks. It is natural that a user who is inside the building can more easily cause damage to the infrastructure, for example by implementing malicious configurations within the network equipment [5]. Some examples of internal attack sources are reported below.

**Baiting** This source of attack is the one most commonly used still today. It consists of leaving an external storage unit (e.g., pen drive, hard disk, etc.) unattended in a public place (e.g., a library, internet point, etc.). A series of malicious software that are invisible at first glance are installed on this drive. It often happens that these are also invisible to any anti-malware installed within the device to which the storage unit is connected to. When this type of devices are then connected to the corporate network, they have the ability to expand across the entire network causing huge problems. At this point, in addition to making the network almost inaccessible, some of these software have the purpose of stealing data (whether sensitive or not) or to encrypt it. The company is therefore enforced to pay a ransom to obtain the keys from malicious users to decrypt previously compromised resources. In this scenario, the malware used is referred to as ransomware.

**Phishing and Pretexting** These represent 2 sources similar to those defined for baiting. In fact, for these scenarios, the user is deceived through requests for credentials (e.g., to log into their bank account). However, in addition to the theft of credentials, malware of any kind (e.g., viruses, spyware, trojans, worms, etc.) is installed on the user's device. The user therefore falls into the scenario described above.

**Non-compliant software** With the ever stronger confirmation of the Bring Your Own Device (**BYOD**) paradigm, each device represents a threat to the security of the internal network. Any personal device (e.g., smartphone, tablet, etc.) can therefore try to connect to the network, putting the availability of resources and services at risk. By downloading unverified applications and installing outdated software, attackers can cause extensive damage.

It is common to identify some types of attacks performed from within the company's network as Social Engineering Attacks.



## 2.6 Security tools

When implementing the security plan for a network architecture, one basic concept must be kept in mind: 100% protection against external (or internal) threats cannot be guaranteed. In fact, when the various solutions are implemented, it is necessary to take into account both the required needs and, above all, the cost constraints. There is therefore a trade off between user experience, security level and management costs. The safety concept can be implemented both in hardware and software. Examples of these implementations are briefly presented in the following chapters.

### 2.6.1 Security devices

As already briefly mentioned above, there are numerous devices whose function is to protect the network from malicious attacks. The most common of these are FW, IPS, anti-DDOS. Firewalls, as briefly mentioned in the previous chapters, are devices whose function is to filter the incoming and outgoing traffic from the network that they protect, based on the configured policies. Multiple protected areas can be implemented within a network, thus creating more protected areas also known as zones. The traffic between different zones is therefore checked before being sent to the respective destination. Generally the firewall can be divided according to the following 4 categories:

**Network layer firewall** Also called “*packet-filtering firewall*”, have the ability to filter packets based on the information contained within the layer 3 packet. It is the simplest and fastest type of filtering. Each type of firewall can be designed with different default policies. These can in fact be *allow-by-default*, and therefore let messages transit except those dropped, or *deny-by-default*, and therefore deny all traffic except that is expressly permitted. However, having limited functions, they have some drawbacks including the fact of not being able to detect attacks performed at higher layers (e.g., layer 4 or 7). Over time these firewalls have been enhanced by adding inspection functionality up to and including layer 4. In this way it is possible to filter traffic based on the type of service. These advanced versions are commonly referred to as stateful firewalls, as they are capable of monitoring the state of a session. For example, if the device receives a reply message to which no session is associated, it discards it.

**Application layer firewall** In addition to inspecting packets and segments, these types also analyze the content of application layer PDUs. Services such as Trivial File Transfer Protocol (**TFTP**), DNS, and HTTP can therefore be subject to network device scanning. Application-based attacks are constantly evolving, and for this reason, these types of firewalls are constantly updated on known techniques and types of threats. A special type of firewall that falls into this category is the Web Application Firewall (**WAF**). The latter specializes in deep analyses of HTTP/HTTPS messages. An examples of attacks that it can stop are SQL or

OS command injections. The biggest disadvantage and difference from the other types is the increased processing time for each packet. By having to analyze the packet in depth, the latencies of the connections are increased.

**Unified threat management** This term indicates a single network security appliance, which integrates multiple devices (e.g., FW, IDS and IPS). Instead of using a set of devices, this solution acts as a single point of security management and control. The main disadvantage, easily understood, is represented by the single point of failure. If in a generic architecture composed of several devices one were to stop working, the protection functions would be taken over by the others. In this case, however, the network would no longer be protected or in any case easily vulnerable. The great advantage, however, is represented by the fact of being a single node to manage. Unified and centralized management of all systems would greatly increase efficiency and productivity. Another fundamental advantage is the fact of having a single appliance from a single vendor. In an enterprise context where there are several solutions of different vendors, it represents an important facilitation since it is not necessary to study the interoperability (and the related problems) between the single devices.

**Circuit-level gateway** Unlike other firewalls, whose task is to analyze traffic while keeping the source and destination nodes unaltered, this one breaks the end-to-end model paradigm. In fact, from the single connection that would be established, 2 are created. The first between the sender and the firewall, while the second between the latter and the node inside the network. The layer on which it operates is therefore intermediate between network and application of the TCP/IP model. The protected nodes of the internal network are therefore somehow hidden by the presence of this gateway. In this case, however, the basic function mentioned above (i.e., packet inspection) is not present.

IPS are systems that are capable not only of detecting malicious actions, but also of managing the countermeasures to be used to mitigate such attack attempts. Usually these types of devices are inserted in cascade with anti-DDOS devices, firewall or IDS. They are positioned in-line, and therefore monitor network traffic flows in real time. The main difference with firewalls lies in the type of control. In fact, let assume that an attacker is attempting to send malicious packets to port 80 of a web server. For the firewall policies these messages cannot be blocked, because the destination port is considered legitimate. Messages at this point could easily reach the internal machine. The IPS instead detects these attempts and acts accordingly. In other words, the main difference between them lies in the inspection function. While the firewall performs a per-packet inspection, the IPS is able to correlate the various packets over time, identifying anomalous situations. The IPS is therefore able to identify and block, for example, reconnaissance attacks in which multiple messages are received over a period of time. IPS can be divided into 4 categories:

- Network IPS (**NIPS**): device installed in a strategic point of the network

so as to provide total coverage. In enterprise environments it is generally positioned in cascade with an anti-DDOS device;

- Host IPS (**HIPS**): unlike the previous case, this software is installed within the end devices. When integrated with the NIPS, it provides a high degree of protection. In the event that the IPS, and other protection nodes, are bypassed, the HIPS is the last protection node;
- Network behavior analysis (**NBA**): device whose purpose is to identify and block abnormal network flows. Module generally implemented in other types of IPS;
- Wireless IPS (**WIPS**): device implemented within a wireless infrastructure to mitigate wireless attack attempts.

Another very important type of protection devices, especially in environments that act as Service Provider (**SP**), are the anti-DDOS devices. These devices base their operation on the concept of the traffic threshold. These are devices of great computational power, able to manage huge data flows, without affecting the services they protects. When configuring devices of this type, the network administrator must specify which thresholds to use to enable the functionality of the device. It is also possible to configure some rules, specifying the bandwidth limit used for each type of traffic. In the event that the device detects an excessive flow (bps) coming from the outside, it is able to route these messages to a quarantine area where they will be discarded. Through the management consoles and GUIs it is also possible to constantly monitor the origin of these malicious flows. Generally, together with the computational powers of physical devices, it is possible to take advantage of cloud services. In these cases the traffic is then directed towards these services, which make ad-hoc anti-DDOS machines available.

### 2.6.2 Security configurations

In addition to the devices designed to guarantee the security of the infrastructure, it is possible to implement security configurations directly on the devices that manage the flow of traffic (switches, routers, etc.). One of the most used feature for security management is Access Control List (**ACL**). This is essentially a list in which are defined all types of traffic flows that are allowed or not. An ACL is composed of various entries, named Access Control Entry (**ACE**). Each of this ACE is intended to allow or deny traffic from a certain source to a certain destination. The more ACEs make up an ACL, the more complex it becomes. These represent a very important tool as they allow network administrator to block traffic not only based on layer 3 information but also on layer 2 and 4 information. The definition of an ACE therefore starts with 2 types of statements:

- Permit: if the traffic satisfies the conditions it is switched by the device;

- Deny: if the traffic satisfies the conditions it is blocked by the device.

It is important to underline that the implementation of these configurations varies according to the brand of the device. For the following configurations, reference is made to the syntax of Cisco device. In addition, the commands shown are presented in a simplified version. The complete configurations are shown in the following chapters. Once the statement to be used has been chosen, it is necessary to decide which type of traffic should be subject to this rule. Therefore, the following 2 parameters must be specified:

- Address: source and destination IP/MAC addresses;
- Protocol (optional): protocol used within the message;
- Port (optional): source or destination port contained in the message. This parameter refers to the type of layer 4 port, not to be confused with the ID of the interface on which the message passes through.

These checklists can be applied to interfaces in 2 different ways: inbound and outbound. In the first case, each packet entering the interface would be analyzed and compared with the configured ACL. The message at this point can be discarded or accepted depending on the defined ACEs. In the second case, the analyses are carried out on the messages that are coming out of an interface. Therefore, before being transmitted they are compared with the configured ACL. Generally, only one ACL can be configured per verse. It is therefore not possible to apply 2 inbound ACLs onto the same interface. In this case, the last one configured would overwrite the one already applied. As mentioned earlier there are several types of ACLs. Some types of those most relevant are described in detail below:

- Standard ACL: together with the extended ACLs, represent the most used options for device configurations. These can be applied to any device that performs routing functions (i.e., layer 3 functions). For their definition, the above statements are required. They are able to block any packet that has source address compliant with respect to ACEs;
- Extended ACL: these are an enhanced version of the ACL standards. In fact, in addition to using IP addresses, they can use both the type of protocol and the port number (source or destination) to manage network traffic. Using this type of ACL, the network administrator is able, for example, to block HTTP traffic to a specific IP address (e.g., web server);
- MAC ACL: these are based on the same concept of functioning as the previous 2. The substantial difference lies in the fact that the traffic is analyzed through the layer 2 information. In fact it is possible to use the destination and source MAC addresses. As for the extended, it is also possible to select the traffic based on certain protocols.

In an enterprise environment there are scenarios in which traffic must be filtered at the access level, for example managing the flow within the same VLAN (subnet). From the previously defined ACLs applicable to layer 3 interfaces, it is necessary to define other ACLs that can also be applied to layer 2 interfaces. These would therefore be used, for example, in the access switches to which the PCs are connected. These types of ACLs are defined as Port ACL (**PACL**) and can only be configured as inbound. PACLs can be implemented through the use of the 3 previous ACL types.

With the development of NAC technology, a new type of ACL has been presented: the downloadable ACL (**dACL**). These ACL specifications are assigned to the established session and not to the physical interface. Therefore, there could exist multiple dACLs applied to different sessions on the same interface. These are applied inbound, so all traffic generated by the device will be checked with the respective dACL. Sessions are established when the authentication and authorization steps are successful. To be applied, 2 communication steps must take place between the NAD and the NAC server via RADIUS messages. In the first, the NAC server communicates the name of the dACL to apply to the NAD. At this point, since there is no ACL configuration with this name, the NAD asks the NAC server if it can send it the respective entries. The server then sends the content to the NAD. The detailed process is reported in Chapter 5. As previously suggested, ACL configurations may already exist within the login device. The integration of the dACLs with the ACLs already present is strictly related to the software version of the device. In fact, while for some versions the dACLs completely overwrite the ACLs, for others they are concatenated. In this last scenario, the message coming from the device is analyzed by both lists, first by those downloaded and then by those already configured. With the dACL approach, the NAC technology allows centralized management of users who connect to the network.



# Chapter 3

## Survey on NAC technology

After a brief introduction, in Section 3.1 and Section 3.2, the main functionalities of the NAC technology and the main components are presented. The configurations, described in detail, of the 3 components are reported in Chapter 5.

### 3.1 NAC functionalities

NAC technology allows network administrator to control and monitor access to the network while keeping sensible data and services safe. Some examples of possible internal threats are the following:

- Unauthorized users access the LAN;
- Employees do not install or update the anti-malware software;
- Employees do not install OS patches;
- Employees install unauthorized software.

With the implementation of NAC, for example in a business environment, it tries to answer the following 4 questions:

- Who is allowed to access data and services (i.e., the company network)?
- What are they allowed to do?
- What rights do they have?
- How can they communicate?

Through NAC, the user (and/or the machine) requesting access is actively queried (e.g., through the profiling function) attributing certain authorization policies to him. The paradigm on which the functioning of these technologies is based is therefore that of policies. There are therefore different policy sets, each of which has a specific action. A correct implementation of the technology allows:

- Identification of each node connected to the network;

- Authentication and authorization for devices and users;
- Temporary quarantine and remediation services.

The fundamental aspect of NAC that distinguishes it from other solutions, is that it allows centralized management of network access security. The Figure 3.1 describes the general operating process involving the functions described in the current section.

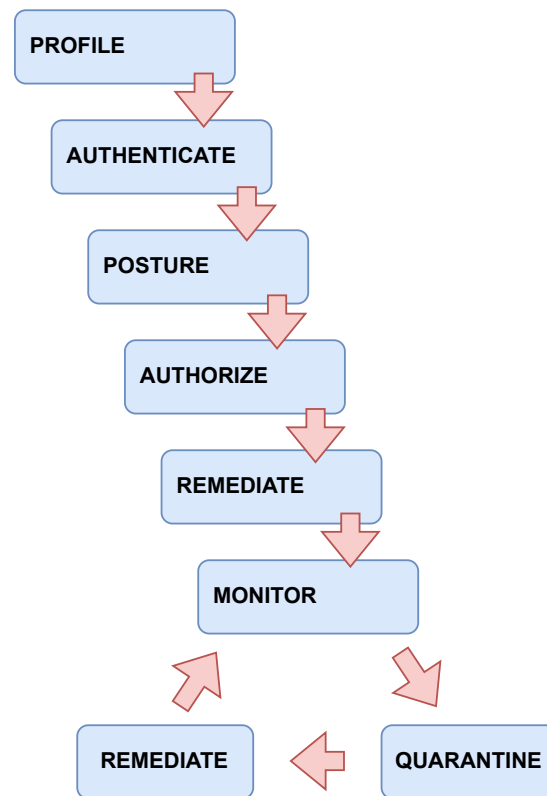


Figure 3.1: NAC flow.

An important emphasis needs to be made regarding the previous figure. As can be seen, the remediation phase is present twice during the user login process. Indeed, the device is not controlled only at the first access, but it is possible to enable Periodic Reassessment (**PRA**) during the period in which it is connected to the network. The assessment task is therefore associated with the monitoring phase.

### 3.1.1 Node detection or Profiling

The ability, by the NAC, to identify any device attempting to access the network is known as node detection. This function is of fundamental importance as it helps the other ones (e.g., authentication, authorization, accounting, posture assessment, etc.) to be carried out. The node detection function, within the solutions on the market, is also commonly referred to as profiling. This functionality



is described in detail in Chapter 6. There are several methods for detecting and identifying a device. Some of these are as follows:

- Through the Address Resolution Protocol protocol (**ARP**), the NAC server or even the device itself, can notify their presence within the network. In fact, the first message of the ARP sequence is transmitted in broadcast. Furthermore, this protocol allows the nodes belonging to the NAC architecture to associate the sender's IP address with the respective MAC address;
- In a NAC system based on 802.1X Port-Based Network Access Control (**PNAC**), the switch (or more generally the NAD), is able to detect the presence of a new device. This function can be implemented either through configuration (i.e., whenever the NAD identifies the change of state of a link from down to up), or through the automatic sending of Hello messages by the device itself. In both cases, the 802.1X protocol would be based on the Extended Authentication Protocol (**EAP**) framework;
- Through the use of SNMP, the NAC server, or even the NAD, can generate specific messages with the aim of detecting and identifying who is trying to access the network. For example, some switches can generate these messages whenever they receive an incoming message having as source MAC address one that is not inside the forwarding table;
- The same procedure explained for ARP messages can be applied to DHCP messages. In fact, one of the first types of messages that an IP device transmits to access the network is a DHCP discover message. As for ARP messages, this type of message is also transmitted in broadcast.

However, it should be noted that the scheme shown in Figure 3.1 takes on a general description. In the case study of InfoCamere, it is not certain that the order of the operations carried out precisely respects this flow.

### 3.1.2 Authentication

Once the device has been identified, it must be subjected to the authentication procedure. During this phase credentials are exchanged, or in general identification tools such as certificates are exchanged. In the scenario in which certificates are used, the NAC server check their validity through the following steps:

- Has the certificate been signed by a trusted CA?
- Is the certificate presented in a correct format?
- Is the certificate expired?

This phase may require the use of multiple protocols. In our case study, 802.1X, EAP, MAB and TLS are used. Once credentials have been provided, access to the network can be validated or denied, based on their validity. In general, authentication can take place in different ways:

- 802.1X: the most widespread among the authentication protocols;
- MAB: framework used in conjunction with 802.1X. Used with all those devices that can not install any software (e.g., printer, old IP phone, etc.) or don't support 802.1x protocol. Often used as a fallback method;
- Web Based: framework used very often in guest connectivity contexts. Access is performed by sending a username and password through a web portal. No software/agents must be installed on the device. However, this type of authentication is only used with interactive devices (i.e., it makes no sense to use it for printers). Furthermore, web authentication can be implemented in 2 different ways, namely *local* or *centralized*;
- Static Port/MAC configuration: highly unsafe and inexpensive method. The MAC address of the device that wants to access is statically configured within the network device. Within this one there is therefore a MAC address/Interface pair;
- Dynamic Port/MAC configuration: in this mode, authentication is managed with SNMP;
- Kerberos snooping: protocol used for authentication within a *Windows* Active Directory domain.

A brief discussion can be made regarding the web based authentication method. In fact, once the user is connected to the network, he is redirected to a specific access portal. However, there are various types of portals. Some examples mostly implemented in enterprise realities are shown below:

- Hotspot guest portal: once connected, the user is redirected to a web page. In this case the user does not have to register and therefore there is no check of any credentials. The user will therefore have access to the internet once he has accepted the Acceptable Use Policy (**AUP**);
- Sponsored guest portal: for this method the use of credentials is required. To access the network, however, it is not enough to provide the correct credentials, but the user must be sponsored by an internal employee of the company. In fact, the latter has the responsibility of approving or denying access to the network;
- Self-Registered Guest Portal: to gain access to the network, the user needs to create his own account with username and password. With this portal it is possible to keep track of the user's operations once logged in.

### 3.1.3 Authorization

Strictly related to that of authentication, the authorization function allows to associate ad-hoc policies to each device (or user) through which access to the

various resources is limited. In fact, once the credentials (or certificates) provided by the NAC terminal have been validated, it is necessary to determine which resources the user can access. If the authorization function was not present, any user with valid credentials would have full access to the network. In the implementation phase, since the organization of all authorization policies is a crucial phase, it takes up most of the time. Correct implementation should answer the following 3 questions:

- What level of granularity of authorization is needed?
- What would be the default access implementation?
- How are authorization policies managed within the enforcement point?

Authorization policies can be implemented in many ways. The exhaustive list is presented in Chapter 5. Two of the most used components that make up an authorization policy are listed below:

- **dACL:** used very often for authorization policies, downloadable ACLs are simple ACLs whose configurations are downloaded from the NAC server and implemented in a specific user session. If an interface was previously configured with a certain ACL, the latter would be overwritten/appendended by the dACL. Each user is therefore dynamically assigned dACLs based on their access authorization level;
- **VLAN:** once identified who tries to access the network, the interface on which the user is connected is assigned to a certain VLAN based on the policies set. An example scenario is when a user trying to access the network provides incorrect credentials (e.g., username or password not valid). In this case, the user is associated with a restricted VLAN, in which he has limited access only to basic services.

Regarding the concept of ACLs, it is important to distinguish between downloadable and dynamic ACLs. For the former, in fact, ACLs are downloaded every time an authorization rule is met. The NAC server sends both the name and the content of the specific ACL contained within the authorization profile. In the case of dynamic ACLs, these are pre-configured in the NAD. The NAC server simply sends the name of this ACL to the NAD which applies them.

### 3.1.4 Posture

One of the features that distinguish NAC technology is the posture. This allows the assessment of the conformity of the end device. This function is essential because, even if the user provides correct credentials, the device could still be the carrier of malicious software. This, for example, would allow malicious users to infect the entire company's internal network. The posture function therefore consists in obtaining information not strictly related to the hardware of the device, but rather to its state of health. Examples of this information are shown below:

- Software version of the OS installed;
- Results obtained from any Anti-Malware (**AM**) installed;
- List of verified applications;
- List of installed certificates and trusted CAs.

As mentioned in Section 3.2.1, the posture function can be implemented in different ways. Mainly two macro categories can be defined: agent-based and agentless. Within these, it is possible to make two other sub-categories. In the agent-based case, there exist thin and fat agents. While the former represent software of limited size, downloaded at the time of the posturing phase, fat agents are software pre-installed within the devices. In fact, these clients can be implemented within firewall or antivirus for end devices. In the case of agentless, the two sub-categories are network-based and applet-based. For the first, the evaluation is done remotely by the NAC server, while for the second java applets are used, which through a browser launch assessment functions.

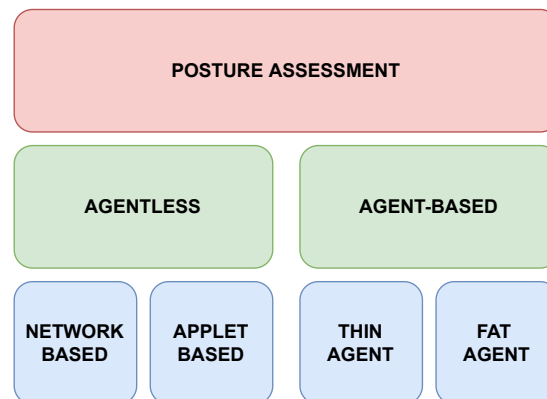


Figure 3.2: Assessment types.

### 3.1.5 Quarantine and Remediation

In the event that a device is considered as *Non-Compliant*, NAC is able to assign it to an isolated area, also known as quarantine. Through specific policies, the NAC server informs the NAD, which in turn restricts access to the end device. An example of such a situation can occur when a device with an outdated OS installed tries to access the network. This OS, no longer receiving security updates, can be exploited by malicious users to attack the network. An example of isolation is to assign such a device to a certain VLAN, in which it will not be able to communicate with the internal network. At this point the functionality of remediation comes into play. In fact, it is possible that the NAC terminal has the ability to solve the problem itself without the intervention of an operator. Returning to the previous example, the NAC server (if properly configured), is able to propose to the device the latest OS version compatible

with it. Once updated, the device is then granted access. Obviously, before being accepted, it is queried again to check that the defined policies are respected. In the case that the remediation function is not implemented, the network would be limited. In fact, due to its lack, many users (and their devices) may not have access to the network due to the fact that these would be not compliant. This could cause a problem for maintaining productivity and business continuity. It is also clear that the automation aspect of processes is in this case a key aspect. The service would therefore not imply any human intervention, increasing its productivity. Another possible option, however, is represented by the intervention of the user himself. In fact, once the device is defined as *Non-Compliant*, the NAC server could redirect the user to a web portal (located in a quarantine network). Through this portal the user is able to see:

- Information about the device that user attempted to sign in with;
- Report of the violations found;
- Instructions on how to address the problem;
- Link to retry access once the anomalies have been fixed.

It is therefore possible to divide the type of remediation service as follows:

- Self-remediation: the user is redirected to a web portal where he finds the information to solve the vulnerabilities found. This mode is suitable for all those devices that cannot be managed directly by the NAC server;
- Auto remediation: as described above, with this type the remediation is automated thanks to the implementation of scripts and executable;
- Third-party remediation: the NAC system, as previously mentioned, can be made up of several systems. If a Patch Management System (**PMS**) already exists within the network, it must be integrated with the NAC system. The PMS must therefore be constantly synchronized with the policies configured within the NAC server. The device is therefore first put in communication with the PMS, and once fixed, it is put in communication with the NAC server in order to be assessed again.

## 3.2 NAC architecture

The NAC architecture, shown in Figure 3.3 includes 3 fundamental components listed below:

- Network Access Source (**NAS**) or NAC terminal;
- Network Access Device (**NAD**) or enforcement point;
- NAC server.

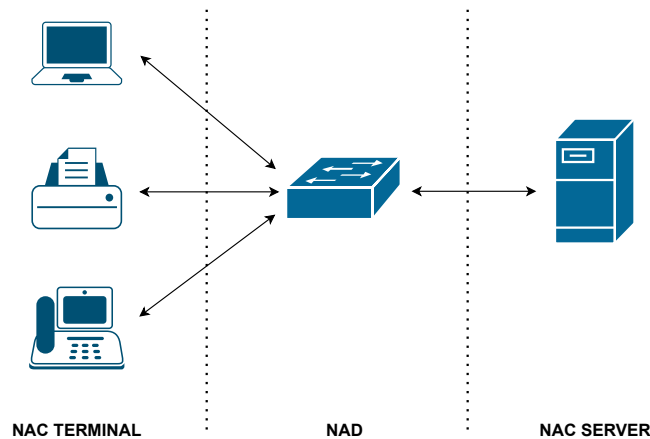


Figure 3.3: NAC architecture.

### 3.2.1 NAC Terminal

The first component of the system is the end device, which is the NAC terminal (e.g., PC, printer, IP phone, etc.). This represents the client trying to access the network. Upon accessing the network, the device will provide credentials (e.g., user, certificate, etc.) to gain access. The devices to authenticate can either integrate a client software, or not be equipped with one (e.g., as they use a native application component). There are therefore two types of clients within the NAC solution:

- Agent-based;
- Agentless.

The first involves installing a software into the device attempting to access the network. Such software is commonly referred to as a NAC agent. This software is able to carry out posture assessments (therefore assess the health of the device) and send this information to the central node, that is the NAC server. Often, these can also communicate with any security applications installed on the device (such as antivirus, firewall, etc.). Furthermore, it has the task of managing all the authentication and authorization processes that the device must perform. As for the second type, there is no software installed (NAC agent). This condition can be verified by the central node by, for example, the use of timeouts. In fact, if the device were interrogated several times without giving any answers, it would make the NAC server understand that it does not have any software installed. At this point, one of the possibilities available is to have a temporary agent downloaded which, once its functions have been performed, is uninstalled. In this way the device returns to the initial state. In addition to this, another possibility is represented by the fact that the NAC server can execute the processes itself, communicating with the device directly, without the use of any software.

### 3.2.2 NAD

The NAD represents the device that acts as an intermediary. It has the task of receiving messages from the NAS (NAC terminal) and sending them to the NAC server. This role is usually entrusted to devices such as switches, Wireless Lan Controllers (**WLC**) and routers. This type of device is typically configured to support Authentication, Authorization and Accounting (**AAA**) services. Being the primary access point to the network of devices, it has the task of implementing some of the basic policies (e.g., blocking or accepting the traffic generated by the NAS).

### 3.2.3 NAC Server

The NAC server is the focal point of the NAC. Also known as decision point, the NAC server is responsible for applying all the rules to regulate access to the network. All requests made by the NAS are then redirected to this device. When deploying NAC within an enterprise environment, most of the work goes into configuring this central node. Being the fundamental component, on some architectures it is possible to divide the functions of this node into several nodes, in such a way as to form a cluster-like architecture. Furthermore, this implementation allows for redundancy. If there were to be some kind of disruption on the part of this node, the devices (and users) that would try to access the services would be isolated. In some cases, however, it is possible to configure fallback systems within the various types of NAD in such a way as not to create a total disservice.

Also, in an enterprise environment, the devices and users are associated with a domain. It is therefore commonly used to interconnect this server to Active Directory (**AD**), in such a way as to use external archives to perform some services (e.g., identity control for authentication). According to the implemented solution and the chosen vendor, the NAC server could be composed of several sub-components. Some examples of deployment are presented in Section 4.1.2.

## 3.3 Pre and Post admission

The implementation of NAC technology can be based on two opposite concepts: pre and post admission. So far have been presented the implementation of the first type, which is that access is granted after being identified as *Compliant*. Pre admission therefore implies the execution of authentication and authorization before any communication. On the other way around, in post admission access to the network is first allowed and, at each attempt by the user to move within the network, checks are carried out.

### 3.4 In-line or Out-of-band

The NAC architecture can be implemented in two different ways which are in-line and out-of-band. Both concepts differ in the way data flows are handled.

In the first case, the authenticator (i.e., NAD) is positioned directly between the end device and the NAC server. In this way all the traffic exchanged between these two nodes passes through the NAD. Thanks to this implementation, it is possible to have a better management and monitoring of accesses. Indeed, assuming has been implemented the NAC system using the post-admission paradigm. In this case it is very simple to control users for each access to a different network segment. Having passed through the data flow, it is possible to inspect each packet. For this reason, NAC can be compared (roughly) to an internal firewall. The devices used in this scenario can also be equipped with Application-Specific Integrated Circuit (**ASIC**).

As for the out-of-band solutions, these are not positioned directly where the data flow travels. The device in fact obviously resides within the network, but it is somehow transparent from the point of view of the end devices. Once the steps to access the network have been carried out, the NAD is no longer able to monitor the traffic as it is disconnected from the flow. This model, unlike the previous one, is used in scenarios where it is necessary to implement a NAC solution on a pre-existing infrastructure.



# Chapter 4

## Case study

The following chapter describes the current implementation of the NAC system within InfoCamere (**IC**). The architecture is based on the multi-vendor paradigm, where the role of NAD is covered by devices belonging to different vendors (i.e., Cisco, Aruba and HPE), while the NAC server is a Cisco solution. As for the test environment, only Cisco NAD has been used. Before reporting an overview of the IC infrastructure, the Cisco NAC solution is presented.

### 4.1 Cisco ISE

Identity Services Engine (**ISE**) represents the NAC solution proposed by Cisco in order to ensure secure network access by authorized users and machines. Through ISE, the enforcement point is moved from the single device (e.g., an application firewall) to the NAC server. This concept bases its operation on the Remote Authentication Dial-In User Service (**RADIUS**) protocol, with which it is possible to take advantage of the AAA functions. Therefore, when a user or device needs to authenticate itself, it is the NAD's role to send all information regarding authentication to Cisco ISE. At this point ISE responds to the request by sending the resulting security policies to the NAD to be applied for that specific user, employing certain options present within the RADIUS packets. Each request is therefore treated individually, applying different policies to devices connected to the same interface. Cisco ISE also offers two advanced functions that allow to increase the level of security, namely profiling and posture (both detailed in Chapter 6 and Chapter 7). Through profiling, the NAC server (ISE) is able to identify the device attempting to connect to the network, recognizing for example the manufacturer and the function of the device (e.g., IP camera, printer, etc.). This information can also be used within the authorization policies. For example, by connecting a new printer to the network, ISE is able to identify it as such and is therefore assigned a specific authorization profile [6]. Being a device that in general must not generate traffic, it must be given as restricted access as possible. One of the main best practices regarding the provision of access levels is to give the minimum possible access to those who request it. In fact extra access levels could be exploited by threat actors.

Term	Description
Node	Any individual physical, virtual or cloud Cisco ISE appliance.
Persona	The type of persona implemented in a node determines which services it makes available. The personas that can be implemented are the following: <b>PAN</b> , <b>MnT</b> , <b>PSN</b> and <b>PXG</b> .
Service	Specific feature that a persona provides. Some of these are network access, profiling, posture and monitoring.
Role	The Administration and Monitoring nodes can be characterized by different types of roles. These are standalone, primary secondary.
Node Type	Cisco ISE node can assume one among the following personas: Administration, Policy Service, Monitoring.

Table 4.1: ISE deployment terminology.

The posture function, on the other hand, is characterized by a deeper level of detail than the profiling one. While profiling uses network-level communications to obtain device information, posture uses information residing within the device. Through an agent installed on the device or through an agentless process, ISE is able to verify that the device is *Compliant* with the defined security policies. Therefore, depending on the results of the posture analyses, different access levels can be guaranteed. It is also possible, for those resulting *Non-Compliant* devices, to provide them remediation functions. Given the numerous functions that ISE offers, it is necessary to have a monitor function with which network administrators can verify the status of the sessions for each access attempt. To meet this need, there is a logging function in which all events are stored. For each event, all the phases of the various processes are also recorded, making it possible to perform a detailed analyses. ISE can fill multiple roles and provide multiple services within an enterprise environment. In order for all of these functions to work together efficiently, it is necessary to implement a distributed architecture. The different types of deployments are discussed in Section 4.1.2. To better understand the following paragraphs, Table 4.1 shows some fundamental definitions. The implementation regarding the single nodes can be carried out in the 3 ways shown in Section 4.1.2. As far as physical implementation is concerned, Secure Network Server (**SNS**) appliances can be used. An example of this type of newer appliance is the *SNS-36XX* series. In addition to physical machines, it is possible to use virtual ones through different hypervisors. These hypervisors are therefore installed on general purpose machines. In order to use these machines correctly, it is necessary to allocate the same resources as the SNS appliances. Supported hypervisor vendors are vmWare, KVM, Microsoft and Nutanix. The last sup-

ported solution is the cloud one. Currently, only the Amazon cloud service is supported, while those of Google, Oracle and Microsoft will also be supported in the future (e.g., with the new releases of ISE 3.X). It is also important to remember that, depending on the persona configured within the single node, different configurations are available.

### 4.1.1 Personas

The most widely implemented architecture in an enterprise environment is undoubtedly the distributed type (described in Section 4.1.2). There are therefore more nodes (often redundant to provide the same usability in case of failure), to which different functions are associated. To define such an architecture, ISE allows the configuration of multiple personas in multiple nodes. A node can therefore be configured with one or more persona (up to a maximum of 4). These personas, already mentioned previously, are described in the following paragraphs. Note that when the term “*ISE cube*” is used, it refers to the deployment of ISE.

#### Monitoring

The Monitoring persona acts as the central node for collecting logs related to the ISE cube. A node that implements this persona is referred to, within the deployment, as Monitoring and Troubleshooting (MnT) node. As previously mentioned it is possible to implement this persona in 2 different nodes. In this way, one would constitute the primary node, while the other the secondary. For example, when a client tries to authenticate (and thus receive some authorization) log events are created. The latter are then sent to the MnT node, which has the task of processing and presenting them in readable form. At this point, the network administrator has the possibility to view all the events, identifying, if any, the critical ones. These events can also be used for other purposes than mere monitoring, or for statistical analyses. This node is in fact able to generate summary dashboards and generate scheduled reports. In the configuration phase, at least one MnT persona must be implemented. When configuring the PSN, it is therefore also necessary to enable MnT. When the secondary node is also implemented, it is possible to disable the one configured previously. A scenario in which both primary and secondary nodes are present can be considered as redundant. In the event that the primary should stop working, it would be dynamically replaced by the secondary. In order to guarantee maximum performance, it is recommended not to implement both the MnT and the PSN in the same node. It is therefore better to divide them into 2 separate nodes. Another function that this persona can provide, mentioned above, is that relating to troubleshooting. Since each event occurred in the PSN is recorded by the MnT, in the case of failed authentication or other problems it is possible to refer to this element of the ISE cube. In this way the MnT also provides a sort of centralized service.

## Administration

One of the most important persona is that of Administration. Through it, the network administrator is able to configure and manage network security policies. Therefore, to make a change to the ISE implementation, the network administrator needs to log in to that persona. By accessing a Graphical User Interface (**GUI**), the administrator can conduct any modification he needs to do. It is important to underline, however, that the use of the GUI is from a certain point of view limiting for the administrator. In fact, not all possible changes can be made from the web portal. Some of those must in fact be executed using the Command Line Interface (**CLI**). This interface is usually used during the configuration of the various network devices such as switches, routers, etc. Through the CLI it is possible:

- Start and stop ISE application software;
- Upgrade the software;
- Restore application data from a storage device for backup's purpose;
- Troubleshooting by viewing the logs;
- Reload or shutdown ISE appliances.

The CLI can be accessed in 2 different ways: the first method (out-of-band) consists by connecting terminal equipment to the appliance via a serial cable. As for the second method, the use of an SSH connection (possibly v2) is foreseen in order to encrypt all the traffic between ISE and the terminal. Generally the latter method is preferred since being the enterprise-level architecture it is difficult to have access directly to the appliance. A node implementing the Administration persona is referred to as a Policy Administration Node (PAN). Once a change has been applied, such as the configuration of a new authentication policy, it is automatically sent to the other nodes/personas. In this way all components are synchronized with each other.

The Cisco ISE deployment follows the licensing model. This means that to take advantage of certain functions it is necessary to purchase extra dedicated licenses. The issue of license management is explored in the Chapter 8. As mentioned above, the maximum number of nodes that can implement this type of persona is 2. In this scenario, they are therefore divided into primary PAN and secondary PAN. Being ISE extensively integrated with certificate management, the PAN can play also the role of root for the integrated certificate authority.

## Policy Services

Another fundamental persona, together with the administration one, is that of Policy Service. In fact, this is considered the control center of the entire system, in which all the functions made available by ISE are applied. The main features for which ISE is configured are authentication and authorization. Both

are implemented via the RADIUS protocol. This persona therefore offers the RADIUS server service with which the various NADs interface to apply policies. A node that implements such a persona is referred to as a Policy Services Node (PSN). Therefore, when a network administrator makes changes through the PAN regarding policies, all PSNs will be updated accordingly, remaining synchronized with each other. Depending on the type of architecture implemented, each NAD may be able to communicate with more than one PSN. While for the PAN, there are limitations in terms of the number of nodes, with the PSNs such a limit does not exist (at least in part, refer to Figure 4.4 for more detail). Therefore, multiple PSNs can be configured according to the required functionalities. It is in common use, in large enterprises, to implement a large number of PSNs. Through balancers, or similar devices, requests are redirected according to the traffic load that is measured at that moment. To use load balancers, it is still necessary to configure additional parameters (both within the NAD and ISE) so that the requests are handled correctly. In general, for each authentication, the NAD sends the Access-Request RADIUS packet and the subsequent ones towards the PSN. The latter, following the policies defined through the PAN, acts accordingly by replying to the NAD. However, the PSN is not required to inform what type of response it has sent to the NAD.

Through the Change of Authorization (**CoA**) functionality, the PSN has the ability not only to respond to requests received from the NAD, but is able to inform the latter about changes to be made to the access levels of a particular user. In this way any end device can be forced to re-authenticate, for example, obtaining in some cases new authorization levels. Again, the participation of the PAN in such scenarios is not foreseen. In the following chapters some of the main functions that Cisco ISE offers, applied to a wired infrastructure context, will be described in detail.

## **pxGrid**

Cisco pxGrid is an open Security Product Integration Framework (**SPiF**) that allows for bi-directional platform interactions. Its main functionality is to share data and information related to the secure implementation of ISE in a scalable and efficient way towards third party systems.

### **4.1.2 Deployment architecture**

The functionalities of the ISE NAC system are available through multiple implementations. As already mentioned above, this technology is aimed at all those environments where the number of devices to be managed and monitored is high. The scale with which ISE is implemented does not generally change the flow of the processes it takes on: there is therefore a user/machine that tries to authenticate to the NAC server (ISE) through a NAD. Each node of the system can therefore assume the role of one or multiple personas. There are 2 main types of deployment:

- Single-node (or standalone);
- Distributed.

During the design phase, one of the main criteria on which to define the architecture to be implemented is undoubtedly the number of devices to be managed. Therefore, based on the number of access requests that are estimated, the network administrator must implement a system that fully satisfies them.

### Standalone deployment

The simplest architecture to implement, using Cisco ISE solution, is the single-node one. With this trivial structure, the entire NAC system resides within a single appliance (physical or virtual). In this scenario, there is therefore the concentration in a single node of all the personas previously described. The requests that the NADs produce for each access attempt are redirected to that node, which by applying the configured policies provides a certain level of access or denies it. Figure 4.1 shows a representative diagram of this architecture.

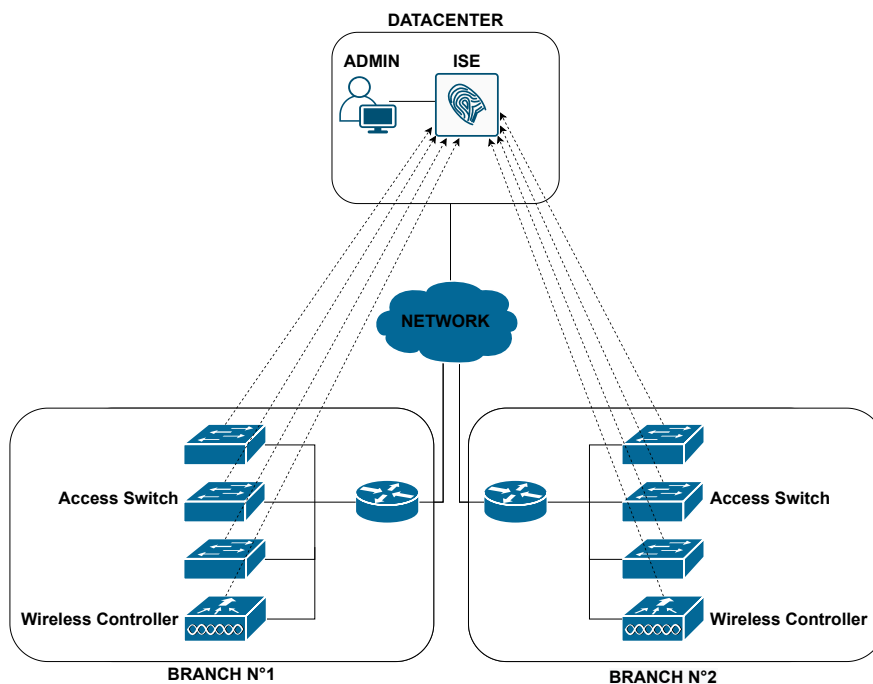


Figure 4.1: Standalone deployment.

The central node, which in this case is represented by ISE, collects all the requests coming from the access devices (switch and WLC). The authentication and authorization phases rely on internal database, whose use is recommended only for temporary entities. In the event that it is necessary to create a temporary user for an external consultant, it is recommended to rely on this internal structure. Since all 4 different personas are concentrated within this node, it represents an access point for network administrators. Through a GUI, presented in the next

paragraph, it is possible to make any changes regarding the infrastructure. With the standalone deployment there are no fallback mechanisms (redundancy). In the event that the physical or virtual appliance should fail or lose connectivity for any reason, the network access control would be disabled thus creating a serious disservice. The central node can thus be considered as a single point of failure. The adoption of this architecture is strongly recommended in a test environment. With this architecture the maximum number of endpoints supported is 10000 for the *SNS-3615* (and equivalent virtual devices) or 50000 with the *SNS-3695* (and equivalent virtual devices).

### Distributed deployment

In production environments, other types of deployment are implemented, which involve the use of 2 or more appliances. These can be divided into:

- Small deployment;
- Medium deployment;
- Large deployment.

In all 3 cases, each node can implement one or more personas, allowing a certain level of redundancy. Figure 4.2 shows a generic distributed deployment in which each node of the system implements a different persona. The PSN manages all accesses from different locations. In fact, there are requests that come from the company's wired campus network, from remote offices and from mobile devices such as surveillance cameras and smartphones. The policies to which requests are submitted are sent from the PAN to the PSN, after the network administrator has defined them through the GUI. The PSN then informs the MnT through logs of all messages generated and received. Finally, both the PAN and the PxG provide context information for third party services.

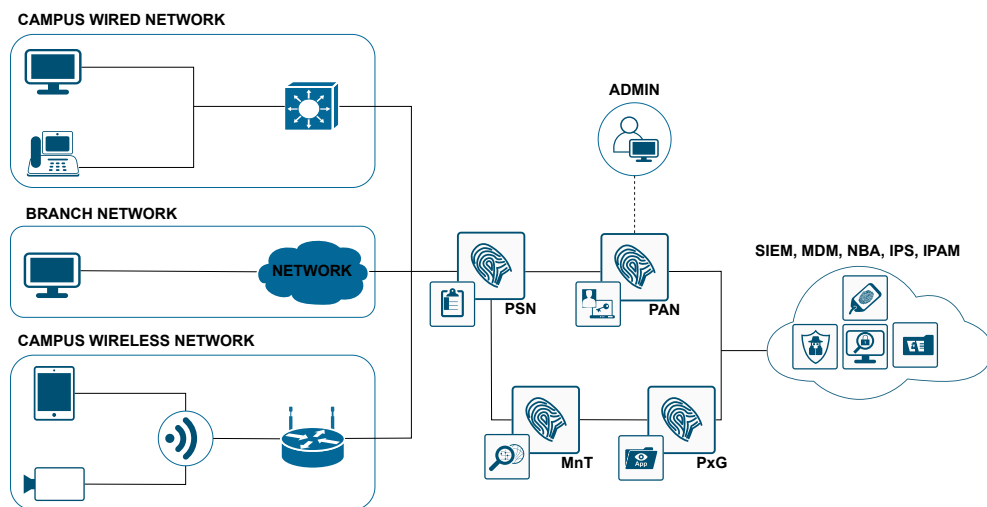


Figure 4.2: Cisco ISE deployment in a distributed architecture.

As for small deployments, they represent the smallest architecture possible within a distribution context. It is important to underline how the difference between standalone and small deployment is that in the latter it is possible to rely on external identity sources such as Active Directory (**AD**). In fact, in the standalone scenario, only users defined within the system database are used, without relying on external repositories. It should be noted that a hybrid deployment was adopted in the laboratory environment. While all personas were integrated within the same node, there was also a connection with an external AD.

As mentioned above, having more nodes available increases the level of redundancy that characterizes the network. In addition to being both physical or virtual, appliances can also be of a mixed nature. Each persona must be enabled on at least one of the two nodes, depending on the network topology. It is important to remember how the Administration and Monitoring personas can be distributed over a maximum of two nodes following the concept of primary and secondary node. On the contrary, the number of PSNs is generally not limited, allowing the requests to be redirected to the various PSNs. The primary and secondary nodes are also constantly in communication through a dedicated channel. The secondary node can therefore be seen as a replica of the primary one. The synchronization status can also be consulted via the GUI. Unlike secondary nodes, the PSN status is always active and therefore always reachable from the various NADs. The decision on which the individual NAD is based on which PSN to contact is closely related to their configuration. All PSNs can be configured within NADs in a hierarchical order. To balance the requests, during the configuration phase, it is recommended to point 50% of the NADs towards one PSN, while the remaining 50% towards the other one. However, there are more advanced systems (e.g., load balancer appliance) to carry out this task.

Starting from the small deployment scenario and adding 2 or more nodes, the medium deployment scenario is created. The nodes added starting from the previous scenario are dedicated exclusively to access management. They therefore play the role of PSN. However, it is essential to remember that the number of PSNs (e.g., up to 5 PSNs as shown in Figure 4.4) is limited if the Administration and Monitoring personas reside within the same node. The fact of being able to distribute PSN dynamically guarantees a decrease in Round Trip Time (**RTT**) and an increase in the level of redundancy. Indeed, Cisco ISE offer an auto switchover function, with which the primary node can switch its functions to the secondary one. To enable this function it is needed to enable the “*Enable PAN Auto Failover*” option from the ISE GUI deployment panel. In this way it is possible to configure a function called “*Polling Interval*”, with which a specific node searches every certain time period (which is configurable), to verify that the primary administration node is still reachable and active. In the event that a certain number of failed attempts is reached, the secondary node takes over the role of primary node. More in detail, once the failover function is enabled, 2 health check nodes must be configured. These, whose role cannot be personified by the PAN, have the task of testing the connectivity with the PANs. The primary health check node then sends probes to the primary administration node. In the



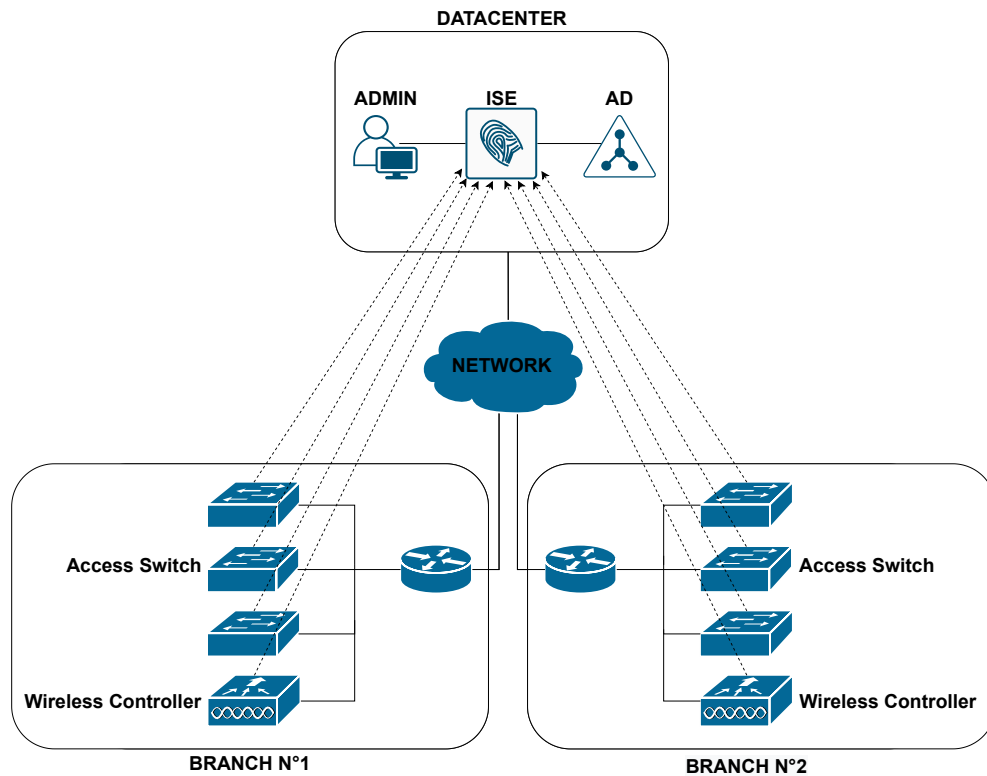


Figure 4.3: Small deployment scenario.

event of anomalies, it is the primary health check node's task to inform and elect the secondary administration node as primary node [7] [8]. Furthermore, within the individual PSNs it is possible to configure different additional functions, such as TC-NAC and SXP. Should the PANs and MnTs stop working, the access control service continues to be available. Compared to the previous scenario, the disservice created would be more limited in this case. For a medium deployment the maximum number of supported sessions is 50000. To reach a higher number of sessions it is necessary to implement the large deployment (described in the next paragraph). Figure 4.5 shows the scenario in which both nodes (located in different DCs) have full functionality.

In the large deployment scenario, each node can be dedicated to the implementation of a single persona, while continuing to take advantage of the concepts of primary and secondary nodes. Traffic, being redirected to a specific node which implements just a single persona, is managed more efficiently. From the previous scenario, however, the concept of nodes dedicated exclusively to the PSN function remains. Another big difference lies in the use of the load balancer appliance. The requests, instead of transit directly towards the PSNs, are directed towards the load balancer, whose task is to balance the traffic along the various PSNs. In general, the implementation of 2 or more appliances of this type is recommended, to meet the redundancy needs. Figure 4.6 shows an example of large deployment.

Deployment Model	Platform	Max Number of Dedicated PSNs	Max RADIUS Sessions Per Deployment
Standalone	3515	0	7500
	3595	0	20,000
	3615	0	10,000
	3655	0	25,000
	3695	0	50,000
PAN and MnT on same node and Dedicated PSNs	3515 as PAN and MnT	5	7,500
	3595 as PAN and MnT	5	20,000
	3615 as PAN and MnT	5	10,000
	3655 as PAN and MnT	5	25,000
	3695 as PAN and MnT	5	50,000
Dedicated (PAN, MnT, PXG, and PSN Nodes)	3595 as PAN and MnT	50	500,000
	3655 as PAN and MnT	50	500,000
	3695 as PAN and MnT	50	2,000,000

Figure 4.4: Maximum RADIUS scaling for SNS 3500/3600 series.

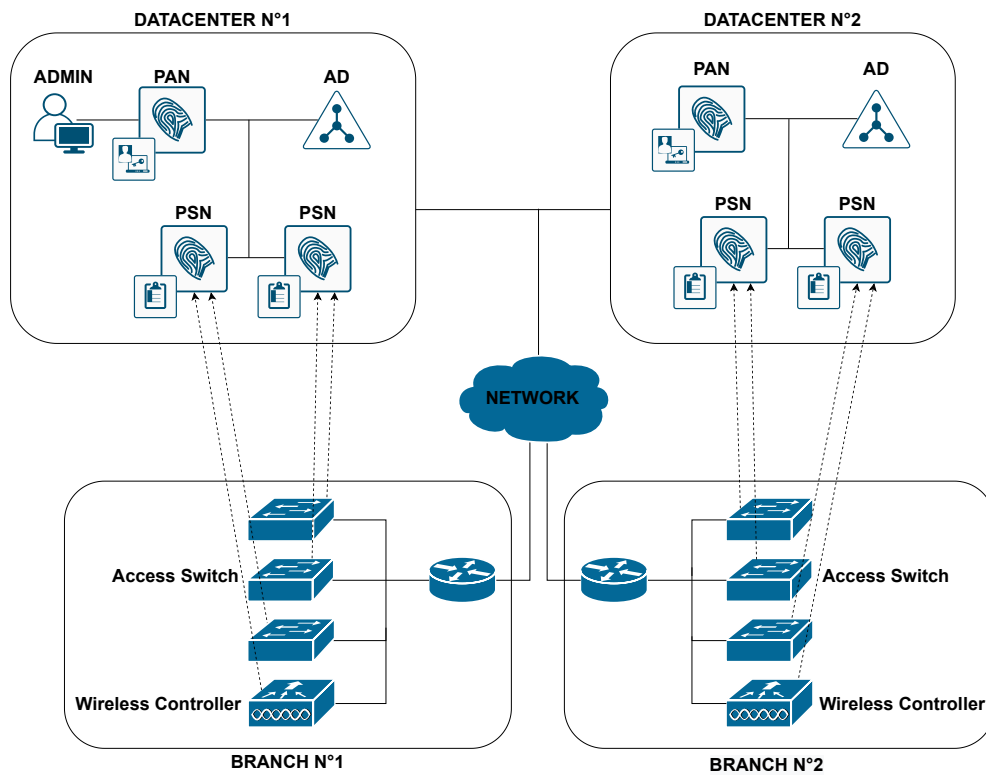


Figure 4.5: Medium deployment scenario.

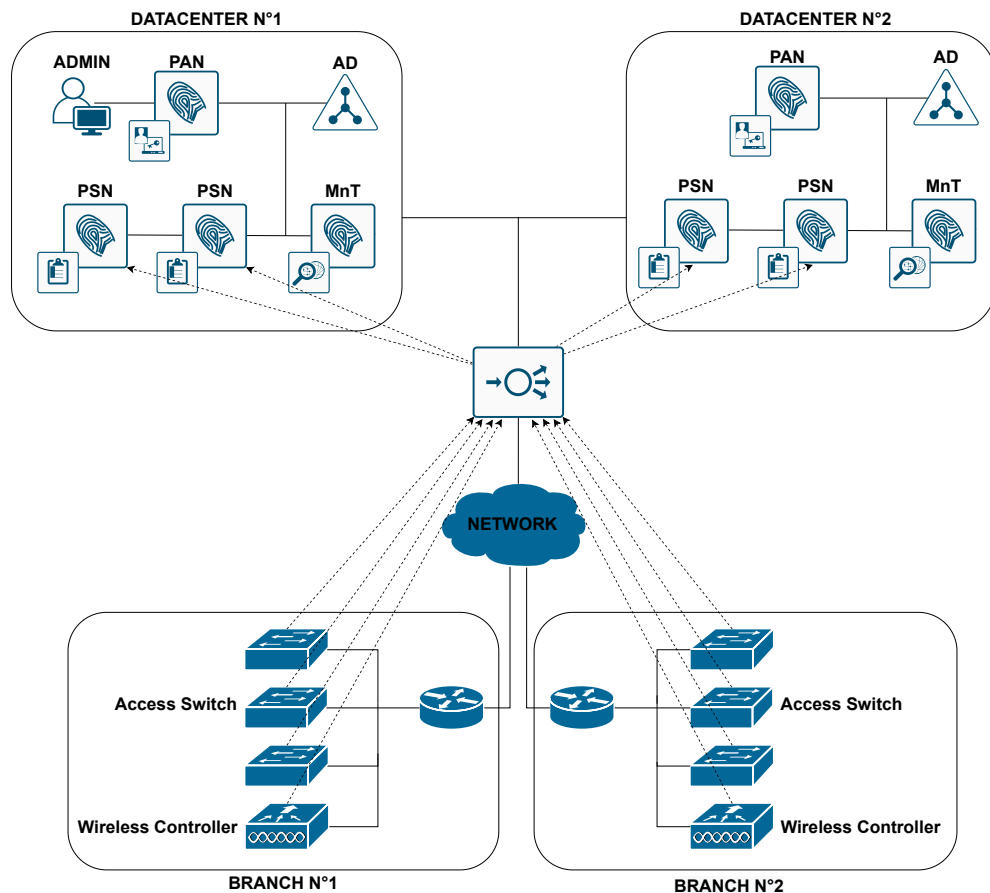


Figure 4.6: Large deployment scenario.

### 4.1.3 ISE GUI

To interact with ISE it is possible, as previously mentioned, to use either the CLI or the GUI. Figure 4.7 shows the main GUI of ISE in the InfoCamere environment. From this first interface it is possible to view a summary of the users and identities that attempted access, the configured network devices (NAD) and any anomalous events identified. It is important to highlight how this dashboard can only be viewed from the PAN node. Depending on the personas configured within the node, different options are available. From the main menu it is possible to switch to other windows, that are:

- Context Visibility;
- Operations;
- Policy;
- Administration;
- Work Centers.

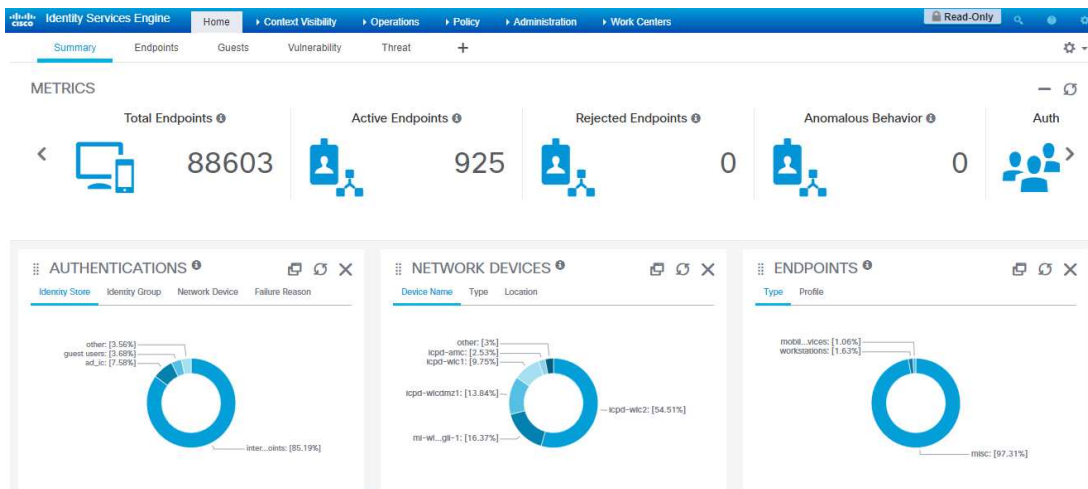


Figure 4.7: ISE GUI.

Each of these tabs also offers submenus where it is possible to make changes or monitor data in real time. The context Visibility tab allows the administrator to see detailed information about endpoints, users and NAD. This information can be organized in multiple ways. All this information is collected by ISE starting from the internal databases, buffers and caches. The policy tab allows to configure and manage all the policies to which requests are subject. Furthermore, in this section there are submenus concerning the profiling and posture functions. The administration tab allows the network administrator to manage the characteristics of the node such as licenses in use, certificates, network devices and guest services. The last tab, on the other hand, is a sort of grouping of the previous ones in which the same section can be reached using a different path. Below are some of the many sections useful for studying the functionalities subject to analysis. Figure 4.8 shows how the authentication and authorization policies are organized within the policy tab. It is important to underline that for all the figures shown, filters have been applied in such a way as to obscure any sensitive contents. While this chapter provides an overview of the interface, for the technical aspects refer to Chapter 5 and following. Policy sets are organized hierarchically. Each user will then be compared to the rules that are ranked higher. In the event that the access request matches the conditions relating to a specific set, the authentication process continues in another section shown in Figure 4.9.

Once associated with a policy set, the request is first subjected to authentication policies and then to authorization one. While in the first section the identity of the device/user is checked using mainly external sources (e.g., AD and LDAP), in the second one the user access levels are defined. For example, a user belonging to the Marketing group is associated with a certain authorization level. Thus it is not possible to access some sensitive areas of the DC. In this section it is possible to use the results obtained from the profiling function. Figure 4.10 shows the profiling management interface within ISE. A device can be profiled through multiple methods. Based on the exchange of messages and the in-depth

Status	Policy Set Name	Description	Conditions	Allowed Protocols / Server Sequence	Hits	Actions	View
✔	Datcenter-Network		DEVICE: Device Type CONTAINS All Device Types#InfoCamera RADIUS#	Default Network Access	2922	⚙️	➔
✔	ARUBA-Radius-NetworkDevices		AND OR Radius-Service-Type EQUALS NAS Prompt Radius-NAS-Identifier STARTS_WITH Network_Switch Radius-NAS-Identifier STARTS_WITH	Default Network Access	2776	⚙️	➔
✔	IC-NAC_Test		Radius-NAS-Identifier STARTS_WITH	Default Network Access	892	⚙️	➔
✔	IC_NAC		Radius-NAS-Identifier STARTS_WITH	Default Network Access	4129870	⚙️	➔

Figure 4.8: Policy set section.

Status	Policy Set Name	Description	Conditions	Allowed Protocols / Server Sequence	Hits
✔	IC_NAC		Radius-NAS-Identifier STARTS_WITH	Default Network Access	4129870

- ➔ Authentication Policy (5)
- ➔ Authorization Policy - Local Exceptions
- ➔ Authorization Policy - Global Exceptions
- ➔ Authorization Policy (15)

Figure 4.9: Authentication and Authorization policy of a specific set.

analyses of their characteristics, ISE is able to profile the devices. These profiles can also be grouped into logical containers, called “*Logical Profile*”. These containers can be used as conditions within the policy sets. Naturally, each logical container is composed by profiles of devices that perform similar functions. For example, one of the predefined logical profiles is that of the “*IP-Phones*”, which contains all the IP phones, also from different vendors. Another important interface that a network administrator has to deal with is that relating to logs. In fact, from this page it is possible to check the status of the authentication and related sessions. In the event that there should be anomalous behaviors or more simply it is necessary to carry out a detailed analysis, consulting this section is strongly recommended. Figure 4.11 shows the page just described. This interface, together with other tools, proved essential for the analyses presented in the following chapters.

The following figure shows the interface relating to troubleshooting operations. Figure 4.12 is part of the screen shown in Figure 5.35. It should be noted that the codes relating to the individual lines are not used to sort the various

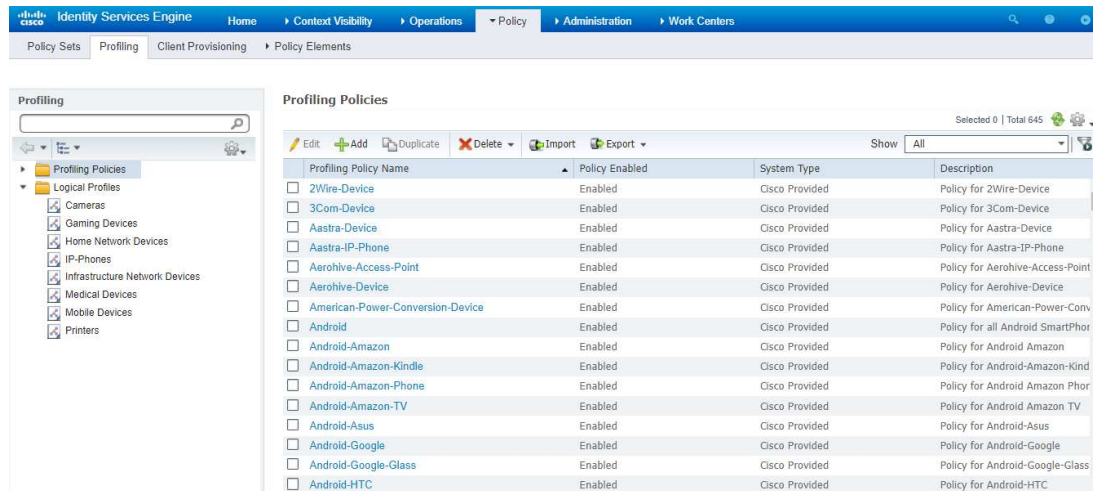


Figure 4.10: Profiling interface.

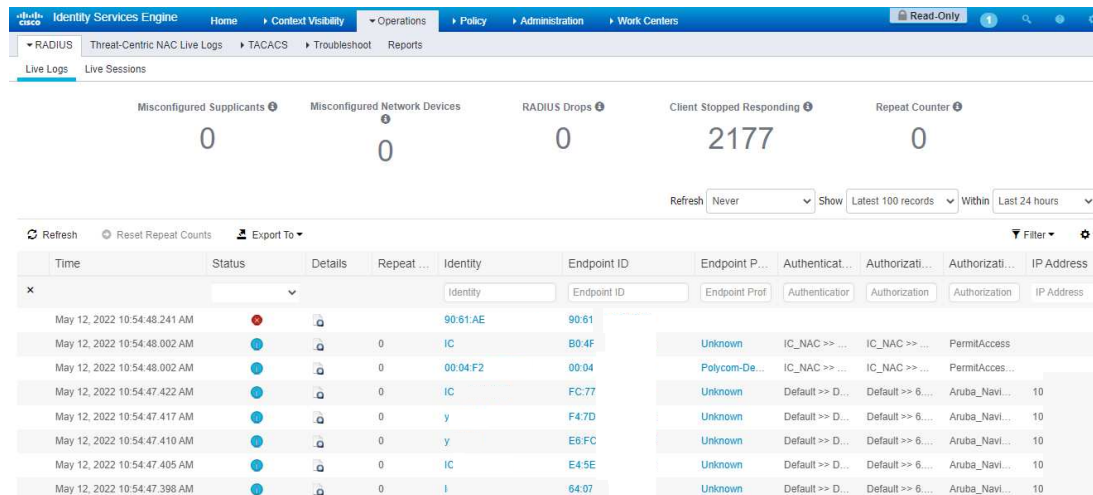


Figure 4.11: Radius logging interface.

stages of the process, but identify the type of operation performed.

In addition to the Cisco ISE solution, there are numerous alternatives to implement NAC technology within an enterprise network. One of the most adopted, and the one on which the comparison will be based in the last chapter is ClearPass from Aruba. Although they are different, especially in terms of user experience, they both provide the main functions that a NAC system should be equipped with. By way of example, Figure 4.13 shows the ClearPass interface.

## 4.2 InfoCamere

InfoCamere S.C.p.A. (IC) is the IT consortium of the Italian Chambers of Commerce (CCIAA) for digital innovation. It has created and manages the national telematic system that interconnects all the Chambers of Commerce and their 300 branch offices. Its institutional function is also the management and

### Steps

11001 Received RADIUS Access-Request  
 11017 RADIUS created a new session  
 11027 Detected Host Lookup UseCase (Service-Type = Call Check (10))  
 15049 Evaluating Policy Group  
 15008 Evaluating Service Selection Policy  
 15048 Queried PIP - Normalised Radius.RadiusFlowType  
 15041 Evaluating Identity Policy  
 15013 Selected Identity Source - Internal Users  
 24210 Looking up User in Internal Users IDStore -  
 24216 The user is not found in the internal users identity store  
 22056 Subject not found in the applicable identity store(s)  
 22058 The advanced option that is configured for an unknown user is used  
 22060 The 'Continue' advanced option is configured in case of a failed authentication request  
 24715 ISE has not confirmed locally previous successful machine authentication for user in Active Directory  
 15036 Evaluating Authorization Policy  
 24209 Looking up Endpoint in Internal Endpoints IDStore -  
 24211 Found Endpoint in Internal Endpoints IDStore  
 15016 Selected Authorization Profile -  
 11022 Added the dACL specified in the Authorization Profile  
 24209 Looking up Endpoint in Internal Endpoints IDStore -  
 24211 Found Endpoint in Internal Endpoints IDStore  
 11002 Returned RADIUS Access-Accept

Figure 4.12: Detailed radius logging interface.

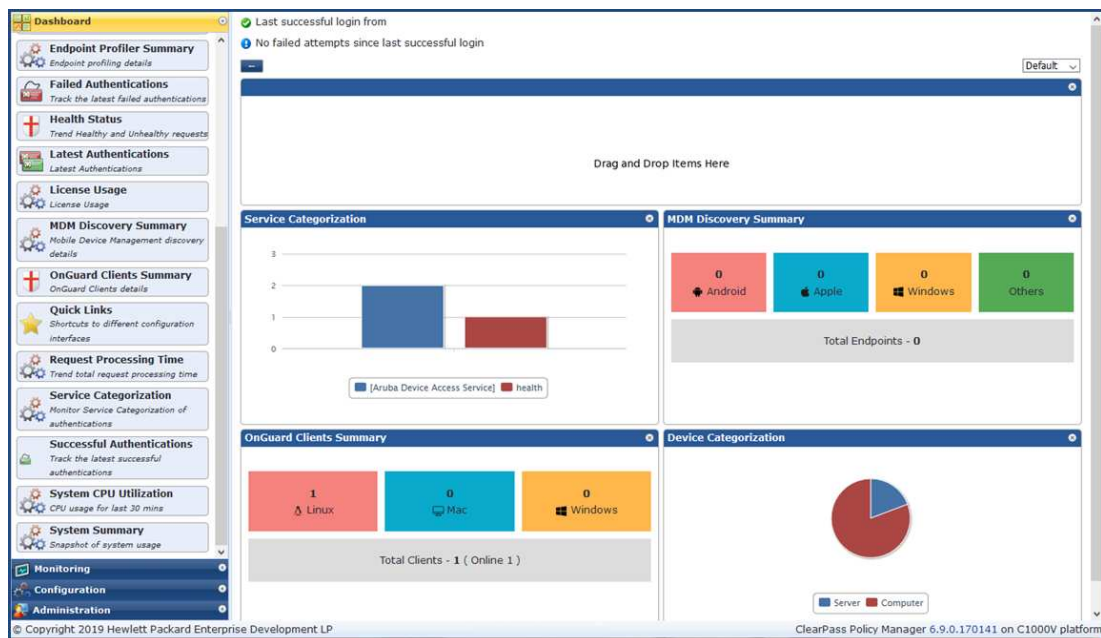


Figure 4.13: ClearPass interface.

dissemination of the information assets of the Chamber, with particular reference to information deriving from the Business Register. Initially born as Veneto Re-





Figure 4.14: InfoCamere’s logo.

gional Data Processing Center (**CERVED**), it was founded in December 1974 in Padova by Professor Mario Volpato, then President of the Padova Chamber of Commerce and Professor of Probability at the University of Padova. It has its registered office in Roma, while the operational headquarters is still located in Padova. Other branch offices are located in Milano and Bari. In order to provide an interconnection service between the Chambers of Commerce and secure and reliable access to the DC, InfoCamere has obtained multiple ISO certifications over the years. Some of these are *ISO 9001* for quality management, *ISO 22301* for IT service continuity and *ISO 27001* for information security. One of the aspects characterizing the IC infrastructure is the continuity of service offered. While the main DC is located in the Padova headquarter, and therefore the Continuous Availability (**CA**) site, the Disaster Recovery (**DR**) site is located in Milano. The latter is solicited during emergency situations where the CA is not reachable. To provide a very high continuity of service, the two sites are constantly synchronized with each other. In 2015, in order to increase energy efficiency, reduce environmental impact and reduce maintenance costs, the Padova DC was completely redesigned. To evaluate the efficiency achieved, it is possible to use the Power Usage Effectiveness (**PUE**) parameter. With this one it is possible to assess the energy consumption related to resources not directly connected to the IT systems. The definition of PUE is shown in the Equation 4.1.

$$PUE = \frac{TotalFacilityPower}{ITEquipmentPower} = 1 + \frac{NonITEquipmentPower}{ITEquipmentPower} \quad (4.1)$$

It turns out to be a fundamental factor, so much so that IC was able to lower it through the renewal that took place in 2015. The ideal value would be unitary, however IC claims to have a  $PUE < 1.5$ , which is a good result. The following chapters introduce the structure of the IC network, and the implementation of the NAC within it.

### 4.2.1 IC NAC solution

Since InfoCamere is a reality covering the entire national territory, the technological infrastructure that compose it can be considered as of a high level of complexity. The IC network can be mainly divided into 2 sub-sets such as geographic and campus network. The geographic network represents the technological infrastructure that connects the customers of IC (most of these CCIAA). This



infrastructure is implemented through a Multiprotocol Label Switching (**MPLS**) network, where a private network is based on the public infrastructure. This MPLS therefore connects the local networks of the various offices of the chambers of commerce. Each of these, depending on the service purchased by IC, will have a certain architecture. In general, these may consist of different LANs (to provide different services such as data and VoIP traffic) defined on access devices (e.g., switch layer 2). These devices are therefore connected to L3 devices that provide connectivity to the MPLS network. In the last year, Software Defined WAN (**SD-WAN**) technology has also been implemented to offer an ever higher level of reliability and performance. The customers' local networks are therefore interconnected, through the MPLS, to the Padova/Milano DC. The core of the services is provided by the Padova DC, however the one located in Milano offers a High Availability (**HA**) service. In this way, IC offers a highly reliable service, identifying a disaster recovery facility in the Milano site should the Padova site be unreachable. The networks that are in production in the 4 InfoCamere branches (Padova, Milano, Roma and Bari) are called campus. In particular, in that of Padova the number of networks present is very high. In fact, there are portions of the network intended to the Wi-Fi infrastructure (with the relative Access Points, WLCs, dedicated monitoring services), VoIP, Multimedia and finally to that relating to the data center. The latter represents the beating heart of the provision of services. Within it, it is possible to find many technologies, ranging for example between the world of virtualization and those of computing and storage.

The MPLS network is implemented by relying on two different ISPs, to ensure an additional level of reliability. Each CCIAA is therefore equipped with a double connection to the MPLS network. These connections are called dorsal A and dorsal B. Through the BGP routing protocol the various local networks are communicated to the MPLS. A very important component that characterizes the Padova infrastructure is that relating to security. Multiple security devices (e.g., NGFW, IPS, etc.) are in fact located between the various network nodes. Part of this infrastructure allows NAC technology to monitor and manage access to the network. It is important to specify that NAC (implemented through Cisco ISE) is not yet in its final implementation. The purpose of this thesis is therefore to implement features not yet deployed that can bring benefits in terms of effort, costs and above all security. The studio's focus is therefore on the architecture implemented in the Padova site. Therefore, the architectures, both production and laboratory currently implemented, are presented below. It is important to underline that, for reasons related to the protection of sensitive data, the representations shown are a simplified version of those currently used. The architectures are presented in Figure 4.15 and Figure 4.16.

As far as the test environment is concerned, in the network laboratory there are more switches (from several vendors) that take on the role of NAD. These appliance is connected to a L3 switch, which has the task of routing the packets towards the core campus network appliances. Links in this case are implemented via Link Aggregation Control Protocol (**LACP**). This protocol allows the ad-

administrator to aggregate multiple physical links into a single logical link, thus increasing the total bandwidth. On the side of the data center there is Cisco Application Centric Infrastructure (**ACI**), which is a software defined solution for control in the DC environment. Through the spine-leaf architecture, the NAD is put in communication with ISE (virtualized through a vmWare solution).

As for the production environment, considering the third floor of the IC building in Padova, there is a more complex implementation than the previous one. In fact, from the internal IC network, traffic pass through the MPLS network (albeit for a single hop). Having redundant architecture, the connections lead to the 2 backbones of the two operators. Once traffic have passed these, it return back to the internal network of IC, where the first security checks are carried out through the NGFW present at the edge of the TLC room. This choice may seem weird at a first glance. In fact, why does the traffic generated by the Padova site have to pass through the MPLS geographic network to access the services, when it could access them directly? This choice was adopted to standardize the flow behavior of the 4 IC sites. Finally traffic arrive to the DC network, as in the case of the test environment.

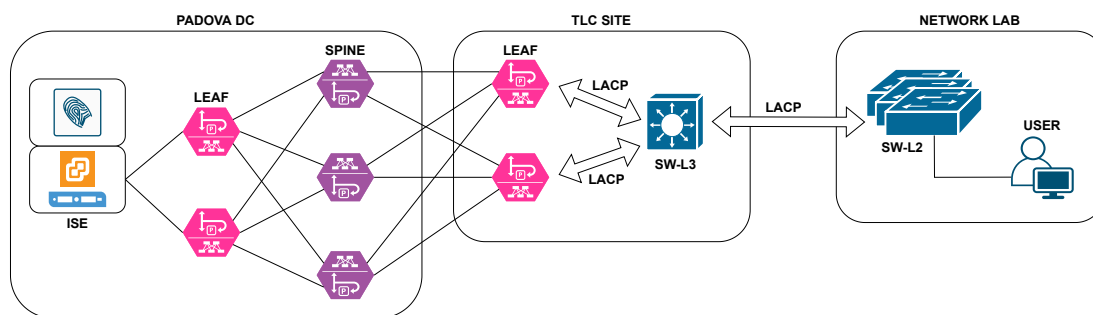


Figure 4.15: NAC architecture in test environment.

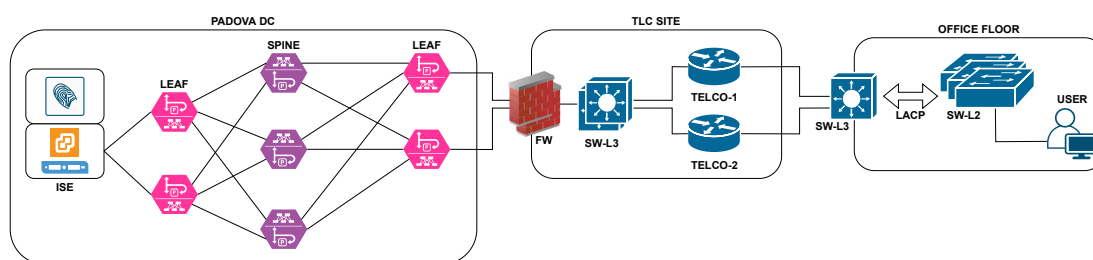


Figure 4.16: NAC architecture in production environment.

Regarding the type of deployment of the various personas on the various nodes, 2 different scenarios have been configured for the production and test environment. In the test environment there is a mixed deployment (standalone and small deployment), in which all the personas are implemented within a single (virtualized) node. This is because being a test environment, the different nodes would not be subject to a load of requests comparable to that present in a real

context. Furthermore, the fact that only one node exists facilitates configuration and troubleshooting during the design phase. Regarding the production environment, this is a completely different scenario. In fact it can be cataloged as a medium deployment. There are therefore two nodes, which assume the roles of MnT and PAN (respectively with primary and secondary crossed roles). The PSNs instead are implemented in 2 nodes which are currently not balanced. Note that from Figure 4.4, it is possible to see how the maximum number of PSNs that can be implemented in the context of IC is 5. For the sake of completeness, it is noted that the implementation of ISE, both for the test and production scenarios, via virtualization can also be seen through a packet capture. Paying attention to the exchange of messages between NAD and NAC server in Figure 4.17, it is possible to see how the MAC destination address appears to belong to VMware vendor.

```
> Frame 229: 313 bytes on wire (2504 bits), 313 bytes captured (2504 bits)
> Ethernet II, Src: Cisco_ (70:c9:c6: ), Dst: VMware_ (00:50:56: )
> Internet Protocol Version 4, Src: 10. , Dst: 10
> User Datagram Protocol, Src Port: 1813, Dst Port: 1813
> RADIUS Protocol
```

Figure 4.17: Virtualized ISE node.

## 4.3 Wireshark software

To perform a detailed analysis, in addition to the study of the protocols involved, it is necessary to analyze the actual traffic between the various nodes of the network. With both Cisco ISE and Aruba ClearPass it is possible to perform a tcpdump (i.e., a packet capture) on the interfaces involved. It is therefore possible to analyze RADIUS packets (and possibly other protocols) in order to conduct a complete analysis. In some scenarios, however, sniffing traffic only on the NAC server side is not enough. In fact, an important part of the problems concern the communication between NAD and NAC terminal. At this point, the free Wireshark software is of great help. Available for numerous platforms (*Windows, macOS, Solaris, etc.*), it provides multiple functions including:

- Deep packet inspection;
- Standard three-pane packet browser;
- Decryption support for many protocols such as SSL/TLS;
- Display filters.

During the study phases of the various protocols used (e.g., 802.1X and RADIUS) and that relating to configuration and implementation, Wireshark represented an excellent support and troubleshooting tool. Figure 4.18 shows an example of the Wireshark interface. In this case, as can be seen from the search bar located at

the top of the screen, a filter has been applied to the capture. This means that among all the packets that have been captured by the Network Interface Card (NIC), only those characterized by the filter values are shown. It is also possible to see how the search bar takes on the green color due to the validity of the applied filter. If the filter would not be compliant, the search bar would turn red. While in the upper part all the packets are shown, in the central one the single selected packet is shown in detail. Since the interface is tree structured, it is possible to analyze packets following the concept of encapsulation. The parseable fields are decoded internally by Wireshark, which then makes the information readable. From Chapter 5 onwards, the analyses and studies carried out are also presented with the aid of this fundamental tool.

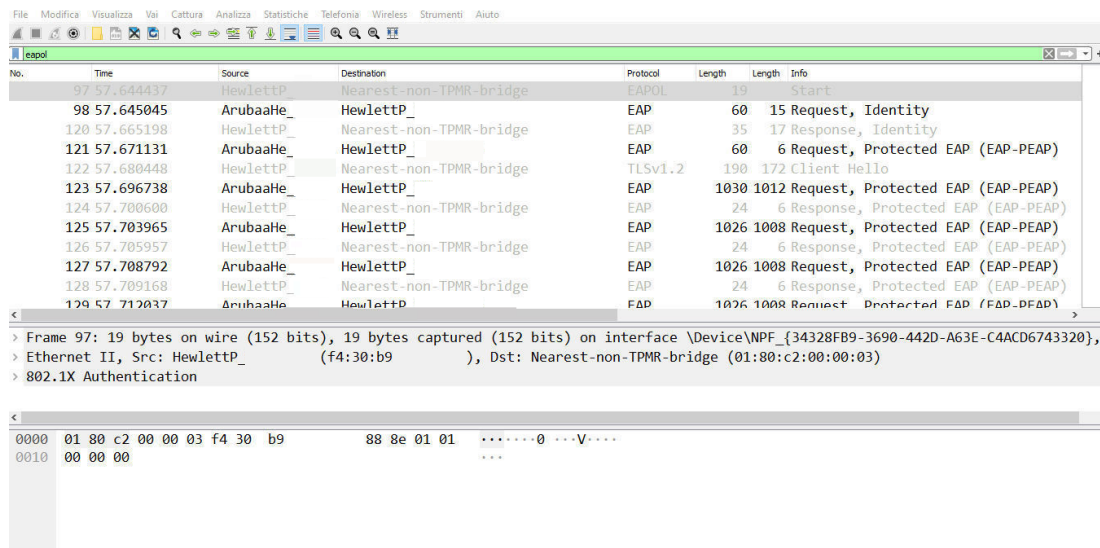


Figure 4.18: Wireshark’s interface.

This software is commonly used in combination with specific configurations on network devices called “*span ports*”. Through these, the network administrator is able to replicate all traffic passing through an interface to another one. By then connecting a device with Wireshark installed, on this interface, it is possible to analyze all the traffic. It is important to note that in addition to this software tool, it is possible to use physical ones. These solutions are the most popular within an enterprise environment. Known as ethernet (or fiber) taps, these devices operate at layer 1. These tools are preferred over direct configurations as they try to make the analyses operations transparent to the network traffic flow. Furthermore, relying on configurations, it would be necessary to dedicate interfaces (and therefore resources) for an external purpose, thus degrading the quality of the services offered. The taps therefore receive the traffic from an interface, and redirect it bit by bit on other physical interfaces on which the analyses are carried out using application software.

# Chapter 5

## AAA applied to NAC

To ensure secure access to the corporate network, NAC technology is necessary to set up an AAA service. Through authentication, authorization and accounting, a system is able to manage the accesses of multiple users/devices. The following paragraphs describe in detail the standards, protocols and frameworks used to implement the AAA service. In general 2 types of systems that implement authentication can be defined: centralized and distributed. In a centralized system, all policies, controls and decision-making power are concentrated in a single node. For these scenarios it is common to implement such a node through a cluster of multiple nodes. Logically, therefore, it would result in a single node, while physically it would be made up of several entities. Regarding the distributed deployment, the endpoints, via intermediate network nodes establish direct communication with the central node. This one could take therefore advantage of the right to rely on as many external nodes to validate access. Two of the most common examples of the latter are databases and ADs. A centrally designed architecture is characterized by:

- Simplified node management and monitoring;
- Flexibility and scalability;
- Need for high computational power.

Figure 5.1 shows an example of a centralized system. In analogy with what has been shown for the NAC architecture, also in this case the system is mainly made up of 3 components:

- User or endpoint;
- NAS (which corresponds to NAD in the NAC architecture);
- RADIUS server (which corresponds to NAC server in NAC architecture).

As mentioned several times, it is possible to adopt an external entity (identity store), which contains all the identities that are allowed to access. In an enterprise environment, such as that of IC, the strongly recommended solution to be adopted

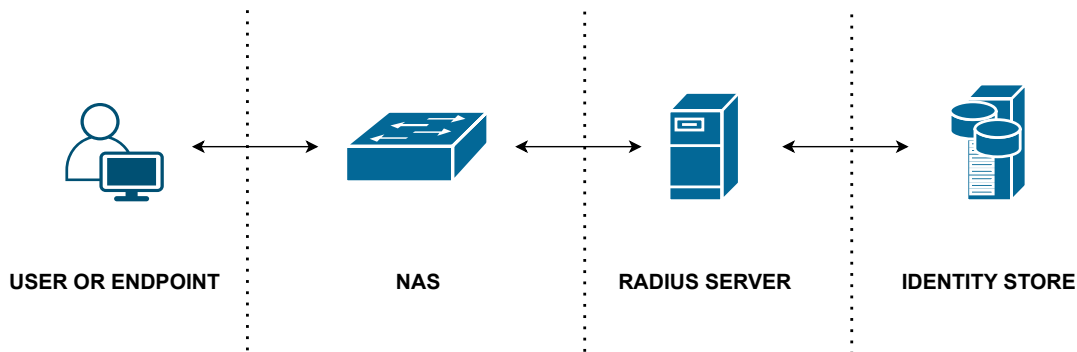


Figure 5.1: RADIUS centralized architecture.

is the centralized one. In fact, taking into consideration the ISE solution, this one implements a RADIUS server within it. In the event that a distributed system is chosen, it is necessary to use external application software. An example well established in such contexts appears to be FreeRADIUS.

## 5.1 Authentication

The first operation that a NAC system must carry out as soon as it identifies a new request is to authenticate whoever presents itself. Through this operation, access is therefore guaranteed to the applicant or not. The latter is therefore obliged to provide the authenticator with a valid identity in order to access. This identity can take many forms. Some of these are username and password credentials or certificates. Figure 5.2 shows the different types of authentication applications (some of which are analyzed in detail in the following sections).

It is important to underline that, during the design phase of a security solution, it is always necessary to analyze the trade off between the desired security level and the associated effort. This trade off is always represented in Figure 5.2, in which the 2 aspects are associated with the axes of the plane. A solution that is not very secure, requires limited support and almost zero effort (both from the point of view of time and costs). This type of scenario is present when implementing an open type solution. Obviously in an enterprise environment where the provision of services is the core business, it is strongly discouraged. On the contrary, a solution whose perimeter security level is extremely high translates into an equally high effort.

### 5.1.1 RADIUS

There are different standards and protocols that implement the AAA service. The most common are RADIUS and TACACS/TACACS+. In addition to these two, there is another protocol developed in the last decade, which is DIAMETER. The RADIUS protocol is based on a client/server model. The requests generated by the client are therefore addressed to a server, which will process them and

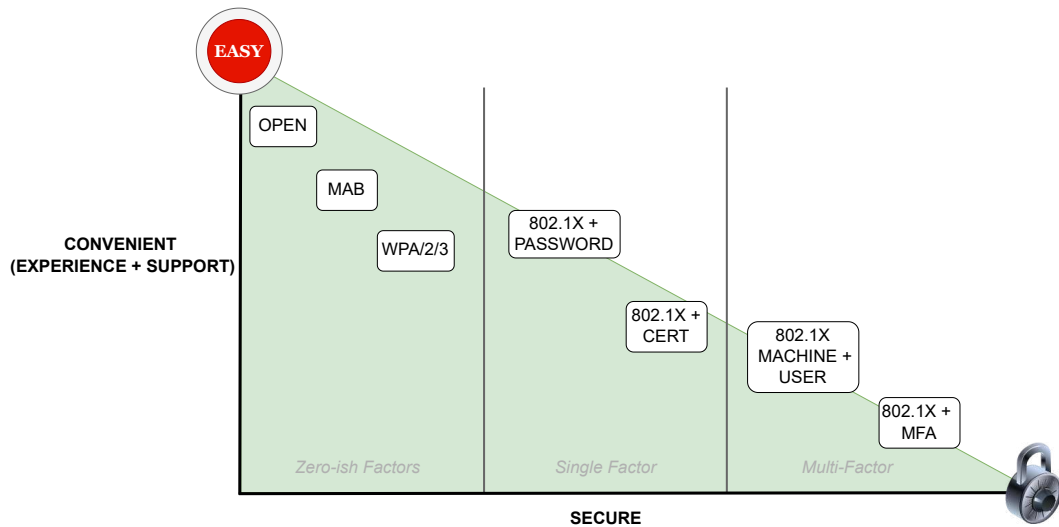


Figure 5.2: Authentication method.

generate the relative responses. In a context such as NAC, RADIUS is used together with 802.1X and EAP to allow end-to-end communication between end device and NAC server (Cisco ISE). The characteristics that distinguish it are the following:

- Open standard protocol;
- Based on UDP;
- Different port for authentication (1812) and accounting (1813) are used;
- Authentication and authorization process are combined;
- The messages are transmitted in clear text, in which only the password is encrypted (MD5);
- No external authorization of commands is supported;
- No multiprotocol support.

Figure 5.3 shows the formatting defined by the RADIUS protocol for a packet. The various fields of the packet are described below [9]:

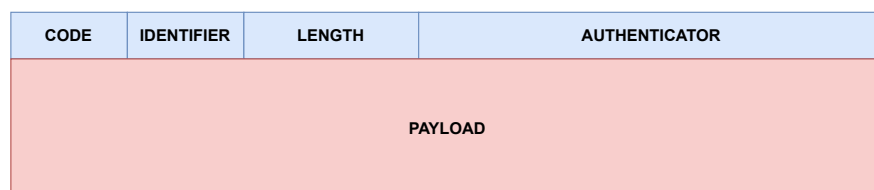


Figure 5.3: RADIUS packet.

- Code (1 octet): identifies the type of RADIUS packet. When a packet receives an invalid queue value, it is discarded without generating any feedback. The codes available for RADIUS packets are:
  - 1 Access-Request;
  - 2 Access-Accept;
  - 3 Access-Reject;
  - 4 Accounting-Request;
  - 5 Accounting-Response;
  - 11 Access-Challenge;
  - 12 Status-Server (experimental);
  - 13 Status-Client (experimental);
  - 255 Reserved.
- Identifier (1 octet): identifier used to identify requests and relative responses. RADIUS server is able to detect duplicate requests if it contains the same source IP address, same UDP source port and the same *Identifier* field value within a limited amount of time;
- Length (2 octets): indicates the length of the packet, including the *Code*, *Identifier*, *Length*, *Authenticator* and *Payload* fields. In the event that more bytes are received than those expressly indicated in this field, they must be treated as padding. If, on the other hand, the actual length of the received packet is less than that declared, the message must be discarded. This field has a minimum value of 20 bytes;
- Authenticator (16 octets): field used to authenticate the response received from the RADIUS server, and is also used in the password hiding algorithm. This value should be generated completely randomly and should be unique. This field can be classified in 2 different ways depending on the type of packet sent:
  - Request Authenticator: used in Access-Request packets, it is a unique and random value;
  - Response Authenticator: used in Access-Accept, Access-Reject and Access-Challenge packets, it is calculated starting from the concatenation of all fields together with the secret password.
- Payload: field containing the attributes with their associated values. Some of these are as follows:
  - 1 User-Name;
  - 2 User-Password;
  - 61 NAS-Port-Type;



79 EAP-Message.

The attributes are hence divided according to the following 3 groups:

- 0: Reserved;
- 1-191: Assigned or assignable by IANA;
- 192-240: Used for private purposes;
- 241-255: Reserved.

In the following chapters, also reporting some examples captured in a real environment, the RADIUS attributes used will be described.

During the authentication phase, different types of packets are used (with different codes). The Access-Request packets are generated by the client (in the case of a NAC solution from the NAD), and are sent to the RADIUS server. Within it are the attributes containing the information to request access. This packet should contain attribute number 1 (User-Name), while it must contain one between attributes 4 and 32 (NAS-Identifier). When attribute 2 is present, it must be hidden using the RSA Message Digest Algorithm MD5. Furthermore, for each Access-Request generated, the client attributes a new identifier value to the packet. The Access-Accept packets, on the other hand, are used by the RADIUS server to inform the client about the specific information needed to allow access. The identifier present in this packet is the same as that contained in the previously received request. Contrary to the previous message, when a RADIUS server receives a request that is not compliant, it responds to them with an Access-Reject message. Another type of message that the RADIUS server can use following an Access-Request is the Access-Challenge message. Other types of packets such as Accounting-Request and Accounting-Response are used for the accounting phase. During this phase, information is exchanged to determine the operations performed by the user once access to the network has been obtained. As reported below, this type of packet can also be used, for example, for the profiling function. Obviously, since the accounting packets are carriers of information with respect to the client, they are used once the user has been authenticated and authorized. As for the *Payload* field, this contains the actual information exchanged between client and server. The formatting with which this information is structured follows the concept of Attribute Value Pair (**AVP**). It should be noted that it is often preferred to use Type Length Value (**TLV**) instead of the term AVP. However, both refer to the same type of data, and therefore to the attributes contained in the payload of the RADIUS packet. This formatting is represented in Figure 5.4. A capture made with Wireshark in Figure 5.5 is also proposed, from which it is possible to see how the 2 terms are used to analyze the various attributes. As can be seen from the Wireshark capture reported in Figure 5.5, the RADIUS message is encapsulated within the UDP protocol. A representation that schematizes the encapsulation process is shown in Figure 5.6.

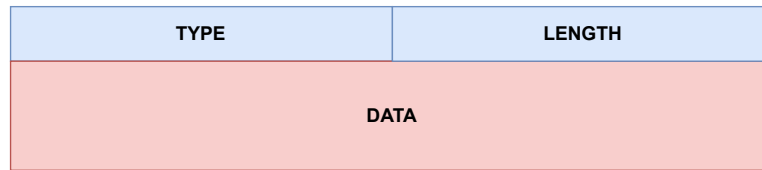


Figure 5.4: Radius AVP.

```

> Frame 21: 180 bytes on wire (1440 bits), 180 bytes captured (1440 bits)
> Ethernet II, Src: aa:bb:cc:80:64:00 (aa:bb:cc:80:64:00), Dst: PcsCompu_55:99:d4 (08:00:27:55:99:d4)
> Internet Protocol Version 4, Src: 192.168.10.10, Dst: 192.168.10.150
> User Datagram Protocol, Src Port: 1645, Dst Port: 1645
v RADIUS Protocol
  Code: Access-Request (1)
  Packet identifier: 0x1b (27)
  Length: 138
  Authenticator: d393e91143f1fd182d285c0607e97ef0
  [The response to this request is in frame 22]
v Attribute Value Pairs
  > AVP: t=User-Name(1) l=7 val=cisco
  > AVP: t=Service-Type(6) l=6 val=Framed(2)
  > AVP: t=Framed-MTU(12) l=6 val=1500
  > AVP: t=Called-Station-Id(30) l=19 val=AA-BB-CC-00-64-00
  > AVP: t=Calling-Station-Id(31) l=19 val=08-00-27-6E-C5-50
  > AVP: t=EAP-Message(79) l=12 Last Segment[1]
  > AVP: t=Message-Authenticator(80) l=18 val=ffed203600d285feed8b722fe3e065fe
  > AVP: t=NAS-Port-Type(61) l=6 val=Ethernet(15)
  > AVP: t=NAS-Port(5) l=6 val=50000
  > AVP: t=NAS-Port-Id(87) l=13 val=Ethernet0/0
  > AVP: t=NAS-IP-Address(4) l=6 val=192.168.10.10

```

Figure 5.5: AVP/TLV fields in a RADIUS packet.

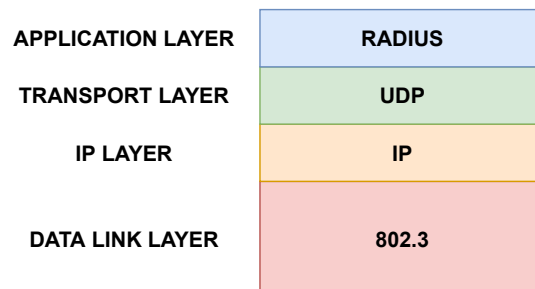


Figure 5.6: Encapsulation of a RADIUS message.

One of the attributes, as mentioned above, is reserved for the transmission of the password from client to server. The procedure by which the password is encrypted is shown below:

- 0 The password to be encrypted is taken as input;
- 1 Password is padded to reach a total length of 16 bytes;
- 2 Generation of 16 bytes of hash (MD5) using the secret and the value related to the *Authenticator* field of the Access-Request packet;
- 3 Execution of the *XOR* between the plain text password and the result of the operation obtained in step 2.

If the password is longer than 16 bytes, another 2 steps are added to step 3:

- 4 Generation of a hash between the secret and the result obtained in step 3;
- 5 Execution of the *XOR* between the hash obtained in step 4 with the remainder of the password provided by the client.

### RADIUS attributes

For the transmission of information, both client and server, use the attributes defined by the RADIUS protocol. While some of these are fundamental, others are highly optional. As an introduction to the following chapters, the most significant attributes are presented in Table 6.2.

Number	IETF Attribute	Description
1	User-Name	Indicates the name provided by the terminal to authenticate towards the RADIUS server.
2	User-Password	Indicates the password used by the device to authenticate to the RADIUS server. The process by which it is calculated is described in the previous paragraph.
4	NAS-IP-Address	Indicates the IP address of the NAS (or NAD) that is requesting the authentication service.
5	NAS-port	Indicates the physical port number towards which the end device is authenticating. It is a field made up of 32 bits, in which, depending on the configuration of the NAS, all or only half of them are used.
6	Service-Type	One of the fundamental fields at the base of NAC, indicates the type of service required to authenticate from the end device. Some of the values that it can assume are listed below: <ul style="list-style-type: none"> <li>• Login;</li> <li>• Framed;</li> <li>• Callback Login;</li> <li>• Callback Framed;</li> <li>• Call Check.</li> </ul>

11	Filter-ID	Indicates the name of the filter list for the user session. This name can be set inside the Access-Accept packets.
18	Reply-Message	Contains text that the server can send to the end device of its choice.
26	Vendor-Specific	Another very important attribute, with which vendors can use specific attributes. By way of example, the attributes related to Cisco are characterized by a Vendor-ID equals to 9, while the supported option has the value of 1.
30	Called-Station-Id	Indicates the MAC address of the network interface that received the authentication request on the NAS.
31	Calling-Station-Id	Indicates the MAC address of the network interface that made the authentication request.
64	Tunnel-Type	Indicates the type of tunneling protocol used. Some examples are listed below: <ul style="list-style-type: none"> <li>• Point-to-Point Tunneling Protocol (<b>PPTP</b>);</li> <li>• Virtual Tunneling Protocol (<b>VTP</b>);</li> <li>• Generic Route Encapsulation (<b>GRE</b>);</li> <li>• VLAN.</li> </ul>
65	Tunnel-Medium-Type	Indicates the means of transport used to create the tunnel. Generally the value of this field is IP (which is also the default one).
79	EAP-Message	Field containing EAP packets. Depending on how the NAS/NAD works, EAP packets can be translated into RADIUS attributes, or they can be directly encapsulated.

81	Tunnel-Private-Group-ID	Indicates the group ID for a particular tunneled session. As shown in the following sections, attributes 64, 65 and 81 play an essential role in the authorization phase.
----	-------------------------	---

Table 5.1: Main RADIUS attributes.

One of the aspects to be explored about the RADIUS protocol is the way in which transmissions take place. In fact, each packet is sent without being encrypted in any way, allowing a malicious user to intercept the traffic and sniff it without a particular effort. Instead, by using attribute 2, the value associated with it is given processed together with an MD5 hash. A threat actor could therefore intercept RADIUS packets containing useful information to be used for a possible attack such as User-Name, NAS-Identifier and so on. Another possible attack, having a particularly devastating impact, is the ability to modify a packet without any effort. This attack corresponds to the MITM attack.

However, communications can be secured using TLS tunnels (described in Section 5.1.6). This option is present for example in ISE where it is possible to enable Datagram Transport Layer Security (**DTLS**). This excludes the possibility of being subject to attacks such as eavesdropping, tampering or message forgery. To enable this possibility, it is however necessary to make the appropriate changes also on the network devices (NAD). For more information refer to Section 8.2.3

### 5.1.2 TACACS+

Another protocol that implements the AAA service is the Terminal Access Controller Access Control System plus (**TACACS+**) [10]. An evolution of the previous TACACS, this protocol was developed by Cisco in the 90s and initially released as an open standard. Both TACACS+ and RADIUS can be used to manage network access. However, compared to RADIUS, TACACS+ is commonly used for managing access to network devices. This reason stems from an important difference in the functions provided by these two protocols. In fact, TACACS+ is able to separate the authentication and authorization processes. Therefore, every command sent by the administrator to the network device is redirected to the TACACS+ server, which verifies its authorization. Also using external ADs, each user is associated with a certain level of privilege with which he can use a limited set of commands. Another very important difference lies in the fact that RADIUS is not able to generate logs relating to the commands sent by the network administrator during the configuration/troubleshooting phase. If there were two administrators connected to the same device, it would be impossible to distinguish which of the commands are associated with the respective senders. In Figure 5.7 the 2 different functioning processes are shown.

Other important characteristics that distinguish the TACACS+ protocol from the RADIUS one are the following.

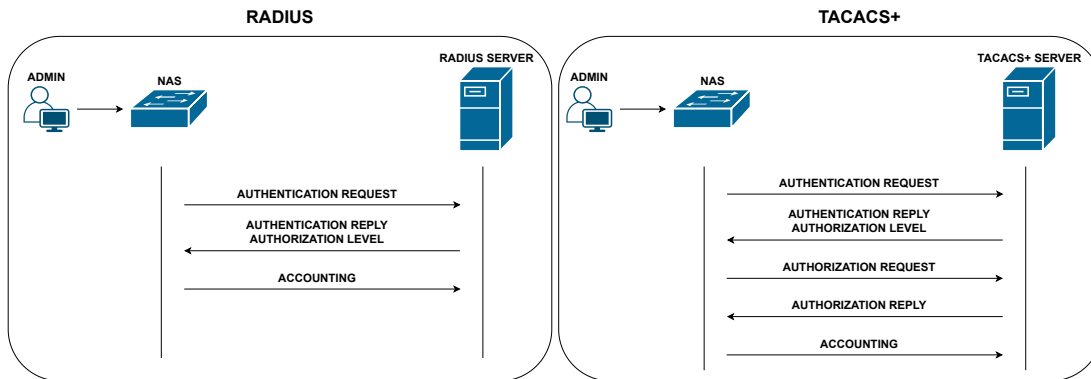


Figure 5.7: RADIUS and TACACS+ flow comparison.

- Use TCP port 49;
- Native encryption support for the entire payload, thus hiding sensitive information;
- 15 privileged modes supported.

Figure 5.8 shows the TACACS+ packet. The various fields are briefly described below:

- Major version: field that specify the major TACACS+ version number;
- Minor version: field that specify the minor TACACS+ version number;
- Type: indicates the type of packet. It can take the following values:
  - 0x01: Authentication;
  - 0x02: Authorization;
  - 0x03: Accounting.
- Sequence number: contains the number associated with each packet for a session. The first packet of the session must contain the value 1, while subsequent packets will increment this value. This operation implies that the TACACS+ client only sends packets containing odd sequence number values, while the TACACS+ server only sends even values;
- Flags: field containing some flags useful for transmission. In particular, one of these (unencrypted flag) informs whether encryption is used on the body of the packet or not. This flag (i.e., “*TAC\_PLUS\_UNENCRYPTED\_FLAG*”) assumes binary values (i.e., 0 or 1). When it is set to 0 it means that the payload is encrypted, while when it is set to 1 it means that the payload is in clear text;
- Session ID: field containing the identifier of the session that is generated randomly. This value therefore does not change for the entire duration of the session;

- Length: contains the total length of the packet payload (therefore excluding the header);
- Body: field that contains the actual data.

MAJOR VERSION	MINOR VERSION	TYPE	SEQUENCE NUMBER	FLAGS
SESSION ID				
LENGTH				
BODY				

Figure 5.8: TACACS+ packet.

Communications using TACACS+ use messages of different types, depending on the required action. The following section briefly introduces these different types of messages.

### TACACS+ message

TACACS+ implements different types of messages according to the service considered. During the authentication phase, 3 types of messages are used:

- Start: packet used to initialize the authentication request between client and server;
- Reply: packets used by the AAA server to communicate with the AAA client;
- Continue: type of packets sent by the client in response to server requests.

An example of the use of these messages is shown in Figure 5.9. The last reply message contains the result of the authentication request. This can be broken down into 4 possible values:

- Accept: the identity provided by the client is valid, and the authorization process can begin (in case the client is configured accordingly);
- Reject: authentication failed. The user may be banned, or may use another attempt depending on the NAS configuration;
- Error: a generic error occurred during the process;
- Continue: message sent via a reply packet to request additional information from the client.

As for authentication, there are also 2 different types of messages for the authorization phase:

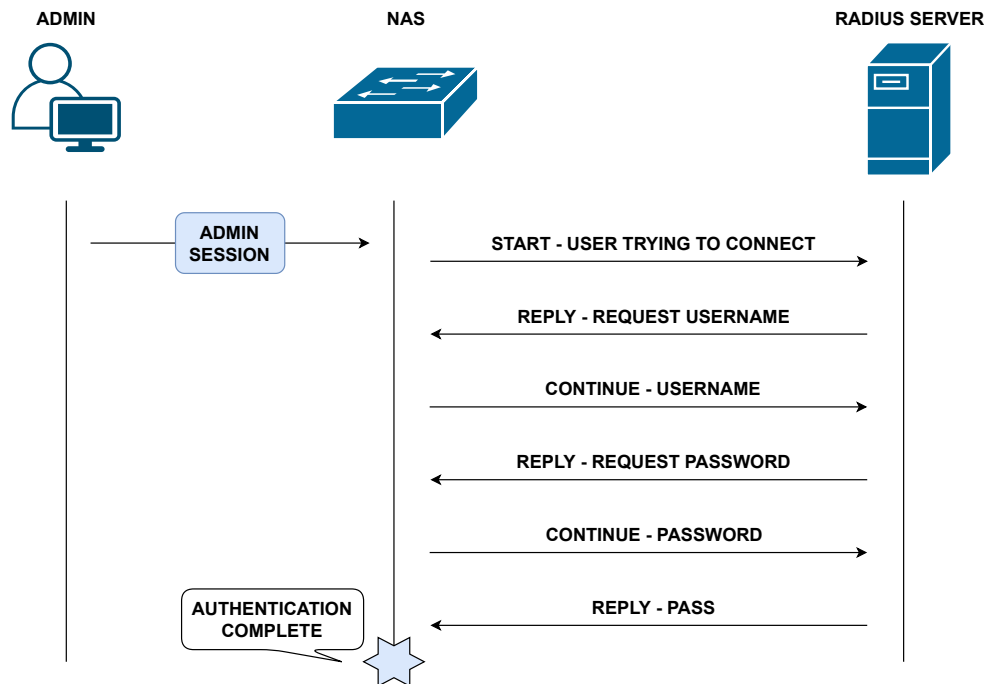


Figure 5.9: TACACS+ authentication process.

- Request: this message is sent from the client to the server to request authorization. This type of service is used to restrict the range of access to the various commands within a network device. These commands are sent from the client via the AV pair (described above);
- Response: this message is used by the server to respond to requests made by the client. The outcome of these requests is divided into:
  - Fail: the user is not authorized to access the requested service (e.g., user cannot use the specific command requested);
  - Pass\_add: the authorization has been granted, and the additional information contained in the packet must be used to provide the service;
  - Pass\_repl: permission granted, but the request is ignored;
  - Follow: the server informs the client that authorization must be requested from another server. The information on how to find it is inserted in the response message;
  - Error: a generic error occurred during the process.

### 5.1.3 802.1X

In order to authenticate a user correctly, it is necessary to use an ad-hoc protocol. This is where 802.1X protocol comes in [11] [12]. NAC technology therefore also relies on it to manage access by devices/users. 802.1X is a PNAC defined in the early 2000s. With the continuous evolution of technology, 3 successive new



releases have been presented, each of which introduces improvements both from the point of view of the security, and the features offered. Basing its operation on the concept of port, or rather on the Port Access Entity (**PAE**) associated with the port, the protocol defines 3 main components (similar to those proposed in the previous chapters):

- Supplicant (NAC terminal): terminal that wants to access the services within the LAN;
- Authenticator (NAD): network device, intermediary between supplicant and authentication server, whose role is to redirect access requests presented by the supplicant to the server. It also has the task of applying the restrictions to the terminals/users indicated by the server;
- Authentication server (NAC server): device in charge of checking the validity of the identities received, and assigning the respective authorizations.

The PAE can be either supplicant or authenticator. It is important to underline that a port can be associated with both. In this case, for the Controlled Port (**CP**) to be authorized, both PAEs must be. Terminal access is therefore managed at the port level. Each port is logically divided into 2 entities:

- Uncontrolled Port (**UP**): port having full access to the network;
- Controlled Port: port having access to the network based on the status of the *AuthControlledPortStatus* variable. However, this port only allows access to the network to a specific type of packet, that is to the EAP packets. The EAP protocol is described in Section 5.1.5.

Each frame received by the physical interface is made available to both ports. The system, composed of these two logical entities, is shown in Figure 5.10. In

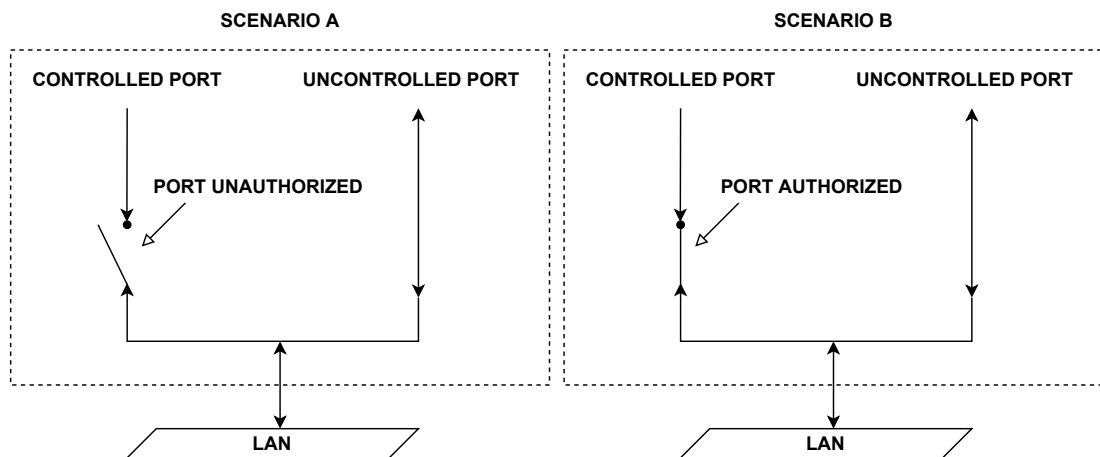


Figure 5.10: Controlled and uncontrolled port states.

the latter also the results of the different status of the *AuthControlledPortStatus*

variable are shown. In scenario A, the variable takes the value of *unauthorized* and therefore the port is disabled. In scenario B, the state is *authorized*, and therefore the port can let the traffic pass. Another parameter relating to the operating status of the port is *AuthControlledPortControl*. This parameter can override the value of the other variable. The 3 values it can take are:

- ForceUnauthorized: this state forces the PAE to set the value of the *AuthControlledPortStatus* parameter to *unauthorized* value (i.e., the controlled port is always unauthorized);
- Auto (default value): this state allows the PAE to manage the value of the *AuthControlledPortStatus* variable in such a way as to reflect the authentication status between supplicant, authenticator and server;
- ForceAuthorized: this state forces the PAE to set the value of the *AuthControlledPortStatus* parameter to *authorized* value (i.e., the controlled port is always authorized);

There is also the possibility of disabling access to the network not only from a logical point of view (via the CP), but also from a physical point of view. Figure 5.11 shows how in scenario A both ports can access the network, while in scenario B none of the 2 is able to communicate with other systems within the network. This is possible thanks to the configuration of the layer 2 status parameter. There are also two other parameters used to limit the traffic that

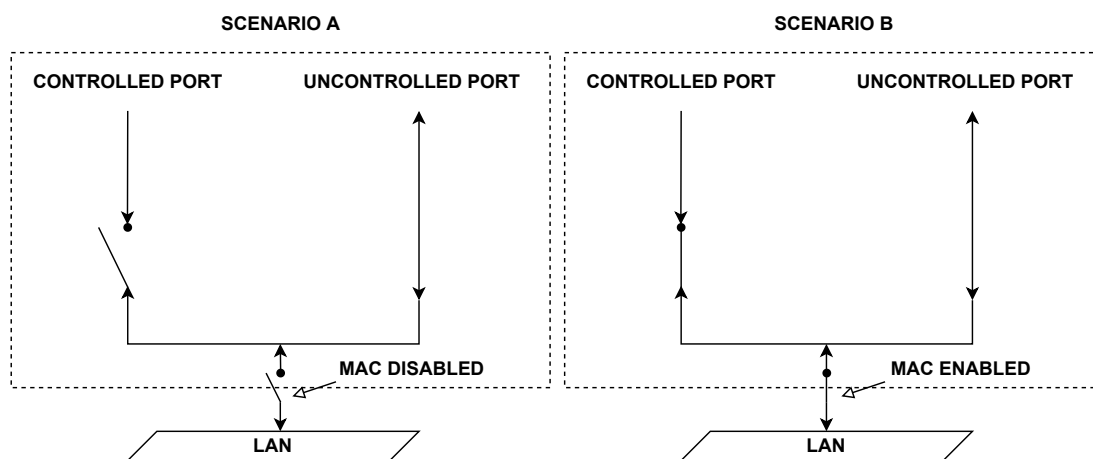


Figure 5.11: Effect of MAC states.

can affect the traffic. These parameters define the direction of the traffic to be submitted to the controlled port. The parameters are *AdminControlledDirections* and *OperationalControlledDirections*. The latter then determine whether the CP in which it is in *unauthorized* state is allowed to exercise control over network traffic in both directions (i.e., in and out), or only for incoming traffic. The value of the *AdminControlledDirections* parameter can only be changed by the administrator. The authentication process, with the relative description of the various phases of the flow, is shown in the following sections.

### 802.1X packet

Figure 5.12 shows an example of 802.1X PDU. The 802.1X packet consists of

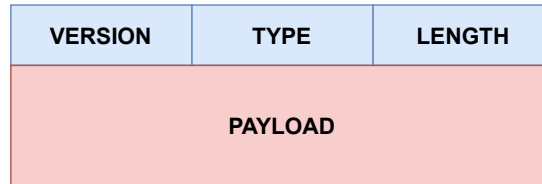


Figure 5.12: 802.1X PDU.

the following fields:

- Version (1 octet): field containing an unsigned binary number. This field informs the receiving node which version of the 802.1X standard (version of the EAPoL protocol). As can be seen from Figure 5.13, in the real case of IC the supported version is *802.1X-2001*. Available versions are the following:
  - 0x01 *802.1X-2001*;
  - 0x02 *802.1X-2004*;
  - 0x03 *802.1X-2020*.
- Type (1 octet): unsigned binary value, determines the type of packet transmitted. The different packets are as follows:
  - 0 EAP-Packet;
  - 1 EAPoL-Start;
  - 2 EAPoL-Logoff;
  - 3 EAPoL-Key (used only when the key transmission functionality is supported);
  - 4 EAPoL-Encapsulated-ASF-Alert;
  - 5 EAPoL-MKA;
  - 6 EAPoL-Announcement (Generic);
  - 7 EAPoL-Announcement (Specific);
  - 8 EAPoL-Announcement-Req.
- Length (2 octets): unsigned binary value, defines the length of the payload (EAP packet) in number of bytes. If it assumed the value 0, then the payload would be empty. This case is observed when the supplicant sends an EAPoL-Start message;
- Payload: this field is present only in type 0, 3 and 4 messages. Furthermore, based on which of these 3 messages is used, the payload will contain different messages (current study focuses on type 0 message);

```

> Frame 120: 35 bytes on wire (280 bits), 35 bytes captured (280 bits) on interface
> Ethernet II, Src: HewlettP_ (f4:30:b9: ), Dst: Nearest-non-TPMR-t
  802.1X Authentication
    Version: 802.1X-2001 (1)
    Type: EAP Packet (0)
    Length: 17
  Extensible Authentication Protocol

```

Figure 5.13: EAPoL packet captured in IC scenario.

Finally, the maximum total length of the EAPoL packet depends on the type of transmission medium used (e.g., IEEE 802.3 or 802.11). As detailed in Section 5.1.5, the 802.1X protocol is used in conjunction with EAP to implement NAC technology. It is therefore important to underline that in the literature, the packet containing the 802.1X protocol header and the payload formed by the EAP packet is called EAP over LAN (**EAPoL**). Consequently, to be aligned with the terminology used in the literature, this thesis refers to the packet shown in Figure 5.12 with the term EAPoL. A real representation of an EAPoL packet is presented in Figure 5.13. As can be seen, the payload of the packet corresponds to the EAP packet (described in Section 5.1.5). In addition to the definition of the EAPoL packet, it is necessary to define some parameters at the level below. As shown by Figure 5.14, it is necessary to define parameters inside the Ethernet header such as destination MAC and the type of protocol used.

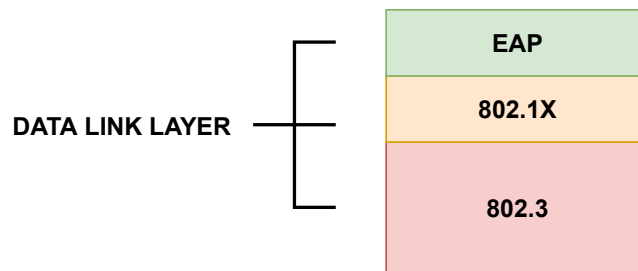


Figure 5.14: Encapsulation of a 802.1X message.

```

> Frame 120: 35 bytes on wire (280 bits), 35 bytes captured (280 bits) on interface \Dev
  Ethernet II, Src: HewlettP_ (f4:30:b9: ), Dst: Nearest-non-TPMR-bridge
    Destination: Nearest-non-TPMR-bridge (01:80:c2:00:00:03)
    Source: HewlettP_ (f4:30:b9: )
    Type: 802.1X Authentication (0x888e)
  802.1X Authentication
    Version: 802.1X-2001 (1)
    Type: EAP Packet (0)
    Length: 17
  Extensible Authentication Protocol

```

Figure 5.15: PAE MAC destination address example.

The protocol type defined within the Ethernet header corresponds to the hexadecimal value *0x888E*. As regards the destination MAC address, the standard defines the use of the address *01-80-C2-00-00-03* (address of the PAE). This information can be seen in Figure 5.15, inside the header of the layer 2 frame. Note that no information about the IP packet has been reported as, in this case, the network layer and above are not used. The 802.1X protocol appears to be a necessary tool for the authentication process. However, in some scenarios, some devices do not support this protocol. All these devices such as printers, IP phones, stampers, etc. are part of these scenarios. A solution to this problem is the Mac Authentication Bypass (**MAB**).

For information purposes, Figure 5.16 shows the list of MAC addresses relating to a certain VLAN (used in the test environment for the 802.1X configuration). It is possible to see that the address previously mentioned is present. Furthermore, the interface (or the VLAN) on which the MAC address is associated has been assigned by the CPU (i.e., by the internal processor that manages the 802.1X).

```
#sh mac address-table vlan 200
Mac Address Table
```

Vlan	Mac Address	Type	Ports
All	0100.0ccc.cccc	STATIC	CPU
All	0100.0ccc.cccd	STATIC	CPU
All	0180.c200.0000	STATIC	CPU
All	0180.c200.0001	STATIC	CPU
All	0180.c200.0002	STATIC	CPU
All	0180.c200.0003	STATIC	CPU

Figure 5.16: PAE MAC destination address in test environment.

### 802.1X authentication mode

Authentication using 802.1X serves a high-risk function. In an enterprise environment, where the number of devices attempting to access the network is in the thousands, it is very risky to carry out radical configurations. It is therefore necessary to follow an implementation starting from a relaxed approach, to then reach the goal of configuring a more strict type of authentication. There are 3 types of implementation that distinguish network access:

- Monitor mode;
- Low-Impact mode;
- Closed mode.

These are described in Section 5.1.7.



Figure 5.17: Service-Type based on the vendor.

### 5.1.4 MAB

Unlike 802.1X, MAB is not a protocol. There is therefore no guide document to follow to implement it on network devices. Each vendor then implements this functionality according to its own description. A practical example of these different implementations lies in the type of service reported with one of the RADIUS attributes described above: Service-Type. This, according to the vendor that implements the NAD, assumes 3 and more different values:

- Service-Type = Framed;
- Service-Type = Login;
- Service-Type = Call-Check (Cisco).

From Figure 5.17 it is possible to see how the different values for the Service-Type attribute are shown from the ISE interface. In particular, in the left figure there is the specific attribute for the BrocadeWired vendor, while on the right for the Cisco vendor.

It is important to note that it is very easy to confuse the values of the Service-Type attribute if not enough attention is given to the vendor of the specific device that acts as a RADIUS client. For example, for Cisco and Aruba devices, the value for attribute number 6 that identifies authentication using 802.1X is “*Framed*”. It may therefore arise to ask the motivation for the use of different nomenclature. Regarding Cisco, it has adopted a different name due to security reasons.

MAB solution is adopted in all those cases where the devices do not support any supplicant (802.1X agent). In this way, a properly configured switch would not receive any response to the request identity messages sent by himself. This behavior would result in an authentication timeout, thus denying access to the requester. Devices such as the aforementioned printers, IP phones, IoT devices or headless devices would be isolated. Initially it was thought to dedicate ports that did not require authentication to such devices. Doing so, by connecting a PC to one of these types of ports it would have been possible to enter the network without having been subject to any policy. With the MAB, the username used to authenticate the device is the respective MAC address. In this way, NAD creates a RADIUS Access-Request packet containing the MAC address as the Username attribute. This value is then used to check the validity of the user. If the MAC address is registered, the ISE server responds with an Access-Accept message. A timeout mechanism was previously mentioned whereby, if the NAD does not receive a response within a certain period of time, it considers the device free

of supplicant. Generally, the NAD sends EAPoL packets every 30 seconds. If after 90 seconds (3 attempts) it does not receive any response, it understands that the terminal does not implement any supplicant. It is important to note (as shown in the configurations), that parameters such as timers or re-transmission attempts can be reconfigured by the administrator. Once the last timeout has expired, if configured correctly, the NAD accepts the first packet to inspect it. For some types of packets such as Spanning Tree Protocol (**STP**), Dynamic Trunking Protocol (**DTP**) and Link Layer Discovery Protocol (**LLDP**), the NAD may not be able to obtain the MAC address. However, this behavior is strictly related to the platform used. From this it obtains the source MAC address, and inserts it in the RADIUS Access-Request packet. Any other packets after the first are discarded anyway. However, timeout-based operation must be carefully configured. If the timeout is too high, the device would be disconnected from the network for too long, causing a significant disservice. Next, the authentication process similar to that relating to 802.1X is shown. MAB can be configured mainly in 2 ways:

- Standalone: only method used for authentication;
- Fallback: MAB is used as a backup mechanism. First 802.1X is used, while if it fails, MAB is used.

Using MAB, it is easy to understand how network resources are exposed to simple attacks, such as MAC spoofing. In the implementation of MAB between the various vendors, it is possible to configure a security feature with which it is possible to disable, shutdown or restrict the port in case of anomalous behavior.

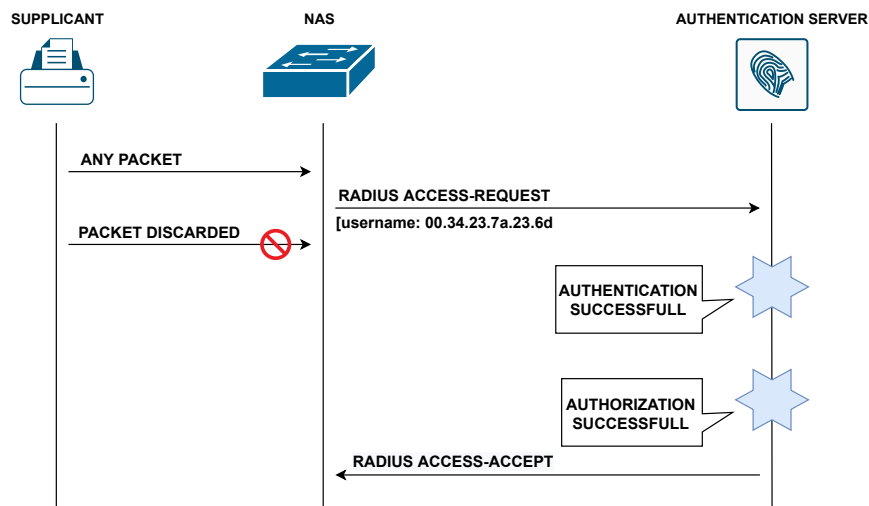


Figure 5.18: MAB authentication process concept.

In Figure 5.19 it is possible to see the behavior of the MAB functionality in a real scenario (IP addresses hidden for security reasons). In this case the switch is configured in such a way that after 3 re-authentication attempts it uses the MAB mode. Therefore, after a first message, and 3 subsequent attempts, a failure

is arised. At this point the switch extrapolates the MAC address of the device from the first useful packet and sends it as username to the NAC server (ISE) through a RADIUS packet. The NAC server finally responds, allowing access to the network.

No.	Time	Source	Destination	Protocol	Info
1036	11.479005	Cisco	Nearest-non-T...	EAP	Request, Identity
2070	21.478326	Cisco	Nearest-non-T...	EAP	Request, Identity
3254	31.478403	Cisco	Nearest-non-T...	EAP	Request, Identity
4336	41.478611	Cisco	Nearest-non-T...	EAP	Request, Identity
6783	51.481051	Cisco	Nearest-non-T...	EAP	Failure
6784	51.482404	10.	10.	RADIUS	Access-Request id=195
6800	51.549072	10.	10.	RADIUS	Access-Accept id=195

Figure 5.19: MAB authentication process in a real scenario.

Figure 5.18 shows the authentication process used by MAB. Note that any packet received prior to a RADIUS Access-Accept by ISE is discarded. One of the main problems in implementing the MAB mode lies in the management of devices. Whenever it is necessary to allow access to a new device, it is requested to add its MAC address to the identity store. It is clear how, in such a system, scalability is quickly lacking. However, it is possible to guarantee access to the network by not knowing the MAC address. This option is not recommended due to the total absence of access management. To overcome this problem, the network administrator can implement the profiling function (discussed in Chapter 6). Through the latter, the NAC server is able to understand which device it is trying to access (with satisfactory accuracy). In this way, recognizing the type of device, it is able to authenticate it and assign to it a certain level of authorization. The most common methods of obtaining information is via RADIUS and DHCP. For example, using RADIUS packets, the authentication server is able to use the attributes contained within it to manage access. At this point it is necessary to make a premise regarding the structure of a MAC address. Each MAC address consists of 2 components:

- Organizationally Unique Identifier (**OUI**): the first 6 bytes are reserved to identify the manufacturer of the device. Each vendor therefore has a list of OUIs that is assigned directly by the IEEE;
- NIC specific portion: the latter 6 octets are instead specific for the device.

By collecting information from the network traffic generated by the device, NAC server is able to understand what type of device it is. In fact, certain vendors are associated with the production of certain devices. So when a device with a certain OUI is identified, it can be profiled as a certain device type. Figure 5.20 shows the representation of a generic basic authentication process. Its implementation (with the addition of some features) is shown in Section 5.4.

### 5.1.5 EAP

EAP is a layer 2 protocol that provides extensibility for authentication methods. It defines the transport and usage of identity credentials for authentication



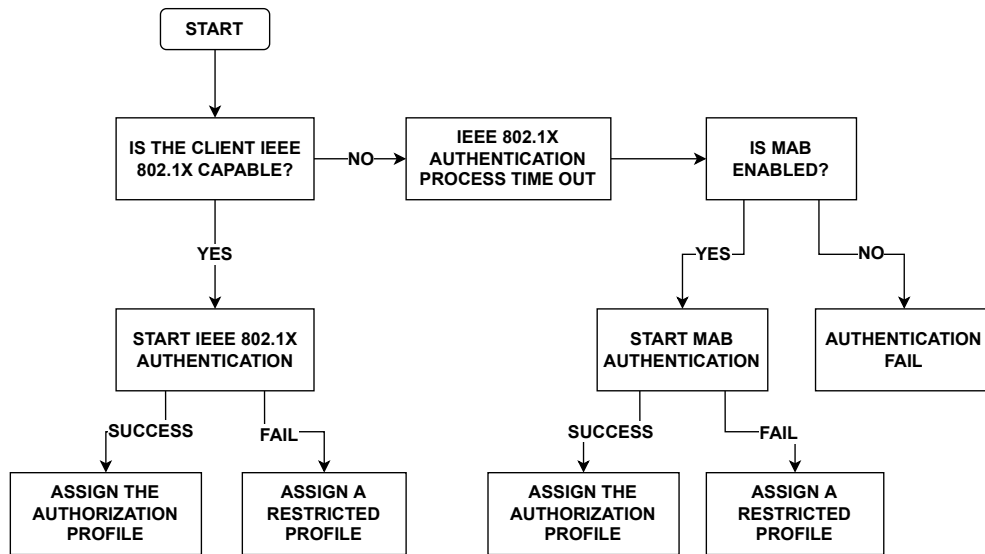


Figure 5.20: Authentication flowchart.

process [13]. It is important to note that EAP is not an authentication method such as MS-CHAPv2. Within a message of this type there are therefore usernames, certificates, tokens and so on. As with the 802.1X protocol, EAP also assumes the role of the “*de facto*” standard for the authentication function. Furthermore, EAP is used only for the authentication process and not, for example, for the authorization process. For the implementation of the protocol it is necessary the presence of 3 entities previously described: supplicant, NAS and authentication server. The connectivity between the supplicant and the NAS can be both wired and wireless. In the first case, a special term is used to indicate the 802.1X packet containing the EAP message, that is, EAPoL. In Figure 5.21 it is possible to see the components that compose it with the respective types of traffic.

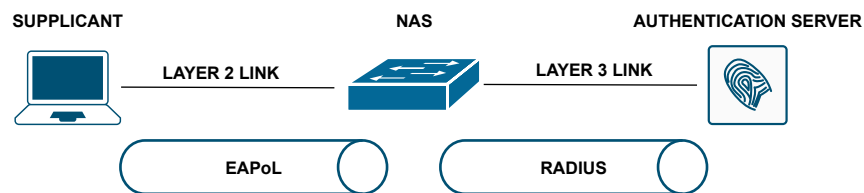


Figure 5.21: Components of 802.1X.

While in the left part there is an EAPoL communication, in the right part RADIUS is used. The NAS therefore uses 2 protocols to interface with the 2 entities. However, the authentication process is completely transparent to the NAS. In fact, it has the sole task of encapsulating (or translating) the EAP packets in the RADIUS messages to be sent to the authentication server. The NAS therefore assumes 2 different behaviours based on the configuration adopted. Depending on the vendor, these 2 approaches take on different names. These are:

- Termination mode (Finish mode): in this mode, the NAS receives EAP packets and translates them into RADIUS attributes. From these, the final RADIUS packets are created to be transmitted to the authentication server;
- Relay mode (Transfer mode): unlike the previous one, the EAP packets are not translated but rather encapsulated directly inside the RADIUS packets. In this case, among the attributes present in the RADIUS packet, can also be noted the one identified by the number 79. In this scenario, these types of packets are called EAP over RADIUS (**EAPoR**). For information, the finish mode is native to Cisco devices.

It is possible to see from Figure 5.22, that attribute 79 (EAP-Message) is present inside the RADIUS packet containing a real access request captured via Wireshark in InfoCamere. This means that the device (in this case an Aruba switch) is configured to operate via EAPoR. Some of the fields have been blanked to protect the privacy of the user the request is associated with. For a detailed explanation of the packet, refer to Section 5.1.7. The one in Figure 5.22 is the first packet of the entire communication between client and RADIUS server. In fact, the encapsulated EAP packet contains the response of the supplicant, which contains the name with which the user (or machine) introduces himself. As a note, for completeness, it is important to underline that in an environment where the workstations are expressly fixed (i.e., all employees are relegated to one and only one workstation), the presentation of the packet in Figure 5.22 could represent a security issue (due to the exposure of the port through attribute 87 on which the request was made). However, in a context such as IC, there is a very high level of flexibility such that productivity does not depend on the location from which one operates.

According to the type of configuration, different packets are therefore used. The 2 implementations, having both advantages and disadvantages, are implemented in certain contexts. Relay mode is used to simplify access-side processes, relegating them to the authentication server. In termination mode, on the other hand, the access device must translate the information contained in the EAPoL packets into RADIUS attributes. Also, this mode generally only supports EAP-MD5. In contrast to what is presented in Figure 5.21, Figure 5.23 graphically represents the concept on which EAPoR is based.

## EAP packet

In Figure 5.24 the generic EAP packet is shown. The various fields are described below:

- Code (1 octet): field that identifies the type of EAP packet. Each type corresponds to a certain code:
  - 1 Request;
  - 2 Response;

```

    v RADIUS Protocol
      Code: Access-Request (1)
      Packet identifier: 0x95 (149)
      Length: 400
      Authenticator: f54abc3f5844e5ca28e08a5930a6e7dd
      [The response to this request is in frame 1251]
    v Attribute Value Pairs
      > AVP: t=Framed-MTU(12) l=6 val=3038
      > AVP: t=NAS-IP-Address(4) l=6 val=
      > AVP: t=NAS-Identifier(32) l=12 val=
      > AVP: t=User-Name(1) l=14 val=
      > AVP: t=Service-Type(6) l=6 val=Framed(2)
      > AVP: t=Framed-Protocol(7) l=6 val=PPP(1)
      > AVP: t=NAS-Port(5) l=6 val=117
      > AVP: t=NAS-Port-Type(61) l=6 val=Ethernet(15)
      > AVP: t=NAS-Port-Id(87) l=6 val=3/11
      > AVP: t=Called-Station-Id(30) l=19 val=
      > AVP: t=Calling-Station-Id(31) l=19 val=
      > AVP: t=Connect-Info(77) l=38 val=CONNECT Ethernet 100Mbps Full duplex
      > AVP: t=Tunnel-Type(64) l=6 Tag=0x00 val=VLAN(13)
      > AVP: t=Tunnel-Medium-Type(65) l=6 Tag=0x00 val=IEEE-802(6)
      > AVP: t=Tunnel-Private-Group-Id(81) l=5 val=
    v AVP: t=EAP-Message(79) l=19 Last Segment[1]
      Type: 79
      Length: 19
      EAP fragment: 029400110149434e545c79796930393330
    v Extensible Authentication Protocol
      Code: Response (2)
      Id: 148
      Length: 17
      Type: Identity (1)
      Identity: _____
  
```

Figure 5.22: EAPoR packet captured in IC context.

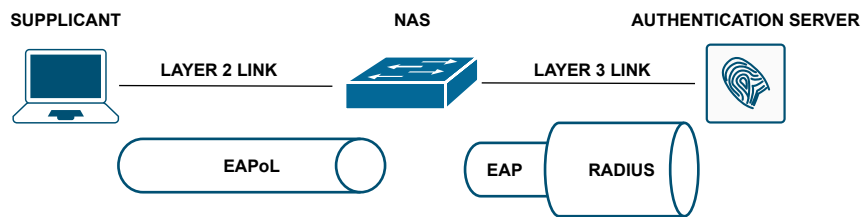


Figure 5.23: Components of 802.1X with EAPoR.

CODE	IDENTIFIER	LENGTH
DATA		

Figure 5.24: EAP packet.

- 3 Success;
- 4 Failure.

- Identifier (1 octet): field used to associate responses to respective requests;
- Length (2 octets): field indicating the total length of the EAP packet including the *Code*, *Identifier*, *Length* and *Data* fields. Excess bytes must be treated as padding by the layer 2 frame;
- Data: payload containing the information exchanged between the 2 peers. Its format depends on the value of the *Code* field used. When the *Code* field contains the values 1 or 2, this field is split into 2 that are *Type* and *Type Data*. The first is one octet long and indicates the type of request or response. The second has a variable length, and its content depends on the value of the *Type* field. When the *Code* field contains values 3 or 4, it means that the message corresponds to a *success* or a *failure* and therefore there is no *Data* field.

If the EAP packet is a Request or a Response, the *Type* field can assume different values as follows:

- 1 Identity: request or returns the username information provided by the user;
- 2 Notification: optional message that notifies the occurrence of a certain event such as password expired;
- 3 NAK: used for reply messages only, indicates a negative reply. For example, if the NAS uses an authentication method that is not supported by the end device, the latter can report lack of support using this value;
- 4 MD5-Challenge: indicates that the authentication method is MD5-Challenge;
- 5 OTP: indicates that the authentication method is One-Time Password (**OTP**);
- 6 GTC: indicates that the authentication method is Generic Token Card (**GTC**);
- 13 EAP-TLS: indicates that the authentication method is EAP-TLS;
- 17 Cisco LEAP: indicates that the authentication method is Cisco LEAP;
- 21 EAP-TTLS: indicates that the authentication method is EAP-TTLS;
- 25 EAP-PEAP: indicates that the authentication method is EAP-PEAP;
- 26 EAP-MS-CHAPv2: indicates that the authentication method is EAP-MS-CHAPv2;
- 29 PEAPv0-EAP-MS-CHAPv2: indicates that the authentication method is PEAPv0-EAP-MS-CHAPv2;

- 43 EAP-FAST: indicates that the authentication method is EAP-FAST;
- 254 Expanded Types: value customizable by vendors;
- 255 Experimental use: message used for experimental purpose.

It is important to note that each request packet contains the requested value in the *Type* field. If re-transmissions of requests are required, these must have the same value as the *Identifier* field to distinguish them from the others. Request messages must be processed in order and, in the event that duplicate requests arrive, they must not be reprocessed again but must be satisfied immediately. In the case of success or failure messages, the re-transmission function is not present. There is therefore no acknowledgment system, such that if the authenticator receives a NAK there is no second re-transmission. A first example of authentication flow is shown in Figure 5.25. For a detailed analysis of a real authentication process (captured in the context of IC) refer to the next section. During the

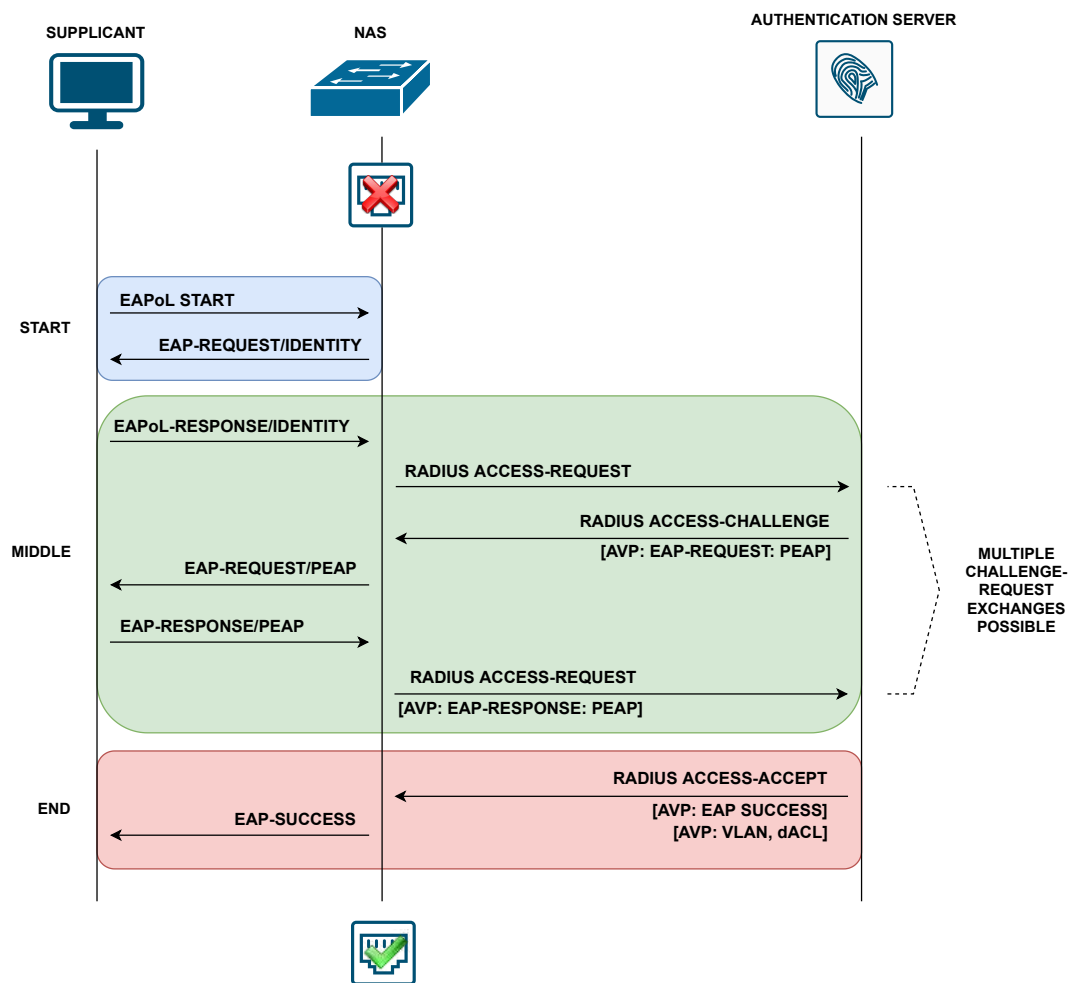


Figure 5.25: 802.1X authentication process at glance.

test phase, process initialization messages are exchanged. In the intermediate

phase, the actual information is exchanged through RADIUS challenge messages. Among these are also the creation of a tunnel (if foreseen), the negotiation of the authentication method and so on. In the final phase, on the other hand, the outcome of the authentication process is returned to the NAS with any attributes associated with authorization profiles. The NAS can now inform the supplicant of successful authentication via an EAP packet.

EAP supports a different number of authentication methods called EAP methods. The indication on which of these is used is found in the *Type* field (therefore only in the request and response messages). The method to be used is therefore negotiated between the supplicant and the authentication server. These are mainly divided into 2 sub-categories:

- Non-Tunneled EAP;
- Tunneled EAP.

What differentiates these categories is the use of an encrypted tunnel before initiating the authentication process. While with the first category the authentication process starts immediately, with the second one there is first a phase in which a TLS tunnel is established and then a second in which the actual authentication proceeds. The next section introduces some of the methods available for implementation.

### Non-Tunneled EAP

EAP methods that fall under this category do not employ the concept of inner/outer method as for tunneled. The authentication process does not therefore take place within an encrypted tunnel but rather in clear text.

### EAP-MD5

EAP-MD5 authentication method is analogous to the Challenge Handshake Authentication Protocol (**CHAP**) protocol [14]. With this type of method, a 4-step mechanism is used:

- 1 Through a request message, the authentication server sends a challenge to the client;
- 2 The client replies with a reply message containing an MD5 hash (16 bytes long) within the *Identifier* field;
- 3 The authentication server calculates the same MD5 hash locally and compares it with the one received from the client. In the event that the 2 elements match, the authentication is completed;
- 4 At random intervals steps 1 to 3 are repeated thus authenticating the client continuously;

For the state of the art, EAP-MD5 should always be supported. However it is up to the client to decide whether it supports it or not. Furthermore, this method is used in limited cases due to its exposure to possible attacks. In fact, since the traffic is not encrypted, a malicious user could capture it between client and server (e.g., using a software such as Wireshark) and carry out a dictionary attack on the hash messages thus deriving the key. Since the secret key is not dynamic, once discovered, the system is compromised. Note that this method is used within the context of IC, to authenticate IP phones. Each IP phone is equipped with a 802.1X user credentials. Once authenticated, the device is assigned a certain specific authorization profile (e.g., “*IP\_Phone\_Cisco*”). Refer to the Section 5.1.7 where the ISE authentication and profile assignment process is reported.

### EAP-TLS

This method employs a TLS tunnel for the transmission of identity credentials. EAP-TLS is an open standard, whose operation is based on X.509 certificates [15]. Unlike EAP-MD5, there is the mutual authentication functionality, which makes it one of the most widely used authentication methods. For information on tunnel creation and data encryption, refer to Section 5.1.6.

### EAP-MS-CHAPv2

This method is an updated version released by Microsoft of the previous MS-CHAPv1, which was exposed to multiple security vulnerabilities. Using this method, the user credentials are encrypted and sent, through an MS-CHAPv2 session, to the authentication server. EAP-MS-CHAPv2 is generally used in conjunction with PEAP as an inner method (as implemented in the context of IC). An example of this method (PEAP) is presented at Section 5.1.7. Figure 5.26 shows an example of EAP-MS-CHAPv2 packet [16] [17] [18]. This one is therefore composed by the following fields:

- Code (1 octet): field indicating the type of packet between request and response;
- Identifier (1 octet): field used to associate responses to related requests;
- Length (2 octets): indicates the length of the EAP packet including the *Length*, *Identifier*, *Code*, *Type*, *OpCode*, *MS-CHAPv2-ID*, *MS-Length* and *Data* fields;
- Type (1 octet): indicates the EAP method used;
- OpCode (1 octet): indicates the type of EAP-MS-CHAPv2 packet. this is broken down into:
  - 1 Challenge;
  - 2 Response;

- 3 Success;
  - 4 Failure;
  - 7 Change-Password.
- MS-CHAPv2-ID (1 octet): field with the same function as the *Identifier* field but related to the association of EAP-MS-CHAPv2 requests and responses. This field is almost always the same as that of the *Identifier*;
  - MS-Length (2 octets): field that must be set to the value of the *Length* field minus 5;
  - Data: the data format depends on the *OpCode* field.

The following is a summary of the operation of the authentication algorithm used by MS-CHAPv2:

- As a first step, a session is established between supplicant and authentication server;
- The authentication server starts the communication by sending a message which includes:
  - Session identifier (*SessionID*);
  - Pseudo-random challenge of 16 bytes (*RandA*).
- Supplicant after receiving these 2 values, replies including the following 3 values:
  - Username;
  - Pseudo-random challenge of 16 bytes (*RandB*);
  - Challenge response *CR* computed as

$$CR = ChallengeResponse(Challenge(RandA, H)) \quad (5.1)$$

where the *ChallengeResponse* function is the one used for MS-CHAPv1 and so uses variables encrypted through Data Encryption Standard (**DES**) while that of *Challenge* is the following

$$Challenge(RandA) = SHA1(RandB||RandA||Username) \quad (5.2)$$

- The authentication server computes  $CR_{check}$  as in the previous step for *CR* and send to supplicant a message with 2 specific values:
  - The result of the connection attempt (which is successful if *CR* matches  $CR_{check}$ );
  - The SHA1-hashed values of *StringC* that includes the user's password.



- Finally, the supplicant authenticates the authentication server. In case it was successful, it uses the session established in the first step, otherwise it closes it.

It is therefore possible to say that the second version of the EAP method is safer because:

- While in the first version, the key is generated starting only from the password (thus always remaining the same), in this second version the key is generated starting both from the user's password and from a random string;
- EAP-MS-CHAPv2 provides mutual authentication between nodes.

CODE	IDENTIFIER	LENGTH	
TYPE	OPCODE	MS-CHAPv2-ID	MS-LENGTH
DATA			

Figure 5.26: EAP-MS-CHAPv2 packet.

## Tunneled EAP

Most of the methods used in enterprise environments where 802.1X is implemented are tunneled methods. Before describing the authentication process, a TLS tunnel (discussed at Section 5.1.6) is therefore established between the supplicant and the authentication server. The creation of the tunnel involves an exchange of certificates (in some cases only one of the 2 nodes sends its certificate). As previously said, the tunneled methods consist of 2 phases: the first in which the encrypted tunnel is established, and the second in which authentication takes place within the tunnel just created. In a context like this one, the method used inside the tunnel is called “*inner method*” and it is simply a native EAP method. In Figure 5.27 the type of communication between supplicant and authentication server is graphically shown. It is good to remember that the tunnel created during the initial phase is encapsulated inside the EAP packet. In Figure 5.28 and Figure 5.29, the encapsulation process between supplicant and authentication server is shown. It should also be noted that the RADIUS packet is created following one of the two types of configurations described in Section 5.1.5. Some examples of EAP tunneled methods are reported below.

## EAP-PEAP

The Protected PEAP (**PEAP**) method was originally proposed by Microsoft, Cisco system and RSA Security to fix the vulnerabilities present in the original versions of EAP [19]. The first version of PEAP, i.e., PEAPv0, was released

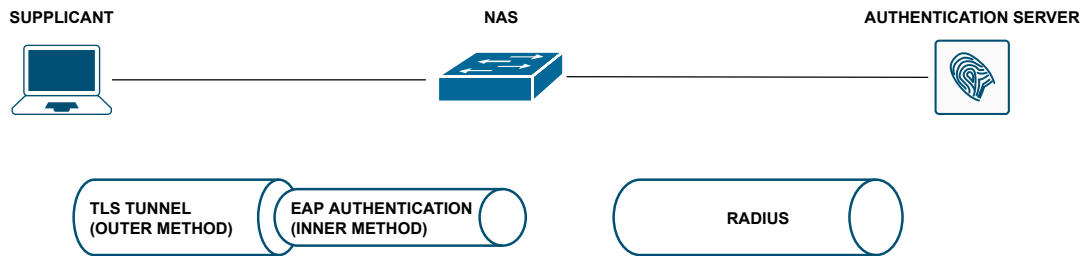


Figure 5.27: EAP tunneled method.

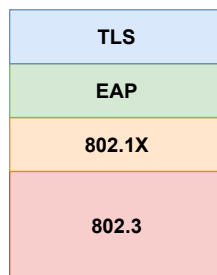


Figure 5.28: EAP tunneled method encapsulation.

```

> Frame 137: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface \Dev:
> Ethernet II, Src: ArubaaHe_ (38:21:c7 ), Dst: HewlettP_ (f4:30
> 802.1X Authentication
v Extensible Authentication Protocol
  Code: Request (1)
  Id: 43
  Length: 36
  Type: Protected EAP (EAP-PEAP) (25)
> EAP-TLS Flags: 0x00
v Transport Layer Security
  v TLSv1.2 Record Layer: Application Data Protocol: eap
    Content Type: Application Data (23)
    Version: TLS 1.2 (0x0303)
    Length: 25
    Encrypted Application Data: cd32e54705ec86daeccb5b939a3a9ec84d562d32f041ebfae8
    [Application Data Protocol: eap]

```

Figure 5.29: EAP tunneled method encapsulation in IC real scenario.

in 2002 for *Windows XP* operating systems [20]. PEAPv1 and PEAPv2 were also presented later. The main vulnerability of PEAPv0 lies in the fact that the information exchanged between peers is in clear text. It is therefore easy to understand how these can be easily sniffed. With the next releases, the identity protection functionality has been provided [21]. In the context under examination (InfoCamere) this method, together with MS-CHAPv2, has been implemented through the Group Policy Objects (**GPO**). In order to be able to use this tunneled method, it is up to the authentication server to propose the version it supports. So, in the first request sent by the server, the version indicated in the packet corresponds to the maximum supported version. This version is indicated with

the *Type* field (which in this case takes on the value 25). If the client supports the proposed version, it will respond by setting the *Type* field with the same value.

Other parameters are also negotiated during the TLS handshake. An example of these is the ciphersuite. PEAP does not support error handling. In fact, it is delegated, in the first phase to the TLS alert messages, while in the second phase to the inner authentication method. As for the inner method, this is one of those described in Section 5.1.5.

## EAP-FAST

EAP-FAST (Flexible Authentication via Secure Tunneling) is a tunneled method very similar to PEAP. There are multiple versions of EAP-FAST such as EAP-FASTv2. The version presented here (EAP-FASTv1) must support at least the following ciphersuite specifications, in order to support TLS:

- TLS\_RSA\_WITH\_RC4\_128\_SHA;
- TLS\_RSA\_WITH\_AES\_128\_CBC\_SHA;
- TLS\_DHE\_RSA\_WITH\_AES\_128\_CBC\_SHA.

The use of ciphersuites and their description are presented in the Section 5.1.6. The key differences between these 2 version is the ability to authenticate quickly and support faster wireless roaming. With EAP-FAST, Protected Access Credentials (**PAC**) are used. With this mechanism, the user is able to resume a previously started session in case he disconnects from the network. A cookie is therefore associated with it, which is shown to prove that it is already compliant. Furthermore, this function is fundamental for the concept of operation of EAP chaining. The PAC is an element made up of 3 components:

- Shared secret: this component contains the pre-shared key between the client and authentication server. This key is used to establish the TLS tunnel in the first phase;
- Opaque element: this component is provided to the client and is shown to the authentication server whenever the user wants to access the network. Variable length field, it is included in the messages intended for the tunnel establishment. This element contains the shared secret and the identity of the client;
- Other information: variable length element containing general information including the identity of the creator of the PAC (and therefore of the server that created it) and the PAC-key (shared secret) lifetime.

Unlike PEAP, only 3 inner methods are supported:

- EAP-MS-CHAPv2;
- EAP-GTC;
- EAP-TLS.

## EAP-TEAP

EAP-**TEAP** (Tunnel Extensible Authentication Protocol) was originally introduced in 2014 by Cisco System and Infineon Technologies [22]. In addition to these, a great contribution was also made by Juniper Networks and Microsoft, where the latter has integrated native support for this method. EAP-TEAP aims to end the competition between vendors using different vendor methods (Microsoft-PEAP, Cisco-FAST and so on). This method can be thought of as the merging of multiple methods, with the addition of various implementations. Some of the most notable features are listed below:

- EAP chaining function supported;
- Capability to distribute the list of trusted EAP servers' certificates to the clients;
- Certificate provisioning and renewal within the outer tunnel.

### 5.1.6 TLS tunnel

As reiterated in the previous chapters, the authentication process turns out to be a critical operation. It is therefore necessary to protect the communication with the recipient in such a way as to prevent any type of attack [23] [24] [25]. It is therefore necessary to use the TLS protocol (e.g., TLS 1.2) to guarantee the following characteristics:

- Confidentiality: ensure that the message can only be read by the correct recipient and therefore not by a MITM (achieved by using Symmetric Encryption);
- Integrity: ensure that the message is not altered in any way by third-party agents (achieved by using Message Authentication Code);
- Authentication: verify the identity of the node to which communication must be established (i.e., achieved by using Certificates/**PKI**).

The TLS protocol is positioned between the TCP and Application layers. The protocol used previously (now almost out of use) is SSL. Subsequently 4 different versions of TLS were released, 1.0, 1.1, 1.2 and the most recent 1.3. Some of the most important updates made to version 1.1 that led to the huge spread of TLS 1.2 are:

- The use of MD5/**SHA1** (Secure Hash Algorithm) in the **PRF** (Pseudo-random Function) has been replaced with SHA-256;
- Addition of new Advanced Encryption Standard (**AES**) cipher suites;
- Ability to specify hash and signature algorithms accepted by both client and server.

The following are the types of messages that are used during the tunnel establishment phase:

- Client Hello;
- Server Certificate;
- Server Key Exchange Message;
- Certificate Request;
- Server Hello Done;
- Client Certificate;
- Client Key Exchange Message;
- Certificate Verify.

The principle on which the TLS *1.2* initial handshake is based is presented below. It is important to note that before the TLS session is established, the server is in possession of a certificate (containing its public key) and obviously its private key. Furthermore, the list of messages presented must respect the order in which it is reported. If the order is not respected, the initial handshake will fail. Just before the explanation begins, it is important to point out that the messages described below can be sent aggregated into a single TLS packet. Refer to Figure 5.31 to understand how these can be aggregated together.

The first type of message is the Client Hello. However, it is important to note that this message can be preceded by a Hello Request sent by the TLS server. In case it was sent, the client would be forced to reply with a Client Hello message. The Client Hello message contains some important fields:

- Version (2 octets): this is the version with which the client wants to communicate with the server. It should also be the highest version supported by the same. Currently, most of the TLS sessions adopt version *1.2* (or *1.3*);
- Random Number (32 octets): random value generated by the client. Some protocol implementations include the timestamp encoded in the first 4 bytes;
- Session ID (8 octets): session identifier that is set to 0 in the Client Hello;
- Cipher Suites: The client specifies all cipher suites it supports to the server. The server chooses one of these proposed. The first in the list is the one preferred by the client;
- Extensions: with the inclusion of new features in the protocol, these have been tagged as extensions. For example, one of these is the server name. In the case of IC this field corresponds to IP address of the ISE PSN.

Once the server has received the Client Hello message, it generates its own message that is a Server Hello composed mainly of:

- Version (2 octets): contains the highest supported version of the TLS protocol. At this point, the version used will be the most common between client and server. In fact, adopting the most recent version guarantees a high level of security;
- Random Number (32 octets): randomly generated number (same concept described for the client);
- Session ID (8 octets): head equal to the client. The server uses it to resume pending sessions;
- Cipher Suites: contains the cipher suite selected from the list provided by the client. In the case of a resumed session, the value must be the same as the cipher suite of that session;
- Extensions: field used as for the client.

The explanation of the various fields of the Client Hello messages has been reported to make it easier to understand from which values the various keys are derived. Furthermore, previously reference was made to cipher suites. These entities are composed by 4 components:

- Exchange of keys: **RSA**, Diffie-Hellman or Elliptic-Curve Diffie-Hellman (**ECDH**);
- Authentication: RSA, Digital Signature Algorithm (**DSA**) or Elliptic-Curve Digital Signature Algorithm (**ECDSA**);
- Data encryption: AES or RC4;
- Hashing: HMAC-SHA256 or HMAC-SHA1.

An example of a cipher suite is shown in Figure 5.30. At this point both the

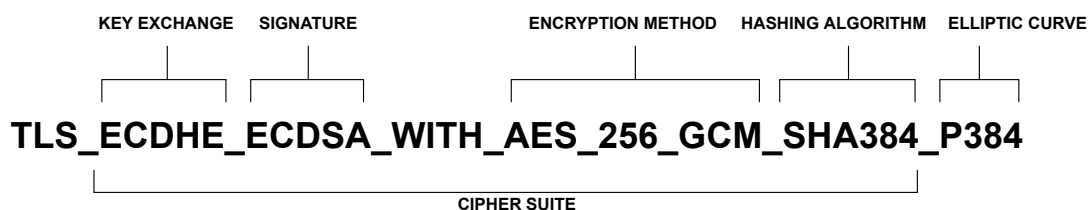


Figure 5.30: Cipher suite example.

client and the server are aware of the various information, such as the version to be used and the random numbers generated. It should be emphasized that all this information is sent in clear text and therefore can be sniffed by a malicious user. In addition to sending the Server Hello message, the server sends Certificate and

Server Hello Done messages. Certificate message is mandatory only when the cipher suite chosen requires the use of certificates. This includes the chain of certificates installed on the server. Inside this certificate is stored the public key of the server. The client therefore knows the certificate and the public key of the server. Another optional message is the Server Key Exchange message. This message type is only used when the Certificate message does not contain enough information to allow the client to exchange the *Pre Master Secret*. Another message that the server can potentially send is the Certificate Request message. As for the Certificate message, if the cipher suite also requests a certificate from the client, the latter must provide one. The last message sent by the server is the Server Hello Done. The server then communicates to the client the end of the message transmission on its part. From this point on, it is up to the client to respond to the server.

The first message that the client can potentially (and is therefore not obliged to) send is of the Client Certificate type. This message is only sent if the server has explicitly requested a certificate. If no certificates are available, the message must be sent anyway but with null content. The second message that must however be sent by the client is of the Client Key Exchange type. Through this message the *Mutual Keying Material* is established. From this *SEED* value it will be possible to generate the *Pre Master Secret*. Furthermore, this message proves that the server is the true holder of the certificate. To carry out the Key Exchange (and produce the *Pre Master Secret*) it is possible to use RSA, Diffie-Hellman or Ephemeral Diffie-Hellman. RSA is described below:

- The client generates a 48 bytes long *Pre Master Secret* starting from a random value. This is then encrypted with the server's public key and sent to the server. This *Pre Master Secret* will therefore only be visible to the server (as it holds the respective private key);
- From this *Pre Master Secret* the *Master Secret* is derived. Starting from this element, in addition with the string "*master secret*" and with the random values of previously exchanged client and server, the *Master Secret* is generated through an hashing-like algorithm;
- The newly calculated *Master Secret* is combined again with the string value "*key expansion*" and the same random values used in the previous point. The result of this operation is given as input to the hashing-like algorithm. This algorithm generates 4 keys as a result which are used to create 2 tunnels on which messages are encrypted. The 4 keys are in fact known to both 2 nodes. Of these 4, 2 are used by the client and 2 are used by the server. They represent symmetric keys, so when a message is encrypted with the *KeyA* key, only that key can decrypt it. The keys obviously conform to what is defined in the cipher suite chosen. These 4 keys are also referred to as "*Session Keys*".

One of the optional messages that the client can send at this point is the Certificate Verify message. This one is used to provide verification of the certificate.

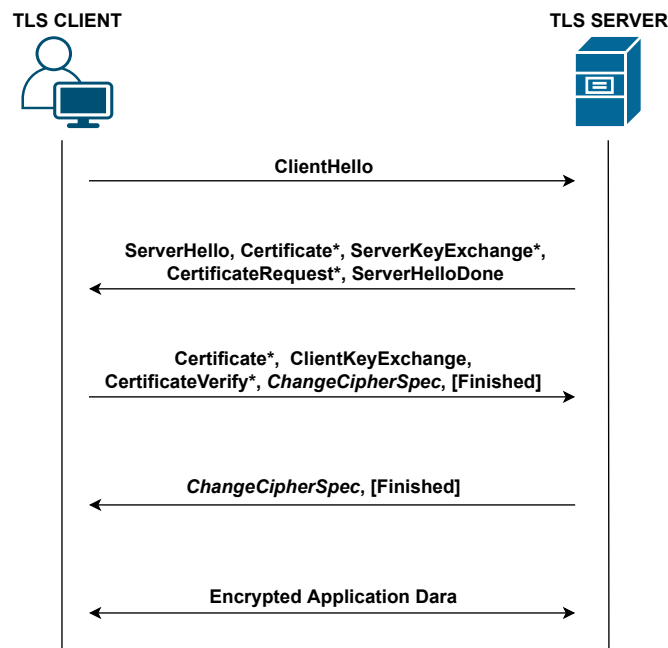


Figure 5.31: TLS 1.2 initial handshake.

Then the client sends a Change Cipher Spec message. Through this, the server is notified that the client has all the information to encrypt the traffic. All messages following the latter will be encrypted by the client. The last message the client sends to the server is the Encrypted Handshake Message. To prove that both client and server have the same *Session Keys*, this last message is required. This one is calculated starting from the information relating to the previous 5 messages through an hashing function (obtaining the *Handshake Hash*). Both nodes now get the same result. This value is given in input to a PRF together with the *Master Secret* and the string “*client\_finished*”. The result of this operation is called “*Verification Data*”. This element is then encrypted with one of the *Session Keys* (in this case by the *Client Session Keys*) and then sent to the server. The server, having all the necessary information, is able to verify that the client has the correct *Session Keys*. However, it should be noted that this type of control is carried out only towards one direction. In fact, the server has not yet verified the correctness of the *Client Session Keys*. In order to do this, the server uses the exact same mechanism just described for the client. The only difference is in the messages used during the first phase to calculate the *Handshake Hash*. In addition to those used by the client, the server also adds the Client Finished received by the client in the previous transaction. If the client is able to verify that the values are correct, the tunnel establishment mechanism can be considered to be working as expected. At this point the actual data is transmitted, encrypted with the *Session Keys*. It is important to make a small clarification: it can be seen that between the description and the capture of Wireshark there is a subtle difference between Client Finished and Encrypted Handshake messages. Both refer to the same value, however Wireshark not knowing the content because



encrypted denotes it with that special name.

No.	Time	Source	Destination	Protocol	Length	Length	Id	Info
1	0.000000	127.0.0.1	127.0.0.99	TCP	74			5555 → 443 [SYN] Seq=0 Win=65495 Len=0 MSS=65495 SACK_PERM=1 TSval=2748636071 TSecr=1035290446
2	0.000008	127.0.0.99	127.0.0.1	TCP	74			443 → 5555 [SYN, ACK] Seq=0 Ack=1 Win=65483 Len=0 MSS=65495 SACK_PERM=1 TSval=1035290446 TSecr=2748636071
3	0.000015	127.0.0.1	127.0.0.99	TCP	66			5555 → 443 [ACK] Seq=1 Ack=1 Win=65536 Len=0 TSval=2748636071 TSecr=1035290446
4	0.000293	127.0.0.1	127.0.0.99	TLSv1.2	317			Client Hello
5	0.000315	127.0.0.99	127.0.0.1	TCP	66			443 → 5555 [ACK] Seq=1 Ack=252 Win=65280 Len=0 TSval=1035290446 TSecr=2748636071
6	0.000799	127.0.0.99	127.0.0.1	TLSv1.2	2820			Server Hello, Certificate, Server Hello Done
7	0.000824	127.0.0.1	127.0.0.99	TCP	66			5555 → 443 [ACK] Seq=252 Ack=2755 Win=63616 Len=0 TSval=2748636071 TSecr=1035290446
8	0.001717	127.0.0.1	127.0.0.99	TLSv1.2	424			Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
9	0.001738	127.0.0.99	127.0.0.1	TCP	66			443 → 5555 [ACK] Seq=2755 Ack=610 Win=65280 Len=0 TSval=1035290447 TSecr=2748636071
10	0.003411	127.0.0.99	127.0.0.1	TLSv1.2	157			Change Cipher Spec, Encrypted Handshake Message
11	0.003433	127.0.0.1	127.0.0.99	TCP	66			5555 → 443 [ACK] Seq=610 Ack=2846 Win=65536 Len=0 TSval=2748636074 TSecr=1035290446
12	16.011200	127.0.0.1	127.0.0.99	TLSv1.2	151			Application Data
13	16.011234	127.0.0.99	127.0.0.1	TCP	66			443 → 5555 [ACK] Seq=2846 Ack=695 Win=65536 Len=0 TSval=1035306457 TSecr=2748652089
14	16.018751	127.0.0.99	127.0.0.1	TLSv1.2	951			Application Data
15	16.018779	127.0.0.1	127.0.0.99	TCP	66			5555 → 443 [ACK] Seq=695 Ack=3731 Win=64768 Len=0 TSval=2748652089 TSecr=1035306464
16	16.018806	127.0.0.99	127.0.0.1	TCP	66			443 → 5555 [FIN, ACK] Seq=3731 Ack=695 Win=65536 Len=0 TSval=1035306464 TSecr=2748652089
17	16.018931	127.0.0.1	127.0.0.99	TLSv1.2	135			Encrypted Alert
18	16.018957	127.0.0.99	127.0.0.1	TCP	66			443 → 5555 [ACK] Seq=3732 Ack=764 Win=65536 Len=0 TSval=1035306465 TSecr=2748652089
19	16.018990	127.0.0.1	127.0.0.99	TCP	66			5555 → 443 [FIN, ACK] Seq=764 Ack=3732 Win=65536 Len=0 TSval=2748652090 TSecr=1035306465
20	16.019013	127.0.0.99	127.0.0.1	TCP	66			443 → 5555 [ACK] Seq=3732 Ack=765 Win=65536 Len=0 TSval=1035306465 TSecr=2748652089

Figure 5.32: TCP three-way handshake and TLS 1.2 initial handshake.

No.	Time	Source	Destination	Protocol	Length	Length	Id	Info
4	0.000293	127.0.0.1	127.0.0.99	TLSv1.2	317			Client Hello
6	0.000799	127.0.0.99	127.0.0.1	TLSv1.2	2820			Server Hello, Certificate, Server Hello Done
8	0.001717	127.0.0.1	127.0.0.99	TLSv1.2	424			Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
10	0.003411	127.0.0.99	127.0.0.1	TLSv1.2	157			Change Cipher Spec, Encrypted Handshake Message
12	16.011200	127.0.0.1	127.0.0.99	TLSv1.2	151			Application Data
14	16.018751	127.0.0.99	127.0.0.1	TLSv1.2	951			Application Data
17	16.018931	127.0.0.1	127.0.0.99	TLSv1.2	135			Encrypted Alert

Figure 5.33: Packet capture of TLS 1.2 initial handshake.

### 5.1.7 Authentication flow

The authentication process is closely related to the type of deployment that is used [26]. The considerations made below for the authentication flow are related to the IC context (and therefore using Cisco ISE as NAC server). Before starting to illustrate the process, it is necessary to specify how the flow is strictly related to the type of deployment. Not to be confused with those described in Section 4.1.2, these are divided into 3 types:

- Monitor mode;
- Low-impact mode;
- Closed mode.

Therefore, on the basis of which of the three deployments is implemented, there will be different traffic flows. These are described in the following sections. In the context of IC, the low-impact mode was adopted. Being an ever-changing environment, the policies are subject to revisions every certain amount of time. During these reviews, changes are usually applied to deflect any issues that occur with some devices/users. Despite this, there are policies that allow the implementation of closed mode. In fact, some of them allow, for example with 802.1X, to deliver an Access-Reject message in case of incorrect credentials. This isolates the user from the campus network. It is important to underline that, based on

the supplicant's configuration, there can be two types of authentication. Both the user and the machine can authenticate their credentials (first the machine is authenticated and then the user). In an enterprise environment, such as that of IC, the configurations of the individual devices are managed by a central node through the GPOs shown in Figure 5.34. In IC, as shown in the Section 5.4.3 section, it has been set up that both entities have to authenticate. While the user authenticates himself with his own credentials, the machines will use as username a value derived from the asset ID assigned to them, and as password a random choice. During the analyses carried out regarding the authentication processes, it has been possible to notice how some devices failed authentication. Since the device periodically updates the password of the company device, it is possible that a misalignment with the domain controller (central node which is entrusted with the centralized management of the GPOs) may occur. As can be seen from the error message in Figure 5.35, the machine ID within the AD has been found, but the password provided is not correct.

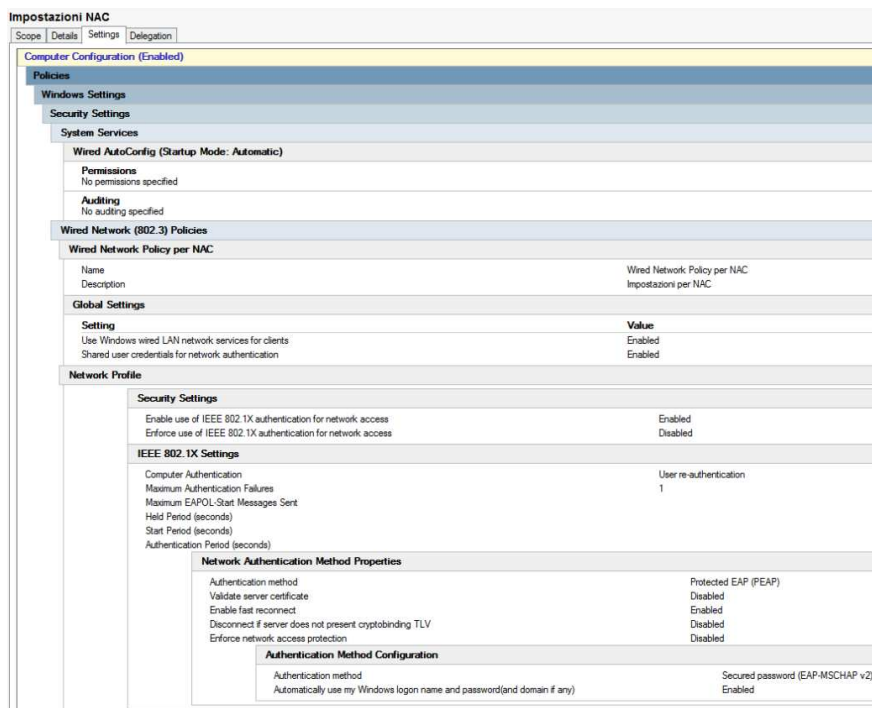


Figure 5.34: IC GPO template.

The entire authentication process is shown in Figure 5.38. Note how the filter used to filter the packets captured by Wireshark corresponds to the string “EAPoL || RADIUS”. This takes advantage of what is reported in Figure 5.21. Before starting to illustrate the process captured in the IC test environment, it is important to make a premise: this process is not the complete one. In fact, as can be seen from Figure 5.59, the authentication method set corresponds to the “User or Machine” value. The authentication process is therefore composed by 2 steps:

Authentication Details	
Source Timestamp	2022-05-25 06:59:27.415
Received Timestamp	2022-05-25 06:59:27.415
Policy Server	
Event	5440 Endpoint abandoned EAP session and started new
Failure Reason	24485 Machine authentication against Active Directory has failed because of wrong password
Resolution	Check if the machine is present in the Active Directory domain and if it is spelled correctly. Also check whether machine authentication is configured properly on the supplicant. Please verify that LSA registry key is set correctly on supplicant.
Root cause	Machine authentication against Active Directory has failed because of wrong password.
Username	
Endpoint Id	

Figure 5.35: Machine authentication report example.

- First step: during the boot phase, the authentication of the machine takes place whose username is derived from the asset ID, while the password is managed centrally by the domain controller;
- Second step: during the user's login, authentication takes place with the respective credentials.

However, there are also two special cases, such that only user authentication occurs. These occur when the device is connected to the wired network after it has already been boot on (and in the second case after the user has also logged in). In these cases, the most specific credentials to be presented to the NAC server are those relating to the user and not to the machine. This situation can therefore occur when an InfoCamere employee does not turn off the device within 2 working days, but leaves it in standby mode. The next day, connecting it to the network, it does not carry out the boot phase, thus not presenting the machine credentials. For clarity, the 2 communication flows are shown in different colors:

- White: communication between end device and NAD;
- Blu: communication between NAD and authentication server (ISE);

The message from which the authentication process begins comes from a device (in this case an HP PC). In general, this situation does not always occur. In fact, there are 2 possible scenarios:

- In the first one, corresponding to the exam case, the first message comes from the supplicant (EAPoL-Start);
- In the second one, the first message comes from the NAD (Request/Identity message);

Performing a port bounce operation (i.e., turning it off and on again), will trigger the transmission of the first packet (Request/Identity type EAPoL packet). By activating instead the 802.1X service within the device, the first transmission is performed by it. Let's consider a scenario in which the device is connected to the network (and therefore to the interface of the switch). As previously discussed, the messages exchanged between the device and the NAD do not occur through the IP protocol, but through a layer 2 communication. The addresses used are in fact the MAC addresses. Specifically, the destination MAC addresses correspond to a particular addressing group (not to be confused with multicast addresses) defined by the IEEE 802 group. This addressing group identifies PAE entities. The NAD then proceeds to respond, requesting identity from the device. This identity is associated with the supplicant installed within the device (which can be related to the machine or to the user). The response of the device may vary according to the configuration of the supplicant (visible in Figure 5.61 in the last item of the tab). If this option is set, and the associated field is filled in, the first reply packet has the specified value as its identity. If, on the other hand, it is not set, the value corresponds to the real identity of the supplicant. In the case of IC, this identity is sent by the device in clear text. This means that any person, placing himself between the 2 nodes, would be able to know who is trying to authenticate himself through a traffic analysis software (e.g., Wireshark). This configuration could be perplexing as the name of the EAP method adopted is defined "*Protected*". However, the reason for this choice goes deeper. In fact, in scenarios where roaming is supported, it may be necessary to locate the correct authentication server before the process begins.

Once the NAD has received the identity, it proceeds to create the Access-Request RADIUS packet to be sent to the RADIUS server (i.e., the authentication server). Since the device operates in relay mode, within it it is possible to find, through attribute 79, the EAP packet previously received. In addition to this, there are multiple attributes that can be configured within the NAD. Through the CLI of the device, it is possible to specify which attributes must be present and in which types of packets. For example, the basic attributes that must necessarily be present are 6 (to know what type of authentication is used), 8 (to indicate the IP address of the requesting device) and 25.

The RADIUS server, after receiving the request, responds with an Access-Challenge message. Within this message there is an EAP packet, through which the server proposes to the supplicant to use PEAP (type 25) as the authentication method. The NAD decapsulates this packet and sends it to the supplicant. The latter can decide whether to use the proposed method or not. In fact it may happen that the method proposed by the server is not supported by the client. As a result, the supplicant would respond with a NAK and in turn propose its desired method. In the case study, the proposed method (PEAP) is accepted by the supplicant. In fact, referring to Figure 5.34 it is possible to see how in the GPO, the configured method is PEAP. Furthermore, in ISE, it is possible to configure which authentication method to propose to the supplicant at the first attempt.

Once agreed on the method, the supplicant inserts a Client Hello message in the same packet to start the establishment of the TLS tunnel. The most relevant parameters relating to this message are:

- Random value: used later to obtain the final keys;
- Cipher suite: the list of all supported cipher suites;
- Version: highest TLS version (when establishing a TLS tunnel, the version used becomes of relative importance both for the specific implementation aspects and for the aspects related to security).

For ease of notation, when it is reported that the supplicant sends a packet to the authentication server, it actually first sends it to the NAD and then the latter delivers it to its destination. The task of the NAD in some cases can be interpreted as an intermediary. In this phase, from Figure 5.38 it is possible to see how a series of exchanges of information take place. In fact, packets no. 17980, 17984, 17988, 17993, 17999, and 18003 represent multiple fragments of a single EAP packet encapsulated in a RADIUS packet sent by ISE. Inside there are the various TLS handshake protocols (Server Hello, Certificate, Server Key Exchange and Server Hello Done). In this case, the TLS version supported by ISE is the same as that of the supplicant. However, within the ISE GUI it is possible to configure the versions that can be supported by the ISE itself. In this way, if a supplicant requests to establish a TLS tunnel using version *1.0*, he can be rejected. At this point, the supplicant knows all the information needed to encrypt a message. In fact, it should be noted that in packet no. 18005, together with the message n. 18009, the creation of the tunnel has been completed. These 2 messages correspond to the process described in Section 5.1.6 in which the two nodes verify the correct creation of the final keys. Furthermore, in accordance with what is defined by the RFC, the supplicant sends an empty EAP message, with the *Type* field set to PEAP.

With the end of the first phase, i.e., the establishment of the TLS tunnel, the second one begins. As can be seen, from now on the information contained within the packet is fully encrypted. The first 2 messages, as occurred in the first phase, are a Request/Identity and a Response. This information is then sent to ISE. With packet no. 18031, the authentication server, after receiving the identity proposes an authentication method to the supplicant. This method is always configurable from the ISE GUI. In the case study, having only been enabled PEAP with MS-CHAPv2, the latter is used as inner method. Before continuing with the explanation, Figure 5.36 shows a generic EAP-MS-CHAPv2 packet.

It should be emphasized that in this specific case the first 3 fields (*Code*, *Identifier* and *Length*) are not present inside the packets. This is because the PEAP implementation means that the EAP header is not used for packets whose *Type* field is not 33 (EAP extension packet). The first available field is therefore the *Type* one. The first packet sent by the authentication server during this second phase contains for example the following values:

CODE	IDENTIFIER	LENGTH	
TYPE	OpCODE	MS-CHAPv2-ID	MS-LENGTH
DATA			

Figure 5.36: EAP-MS-CHAPv2 packet.

- Code (1 octet): 1 (Request);
- Type (1 octet): 26 (EAP-MS-CHAPv2);
- OpCode (1 octet): 1 (Challenge);
- MS-CHAPv2-ID (1 octet): ID used to matching the request and response messages.

The *Data* field, based on the *OpCode* field value, takes different forms. For the request, it contains the following values:

- Challenge (16 octets): unique random value;
- Name (1 or more octets): identifies the sender of the message.

Upon receiving the message delivered to him by the NAD, the supplicant responds with a message with the following characteristics:

- Code (1 octet): 2 (Response);
- Type (1 octet): 26 (EAP-MS-CHAPv2);
- OpCode (1 octet): 2 (Response);
- Response (49 octets):
  - Peer-Challenge (16 octets);
  - Reserved, must be zero (8 octets);
  - NT-Response (24 octets);
  - Flags (1 octet).
- Name (1 octet): identifies the name of the supplicant.

The authentication server, therefore, extrapolates the value contained in the *Response* field and compares it with the value that it expects. If they match, ISE sends a Success Request packet (i.e., *OpCode* field set to 3 and *Code* field set to 1). This packet contains the *Message* field inside the *Data* field. The content of the latter is derived through functions from some values present in the previous

messages. In the event that the supplicant should obtain a positive result after analyzing the packet received by ISE, he will in turn send a Success Response packet (i.e., *OpCode* field set to 3 and *Code* field set to 2) with no values inside it. As for the last 2 EAP-MS-CHAPv2 messages, these are used as ACK messages to make sure that authentication has taken place. Note how, inside the Access-Accept RADIUS packet sent by ISE, there are 2 keys within 2 Microsoft VSAs (Figure 5.37). The last message that concludes the authentication process is the Success message, sent by the NAD, contained in the EAP packet (i.e., *Code* field set to 3).

```

  v AVP: t=Vendor-Specific(26) l=58 vnd=Microsoft(311)
    Type: 26
    Length: 58
    Vendor ID: Microsoft (311)
  > VSA: t=MS-MPPE-Send-Key(16) l=52 val=aceca66f4feaace2e61084017377478ea57167ad83c6c240c4c7482fc8b7b70937f4279...
  v AVP: t=Vendor-Specific(26) l=58 vnd=Microsoft(311)
    Type: 26
    Length: 58
    Vendor ID: Microsoft (311)
  > VSA: t=MS-MPPE-Recv-Key(17) l=52 val=b4ab514c19d533f1e471eb8f7f8154e3cda9298cea80d2b46dff386a582bde0591b1f4c8...

```

Figure 5.37: EAP-MS-CHAPv2 session keys.

For information, it is specified that, in order to analyze the authentication attempts, multiple disconnections were made using different methods. These disconnections can be initiated from the end device (via EAPoL-Logoff message), from the NAD (eliminating the session currently operating through the CLI) or directly from the authentication server (ISE) interface. Considering the scenario of the second type, Figure 5.39 shows a capture (carried out in the IC test environment) of the content of a RADIUS Accounting-Request packet coming from the NAD and directed to ISE. As can be seen, attribute 40 (Acct-Status-Type) contains the value *Stop*. Through this attribute, it is specified whether the accounting packet denotes the beginning of a service relating to the user, or its end. In this case, when the operator terminate the session, the connection between the device and the network is interrupted. As can be seen from an extract of the list of values that this field can assume, it is also possible to specify the stop of the accounting service only:

- 1 Start;
- 2 Stop;
- 3 Interim-Update;
- 7 Accounting-On;
- 8 Accounting-Off.

With the message indicating session termination, other special attributes are included (only available for this type of communication). Attributes 42-43 and 47-48 respectively specify the total number of octets and packets that are transited in input/output through the connected interface. These values could be used in the event that an in-depth analysis is carried out for a troubleshooting operation.



17934	HewlettP_	Nearest-non-TPMR-b...	EAPOL	60	Start
17939	Cisco_	Nearest-non-TPMR-b...	EAP	60	Request, Identity
17954	HewlettP_	Nearest-non-TPMR-b...	EAP	60	Response, Identity
17955	10.	10.	RADIUS	363	Access-Request id=113
17956	10.	10.	RADIUS	168	Access-Challenge id=113
17957	Cisco_	Nearest-non-TPMR-b...	EAP	60	Request, Protected EAP (EAP-PEAP)
17964	HewlettP_	Nearest-non-TPMR-b...	TLsv1_	190	Client Hello
17965	10.	10.	RADIUS	573	Access-Request id=114
17978	10.	10.	RADIUS	1180	Access-Challenge id=114
17980	Cisco_	Nearest-non-TPMR-b...	EAP	1030	Request, Protected EAP (EAP-PEAP)
17981	HewlettP_	Nearest-non-TPMR-b...	EAP	60	Response, Protected EAP (EAP-PEAP)
17982	10.	10.	RADIUS	407	Access-Request id=115
17983	10.	10.	RADIUS	1176	Access-Challenge id=115
17984	Cisco_	Nearest-non-TPMR-b...	EAP	1026	Request, Protected EAP (EAP-PEAP)
17985	HewlettP_	Nearest-non-TPMR-b...	EAP	60	Response, Protected EAP (EAP-PEAP)
17986	10.	10.	RADIUS	407	Access-Request id=116
17987	10.	10.	RADIUS	1176	Access-Challenge id=116
17988	Cisco_	Nearest-non-TPMR-b...	EAP	1026	Request, Protected EAP (EAP-PEAP)
17990	HewlettP_	Nearest-non-TPMR-b...	EAP	60	Response, Protected EAP (EAP-PEAP)
17991	10.	10.	RADIUS	407	Access-Request id=117
17992	10.	10.	RADIUS	1176	Access-Challenge id=117
17993	Cisco_	Nearest-non-TPMR-b...	EAP	1026	Request, Protected EAP (EAP-PEAP)
17996	HewlettP_	Nearest-non-TPMR-b...	EAP	60	Response, Protected EAP (EAP-PEAP)
17997	10.	10.	RADIUS	407	Access-Request id=118
17998	10.	10.	RADIUS	1176	Access-Challenge id=118
17999	Cisco_	Nearest-non-TPMR-b...	EAP	1026	Request, Protected EAP (EAP-PEAP)
18000	HewlettP_	Nearest-non-TPMR-b...	EAP	60	Response, Protected EAP (EAP-PEAP)
18001	10.	10.	RADIUS	407	Access-Request id=119
18002	10.	10.	RADIUS	930	Access-Challenge id=119
18003	Cisco_	Nearest-non-TPMR-b...	TLsv1_	780	Server Hello, Certificate, Server Key Exchange, Server Hello Done
18005	HewlettP_	Nearest-non-TPMR-b...	TLsv1_	154	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
18006	10.	10.	RADIUS	537	Access-Request id=120
18008	10.	10.	RADIUS	219	Access-Challenge id=120
18009	Cisco_	Nearest-non-TPMR-b...	TLsv1_	75	Change Cipher Spec, Encrypted Handshake Message
18023	HewlettP_	Nearest-non-TPMR-b...	EAP	60	Response, Protected EAP (EAP-PEAP)
18024	10.	10.	RADIUS	407	Access-Request id=121
18025	10.	10.	RADIUS	198	Access-Challenge id=121
18026	Cisco_	Nearest-non-TPMR-b...	TLsv1_	60	Application Data
18027	HewlettP_	Nearest-non-TPMR-b...	TLsv1_	91	Application Data
18030	10.	10.	RADIUS	474	Access-Request id=122
18031	10.	10.	RADIUS	230	Access-Challenge id=122
18032	Cisco_	Nearest-non-TPMR-b...	TLsv1_	86	Application Data
18040	HewlettP_	Nearest-non-TPMR-b...	TLsv1_	145	Application Data
18042	10.	10.	RADIUS	528	Access-Request id=123
18052	10.	10.	RADIUS	244	Access-Challenge id=123
18053	Cisco_	Nearest-non-TPMR-b...	TLsv1_	100	Application Data
18063	HewlettP_	Nearest-non-TPMR-b...	TLsv1_	60	Application Data
18064	10.	10.	RADIUS	438	Access-Request id=124
18066	10.	10.	RADIUS	208	Access-Challenge id=124
18067	Cisco_	Nearest-non-TPMR-b...	TLsv1_	64	Application Data
18068	HewlettP_	Nearest-non-TPMR-b...	TLsv1_	64	Application Data
18069	10.	10.	RADIUS	447	Access-Request id=125
18097	10.	10.	RADIUS	365	Access-Accept id=125
18099	Cisco_	Nearest-non-TPMR-b...	EAP	60	Success

Figure 5.38: Authentication process in IC scenario.

With attribute 46 (*Acct-Session-Time*), on the other hand, the time for which the user has used the service (network access) is indicated. It is important to note that this time does not take into account any intermediate re-authentication in case they were successful. The termination of the session, as specified above, can take place in many ways. The latter are specified with attribute 49 (*Acct-Terminate-Cause*). From Figure 5.39 can be seen how, having used the command “*clear authentication sessions interface GigabitEthernet 1/0/46*”, the attribute assumes the value *Admin-Reset*. Here are some of the values related to this attribute:

- 1 User-Request;
- 5 Session-Timeout;
- 6 Admin-Reset;
- 18 Host-Request.



The last attribute to pay attention to is 45 (Acct-Authentic). Through it it is possible to specify from which server the user/machine has been authenticated. The attribute can take the following values:

- 1 RADIUS: if the user is authenticated to a RADIUS server;
- 2 Local: if the user is authenticated to a local authentication server (within the NAD);
- 3 Remote: if the user is authenticated through any other remote server;
- 4 Diameter: if the user is authenticated to a Diameter server.

```

▼ RADIUS Protocol
  Code: Accounting-Request (4)
  Packet identifier: 0xcf (207)
  Length: 329
  Authenticator: 7e33392efaba3bbfe02b8242ab48554c
  [The response to this request is in frame 954923]
  ▼ Attribute Value Pairs
    > AVP: t=Framed-IP-Address(8) l=6 val=10.
    > AVP: t=User-Name(1) l=14 val=IC
    > AVP: t=Vendor-Specific(26) l=49 vnd=ciscoSystems(9)
    > AVP: t=Vendor-Specific(26) l=20 vnd=ciscoSystems(9)
    > AVP: t=Called-Station-Id(30) l=19 val=00-2C-C8-
    > AVP: t=Calling-Station-Id(31) l=19 val=F4-30-89-
    > AVP: t=NAS-IP-Address(4) l=6 val=10.
    > AVP: t=NAS-Identifier(32) l=13 val=
    > AVP: t=NAS-Port-Id(87) l=23 val=GigabitEthernet1/0/
    > AVP: t=NAS-Port-Type(61) l=6 val=Ethernet(15)
    > AVP: t=NAS-Port(5) l=6 val=50146
    > AVP: t=Acct-Session-Id(44) l=10 val=0000007b
    > AVP: t=Acct-Authentic(45) l=6 val=Remote(3)
    > AVP: t=Acct-Terminate-Cause(49) l=6 val=Admin-Reset(6)
    > AVP: t=Class(25) l=58 val=434143533a30413031323039363030303030434636363541353444413a766163696973...
    > AVP: t=Acct-Status-Type(40) l=6 val=Stop(2)
    > AVP: t=Event-Timestamp(55) l=6 val=Jun 15, 2022 10:40:51.000000000 ora legale Europa occidentale
    > AVP: t=Acct-Session-Time(46) l=6 val=2846
    > AVP: t=Acct-Input-Octets(42) l=6 val=48714802
    > AVP: t=Acct-Output-Octets(43) l=6 val=53276
    > AVP: t=Acct-Input-Packets(47) l=6 val=86018
    > AVP: t=Acct-Output-Packets(48) l=6 val=778
    > AVP: t=Acct-Delay-Time(41) l=6 val=0
  
```

Figure 5.39: RADIUS termination session message.

## IP phone authentication

While the previous section illustrated the authentication process of a PC using the PEAP method, this section briefly illustrates the process of connecting a generic IP phone within the test environment. This therefore does not reflect the real authentication process present in the production environment. However, since the IP phones are of considerable importance, an example of authentication flow is shown for completeness. The process is shown in Figure 5.43. As mentioned earlier, the IP phones in production are equipped with an 802.1X supplicant. However, the credentials are not pre-installed on the devices but are downloaded from a company-internal TFTP server. However, the TFTP server

information is also not pre-installed. So how is this information passed to the IP phone? All this happens thanks to DHCP process. The DHCP protocol is used by the devices to obtain an address in a dynamic way, i.e., without a manual intervention. The addressing process takes place mainly through 4 types of messages (generically known as “*DORA*” messages):

- DHCP DISCOVER;
- DHCP OFFER;
- DHCP REQUEST;
- DHCP ACK.

After receiving the broadcast request from the IP phone, the DHCP server offers an IP address. Now it is needed to specify the DHCP packet format. In fact, in addition to the information necessary for the automatic addressing service to be successful, there are also options that can be used according to the needs of the infrastructure. In this case, the interesting options for provisioning are 6 (DNS) and 66 (TFTP Server Name). In some cases, through option 67 (Bootfile name), the name of the configuration file to be downloaded can also be specified. In the IC architecture, the 67 option is not used, due to the fact that the file configuration name contains the MAC address of the IP device (and so each IP phone has its own specific provisioning file). Doing so, each phone can be associated to a specific intern telephone number.

Once the IP address is obtained, the IP phone automatically connects to the server (in the case study the IC switchboard of the Padova office) to download the configuration file. However, it is important to point out that with option 66 it is possible to specify a server which can be both TFTP and HTTP. In fact, in the case study, an HTTP request is sent to the server. In the case described, due to telephone switchboard issues, the 802.1X supplicant has been disabled from the configuration within the server. So, as can be seen from Figure 5.43, authentication takes place via MAB. The EAPoL packets sent by the NAD (Cisco switch Catalyst 3650) to the IP phone do not follow any response. The device, not responding to the requests of the switch, begins to send DISCOVER packets to obtain a valid IP address. Note how in the capture, carried out with Wireshark and shown in the figure, the DHCP packets are shown duplicated. This is due to the fact that, to monitor the traffic, a SPAN port has been configured with two interfaces as a source:

- The access interface towards the IP phone, and therefore also towards the PC;
- The trunk interface that connects the test appliance with the production switch (i.e., uplink interface).

Due to this, the single packet is detected 2 times. The first when transmitted by the device, the packet arrives at the access interface of the switch. The second is

when, after being switched, it is sent to the DHCP server via the trunk interface. Remember that this type of interface allows the transmission of multiple VLANs on the same transmission medium. It is widely used in uplink connections. Figure 5.40 shows the topology with which the captures were made. As can be seen, the connection in monitor mode does not provide for the generation of traffic. All traffic generated and received on the interface is therefore discarded.

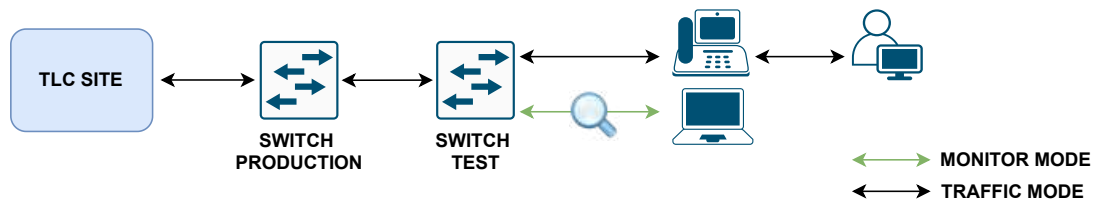


Figure 5.40: Traffic monitoring architecture in test environment.

Returning to the process, following the failure of 802.1X authentication, the switch uses the MAB (identical procedure explained above). Once authenticated, an authorization profile has been associated with the device (e.g., it is assigned to the correct Voice VLAN). The switch then sends a RADIUS Accounting-Request message with which it initiates the session. From now on, the DHCP packets generated by the device will be handled and transmitted correctly to the DHCP server. In fact, during the first access, the access VLAN did not allow packets to be transmitted to the uplink appliances. The latter will provide information such as domain information, lease time, DNS and TFTP servers. Having all the information available, the IP phone tries to contact the server at the obtained IP.

The IP phone, which in the example case is a *Yealink* IP phone, first establishes a TCP connection through the three-way handshake. Next, the actual configuration file is downloaded. Figure 5.41 shows the HTTP response from the server. As can be seen, the configuration in plain text is downloaded. The information in the figure was obtained using the “*follow TCP stream*” function present in Wireshark. As said before, the authentication process carried out by the *Yealink* IP phone is based on MAB. This is due to the fact that at the beginning of the testing phase, a monitor based approach should be used. In this way, any device (in this case IP phone) that connects to the NAD, can access the network. Obviously this cannot in any case be considered as a final deployment. Furthermore, the telephone switchboard implemented, used a provisioning file without 802.1X credentials. Due to this, the device, not being equipped with an 802.1X supplicant, authenticated itself via MAB. Subsequently, for the test phase, a *Polycom* brand IP phone has also been used. This device, unlike the *Yealink*, presented difficulties during the provisioning phase. In fact, once the credentials were enabled in the provisioning file for the *Yealink* IP phone, the latter was able to authenticate via 802.1X correctly. However, this behavior has never occurred with *Polycom* IP phones. After an analysis phase, together with a VoIP expert, it was possible to find what the problem was. The main problem was related to specific portions of 802.1X configuration within the provisioning

```

HTTP/1.1 200 OK
Date: Mon, 04 Jul 2022 07:47:32 GMT
Server: Apache/2.4.6 (CentOS)
X-Powered-By: PHP/5.4.16
Cache-Control: no-cache, must-revalidate
Expires: Mon, 26 Jul 1997 05:00:00 GMT
Vary: Accept-Encoding
Transfer-Encoding: chunked
Content-Type: text/plain; charset=UTF-8

4475
#!version:1.0.0.1

##File header "#!version:1.0.0.1" can not be edited or deleted.##
###Generated on: 22/07/2022, 09:47:32

###
# File di configurazione device Yealink stil
###

#####
##                               Account1 Settings                               ##
#####

```

Figure 5.41: IP phone configuration download sample.

file. In fact, some of the parameters were set incorrectly, causing the IP phone to fail to authenticate. In the following list it is possible to see the configuration items relating to 802.1X.

```

device.net.dot1x.password.set="1"
device.net.dot1x.enabled.set="1"
device.net.dot1x.enabled="1"
device.net.dot1x.method.set="1"
device.net.dot1x.method="7"
device.net.dot1x.identity.set="1"
device.net.dot1x.identity="*****"
device.net.dot1x.password="*****"
device.net.dot1x.anonid.set="0"
device.net.dot1x.anonid=""
device.net.dot1x.eapFastInBandProv.set="0"
device.net.dot1x.eapFastInBandProv="0"

```

In particular, some of the parameters that has been modified are:

- `eapFastInBandProv.set` and `eapFastInBandProv`: this parameters have to be used only for EAP-FAST authentication method. Binary value equal to 0 stands for “*disabled*”;
- `method`: to set the authentication method as EAP-MD5 it is necessary to enter the value 7 within the method attribute. Previously the method was set via a string value.

Once the provisioning file was corrected, it was possible to verify that authen-

tication via EAP-MD5 was successful. The EAP-MD5 authentication process is shown in Figure 5.42. The EAP-MD5 authentication process, described at Section 5.1.5, employs the following method:

$$ChallengeResponse = MD5(Id||Password||Challenge) \quad (5.3)$$

With the previous configuration downloaded from the switchboard (i.e., the wrong one), the password was downloaded correctly. However, because the settings were incorrect, *Polycom* IP phone misinterpreted the information, resulting in the wrong MD5 hash (i.e., *ChallengeResponse*). This problem only occurs with *Polycom* IP phones. The fact of having remedied this configuration, allows the internal group of IC that deals with the management of VoIP telephony, to apply these changes in the production environment.

1756210	19572.705943	Cisco_	YealinkX_	EAP	60 Request, Identity
1756211	19572.707308	YealinkX_	Nearest-non-TPMR-br...	EAP	60 Response, Identity
1756212	19572.709174	10.	10.	RADIUS	303 Access-Request id=58
1756213	19572.713116	10.	10.	RADIUS	170 Access-Challenge id=58
1756214	19572.714510	Cisco_	YealinkX_	EAP	60 Request, TLS EAP (EAP-TLS)
1756215	19572.715303	YealinkX_	Nearest-non-TPMR-br...	EAP	60 Response, Legacy Nak (Response Only)
1756216	19572.716736	10.	10.	RADIUS	379 Access-Request id=59
1756218	19572.718241	10.	10.	RADIUS	197 Access-Challenge id=59
1756220	19572.720157	Cisco_	YealinkX_	EAP	60 Request, MD5-Challenge EAP (EAP-MD5-CHALLENGE)
1756221	19572.721299	YealinkX_	Nearest-non-TPMR-br...	EAP	60 Response, MD5-Challenge EAP (EAP-MD5-CHALLENGE)
1756222	19572.722934	10.	10.	RADIUS	395 Access-Request id=60
1756223	19572.768828	10.	10.	RADIUS	321 Access-Accept id=60
1756236	19572.774182	10.	10.	RADIUS	194 Access-Request id=61
1756241	19572.777250	10.	10.	RADIUS	248 Access-Accept id=61
1756245	19572.800118	Cisco_	YealinkX_	EAP	60 Success

Figure 5.42: EAP-MD5 authentication flow for IP phone.

7917...	56930.862343	Cisco_	YealinkX_	EAP	60 Request, Identity
7917...	56931.867759	Cisco_	YealinkX_	EAP	60 Request, Identity
7918...	56932.856333	Cisco_	YealinkX_	EAP	60 Request, Identity
7918...	56933.172268	0.0.0.0	255.255.255.255	DHCP	590 DHCP Discover - Transaction ID 0xcc62428
7919...	56933.857197	Cisco_	YealinkX_	EAP	60 Request, Identity
7919...	56934.859457	Cisco_	YealinkX_	EAP	60 Failure
7919...	56934.860921	10.	10.	RADIUS	314 Access-Request id=130
7919...	56934.884328	10.	10.	RADIUS	324 Access-Accept id=130
7919...	56934.888194	10.	10.	RADIUS	183 Access-Request id=131
7919...	56934.890078	10.	10.	RADIUS	708 Access-Accept id=131
7919...	56934.972462	10.	10.	RADIUS	351 Accounting-Request id=130
7919...	56934.983281	10.	10.	RADIUS	62 Accounting-Response id=130
7921...	56939.098470	0.0.0.0	255.255.255.255	DHCP	590 DHCP Discover - Transaction ID 0xa50f5d20
7921...	56939.098470	0.0.0.0	255.255.255.255	DHCP	590 DHCP Discover - Transaction ID 0xa50f5d20
7921...	56940.106718	172.	172.	DHCP	379 DHCP Offer - Transaction ID 0xa50f5d20
7921...	56940.106718	172.	172.	DHCP	379 DHCP Offer - Transaction ID 0xa50f5d20
7921...	56940.734101	0.0.0.0	255.255.255.255	DHCP	590 DHCP Request - Transaction ID 0xa50f5d20
7921...	56940.734101	0.0.0.0	255.255.255.255	DHCP	590 DHCP Request - Transaction ID 0xa50f5d20
7921...	56940.738199	172.	172.	DHCP	379 DHCP ACK - Transaction ID 0xa50f5d20
7921...	56940.738199	172.	172.	DHCP	379 DHCP ACK - Transaction ID 0xa50f5d20
7921...	56940.748767	10.	10.	RADIUS	381 Accounting-Request id=131
7921...	56940.755786	10.	10.	RADIUS	62 Accounting-Response id=131

Figure 5.43: IP phone authentication flow.

As can be seen from Figure 5.44, the ISE server tries to propose the PEAP authentication method. However, the phone, being configured to use EAP-MD5, responds to the server with a NAK message (refer to Section 5.1.5). Furthermore, in this response packet, the IP phone proposes its own authentication



```
▼ Extensible Authentication Protocol
  Code: Response (2)
  Id: 128
  Length: 6
  Type: Legacy Nak (Response Only) (3)
  Desired Auth Type: MD5-Challenge EAP (EAP-MD5-CHALLENGE) (4)
```

Figure 5.44: NAK packet example.

method (i.e., in our case study EAP-MD5). In case the server accepts it, the authentication process proceeds. In the next chapter another discussion regarding IP phones issue is discussed (refer to Section 6.2.1).

### Monitor mode

Before deploying within the full production environment, it is needed to ensure that the architecture outlined is 100% functional. It is therefore common to initially rely (as in IC) on a testing environment, to simulate the behavior of the designed solution as if it were in a real environment. That said, when implementing NAC a step-by-step development approach is required. As a first step it is possible to implement the AAA function, without however being applied drastic restrictions (e.g., user cut off from the network). In this mode it is therefore possible to carry out the first troubleshooting operations, modifying any incorrect policy conditions. In this context, the only authorization profile associated with those who attempt to authenticate is Access-Accept. Any device and user can therefore generate traffic on the network without being subject to particular limitations.

### Low-impact mode

Once familiar with the management of the system, the network administrator switches from an open to a limited mode. This means that restrictions on access and authorizations are implemented. This mode is to be considered a halfway between monitor and closed mode. Although closed mode is considered the final mode to be implemented in a production environment, in reality it is not. In fact, the choice on which scenario to develop is strictly correlated with the needs of the company. In a context where devices need to communicate with entities within the company before authenticating, this is the method to consider. An example of such a scenario is when devices are equipped with Preboot Execution Environment (**PXE**). A certain level of access to the network is therefore allowed before the authentication phase. Beginning from the base provided by the monitoring mode framework, limitations are added such as PACLs. Through them, traffic such as DHCP can be allowed to allow the device to acquire an IP address. Remember as in the previous section, access was granted to everyone via MAB. In that case, everyone could get access, but the latter was limited.

### Closed mode

While in the previous modes the network traffic could be passed before the sender was authenticated, with this mode a limiting approach is applied. In fact, any packet (other than EAP, CDP or LLDP type) is automatically discarded. It is strongly recommended, to implement this solution only after having thoroughly tested it. In fact, should anomalous events occur, the end user/device would be completely isolated and this would imply a huge disservice. It is therefore common to switch to this implementation only after passing through the previous 2 approaches. Once successfully authenticated, the respective access levels are provided to the applicant through the authorization process.

## 5.2 Authorization

As extensively described above, once the applicant has been authenticated, it is necessary to check which levels of authorization he has. Each user/device can be associated with authorization profiles that limit the possibility of accessing the network. Therefore, passing authentication does not mean having full access to any resource. The authorization process can base its operation on an extensive number of variables such as the type of device, the location from which access was requested, the group to which it belongs, etc. All these attributes, to be part of a security context, must be used within the authorization policies. Figure 5.45 shows an example of an authorization policy.

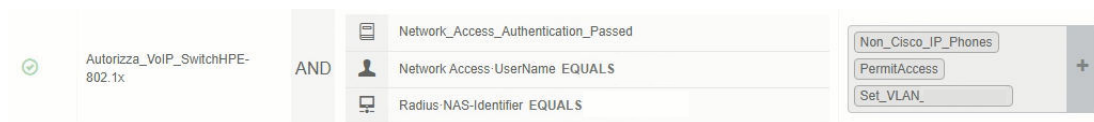


Figure 5.45: Authorization policy example.

In the example shown in the figure, the conditions to obtain the defined profiles must be compliant towards the conditions specified in the central part of the figure. In this case the conditions are related to each other by an “AND” condition. It is however possible to exploit other types of logical operators. In the case in which there is only a single condition, it is denoted as “*simple condition*”, while in the case in which there are several conditions linked together (as in this case) they are defined as “*compound conditions*”. The 3 conditions are:

- Authentication successful;
- Name of the 802.1X supplicant;
- Name of the NAD with which it presents itself to the NAC server.

To ensure that access is restricted, multiple configuration tools are used. In addition to dACLs and VLAN assignment, there are other types of tools:

- Security Group;

- Web Redirection;
- Re-authentication;
- MACsec Policy;
- Voice Domain Permission (used in the next chapters).

The concepts of authentication and authorization have therefore been introduced, but how does the actual transfer of the information relating to the associated profile take place? Consider the scenario where the profile contains information about the VLAN and dACL. The first step to consider is the one related to the reception of the Access-Accept RADIUS message. In the example Figure 5.46 it is shown how the association of the VLAN to the current session occurs through the tunneled attributes (i.e., attributes n. 64, 65 and 81). On the other hand, the association of the dACL occurs through the attribute 26 (VSA). In the case in question, since the NAD is a Cisco device, an attribute defined by this vendor is used. This attribute is called “*CiscoSecure-Defined-ACL*” and the associated value follows the following format “*#ACL#-IP-name-number*”. From the figure it is possible to see how the name is the one defined within ISE (i.e., “*Deny-Ping-ALL*”). The number instead corresponds to the version number. Once the

```

▼ RADIUS Protocol
  Code: Access-Accept (2)
  Packet identifier: 0xc (12)
  Length: 405
  Authenticator: 4c4036787d100906ae37c8ad26227fbc
  [This is a response to a request in frame 937643]
  [Time from request: 0.030150000 seconds]
  ▼ Attribute Value Pairs
    > AVP: t=User-Name(1) l=14 val=IC
    > AVP: t=Class(25) l=58 val=434143533a304130313230393630303030463838463742323841433a766163696973...
    > AVP: t=Session-Timeout(27) l=6 val=1000
    > AVP: t=Idle-Timeout(28) l=6 val=3600
    > AVP: t=Termination-Action(29) l=6 val=RADIUS-Request(1)
    > AVP: t=Tunnel-Type(64) l=6 Tag=0x01 val=VLAN(13)
    > AVP: t=Tunnel-Medium-Type(65) l=6 Tag=0x01 val=IEEE-802(6)
    > AVP: t=EAP-Message(79) l=6 Last Segment[1]
    > AVP: t=Message-Authenticator(80) l=18 val=8147bfb4589bcce107e475a4a62034cb
    > AVP: t=Tunnel-Private-Group-Id(81) l=6 Tag=0x01 val=601
    > AVP: t=EAP-Key-Name(102) l=67 val=\031b\037&\06',\0Kk\0r\0P\0\036q\026\u0095\u07B4\0
    ▼ AVP: t=Vendor-Specific(26) l=70 vnd=ciscoSystems(9)
      Type: 26
      Length: 70
      Vendor ID: ciscoSystems (9)
      ▼ VSA: t=Cisco-AVPair(1) l=64 val=ACS:CiscoSecure-Defined-ACL=#ACSACL#-IP-Deny-Ping-ACL-62a9cb5f
        Type: 1
        Length: 64
        Cisco-AVPair: ACS:CiscoSecure-Defined-ACL=#ACSACL#-IP-Deny-Ping-ACL-62a9cb5f
    > AVP: t=Vendor-Specific(26) l=58 vnd=Microsoft(311)
    > AVP: t=Vendor-Specific(26) l=58 vnd=Microsoft(311)

```

Figure 5.46: RADIUS Access-Accept packet including dACL and VLAN association.

Access-Accept message has been received, the NAD has the task of applying the configurations indicated to it through the attributes. When the NAD receives new dACLs (i.e., not previously configured), it proceeds by sending a RADIUS



Accounting-Request message in which it requests the ACEs that make up the dACL. From Figure 5.47 it is possible to see how this request involves the specific attributes of the vendor. In particular the attributes are set as “*aaa:service=ip-admission*” and “*aaa:event=acl-download*”. Furthermore, the RADIUS Username attribute is fundamental, with which the NAD specifies to ISE which dACL it is requesting information for. Once the request has been received, ISE responds via

```

▼ RADIUS Protocol
  Code: Access-Request (1)
  Packet identifier: 0xd (13)
  Length: 142
  Authenticator: 553a7ed4e42ab4c65cd708d76dd7df99
  [The response to this request is in frame 937646]
▼ Attribute Value Pairs
  > AVP: t=NAS-IP-Address(4) l=6 val=10.
  > AVP: t=User-Name(1) l=36 val=#ACSACL#-IP-Deny-Ping-ACL-62a9cb5f
  ▼ AVP: t=Vendor-Specific(26) l=32 vnd=ciscoSystems(9)
    Type: 26
    Length: 32
    Vendor ID: ciscoSystems (9)
    ▼ VSA: t=Cisco-AVPair(1) l=26 val=aaa:service=ip_admission
      Type: 1
      Length: 26
      Cisco-AVPair: aaa:service=ip_admission
    ▼ AVP: t=Vendor-Specific(26) l=30 vnd=ciscoSystems(9)
      Type: 26
      Length: 30
      Vendor ID: ciscoSystems (9)
      ▼ VSA: t=Cisco-AVPair(1) l=24 val=aaa:event=acl-download
        Type: 1
        Length: 24
        Cisco-AVPair: aaa:event=acl-download
      > AVP: t=Message-Authenticator(80) l=18 val=a9adc0fedbbda5c4c2e5ad5f6f9f9e5e

```

Figure 5.47: RADIUS Accounting-Request packet including dACL request information.

Access-Accept RADIUS packets containing the ACEs relating to the specified dACL. These ACEs are contained within the Cisco VSAs. Through the value “*ip:inacl#1=deny icmp any any*” (shown in Figure 5.47), it is specified that the first entry of the ACL applied to the session corresponds to the block of any ICMP packet. Considering ISE instead, from Figure 5.49 it is possible to see how the download of the dACL associated with the authorization profile is reported. Remember how the reading of these logs is from the bottom up. The correct authentication of the user is indicated in the last line of the figure (in fact, the username starting with the nomenclature “*IC*” representing the individual employee is shown as a credential). The name of the dACL is shown in the line above. Finally, the first one shows the session relating to the user. The configurations of the policy sets (and therefore of the relative authorization policies/profiles) are reported in Section 5.4.

```

▼ RADIUS Protocol
  Code: Access-Accept (2)
  Packet identifier: 0xd (13)
  Length: 231
  Authenticator: aa4a6c8354c312f388a98a83e8c99131
  [This is a response to a request in frame 937645]
  [Time from request: 0.001597000 seconds]
  ▼ Attribute Value Pairs
    > AVP: t=User-Name(1) l=36 val=#ACSACL#-IP-Deny-Ping-ACL-62a9cb5f
    > AVP: t=Class(25) l=85 val=434143533a3061303530343432526a4c45477a796768774c504f35755f61764f5871616a...
    > AVP: t=Message-Authenticator(80) l=18 val=ce1bf312876356fb363a50bb94e7d626
    ▼ AVP: t=Vendor-Specific(26) l=36 vnd=ciscoSystems(9)
      Type: 26
      Length: 36
      Vendor ID: ciscoSystems (9)
      ▼ VSA: t=Cisco-AVPair(1) l=30 val=ip:inacl#1=deny icmp any any
        Type: 1
        Length: 30
        Cisco-AVPair: ip:inacl#1=deny icmp any any
      ▼ AVP: t=Vendor-Specific(26) l=36 vnd=ciscoSystems(9)
        Type: 26
        Length: 36
        Vendor ID: ciscoSystems (9)
        ▼ VSA: t=Cisco-AVPair(1) l=30 val=ip:inacl#2=permit ip any any
          Type: 1
          Length: 30
          Cisco-AVPair: ip:inacl#2=permit ip any any

```

Figure 5.48: RADIUS Access-Accept packet including ACEs.

0	IC	F4:30:B9:	Unknown	802.1X_Wir...	802.1X_Wir...	Employee_C...
	#ACSACL#-IP-De...					
	IC	F4:30:B9:	Unknown	802.1X_Wir...	802.1X_Wir...	Employee_C...

Figure 5.49: Authorization profile download in ISE interface.

## 5.3 Accounting

The accounting function allows a central node to keep track of the actions performed by individual end devices/users. In ISE, login, CoA and identification operations can be tracked throughout the entire network connection. However, the accounting feature (via RADIUS) finds an important functionality in profiling. In fact ISE employs accounting packets to recognize the device that connects to the NAD.

## 5.4 AAA service set-up

The following 3 sections show the configurations with the relative descriptions of the AAA service for the 3 components of the NAC infrastructure. Before moving on to the description, note that the configurations relating to the management of licenses, certificates and anything not strictly related to the work of the thesis have been omitted.

### 5.4.1 ISE set-up

Starting from the configurations relating to ISE, some have already been mentioned previously. First it is needed to register the NADs with which the access to users/devices is provided. Figure 5.50 shows the interface with the relative configurations. The associated IP address is the one with which the RADIUS client interfaces with the RADIUS server (respective configuration in the drop-down menu).

The screenshot displays the 'Network Devices' configuration page. At the top, there are navigation links for 'Network Devices List > IC' and 'SwNAC1'. The main section is titled 'Network Devices'. It contains several input fields: 'Name' with the value 'IC SwNAC1', 'Description' with 'switch test NAC', 'IP Address' with a dropdown menu and a sub-field for the IP address '32', 'Device Profile' set to 'Cisco', 'Model Name' set to 'Unknown', and 'Software Version' set to 'Unknown'. Below these fields is a section for 'Network Device Group' with four expandable settings: 'RADIUS Authentication Settings' (checked), 'TACACS Authentication Settings' (checked), 'SNMP Settings' (unchecked), and 'Advanced TrustSec Settings' (unchecked). At the bottom, there are 'Save' and 'Reset' buttons.

Figure 5.50: Network device configuration interface.

Once the NAD has been registered, it is the turn of the configurations of the policies. For a complete understanding, Figure 5.51 shows the flow to which a device/user who wishes to authenticate is subjected.

### Authentication policies

The definition of authentication policies can be mainly broken down into 2 components. The first are the policy sets where a first level of granularity can be defined. The access request is hence examined and redirected to the correct policy set. The division between the various policy sets is at the discretion of the network administrator. However, it must follow a logical thread, in order to make the division as efficient as possible. Once it has been identified which policy set to forward the request to, another level of granularity is presented. Once authenticated, the request is subjected to authorization policies. For authentication processes it is important to remember that the native *Windows 10* supplicant was used (as shown in the Section 5.4.3). During testing, it has been verified that the supplicant's native timeout is set to 600 seconds. In fact, once a failure EAP packet (i.e., EAP code 4) is received, the *Windows* machine stops responding to requests from the Request-Identity packets (sent by the NAD).

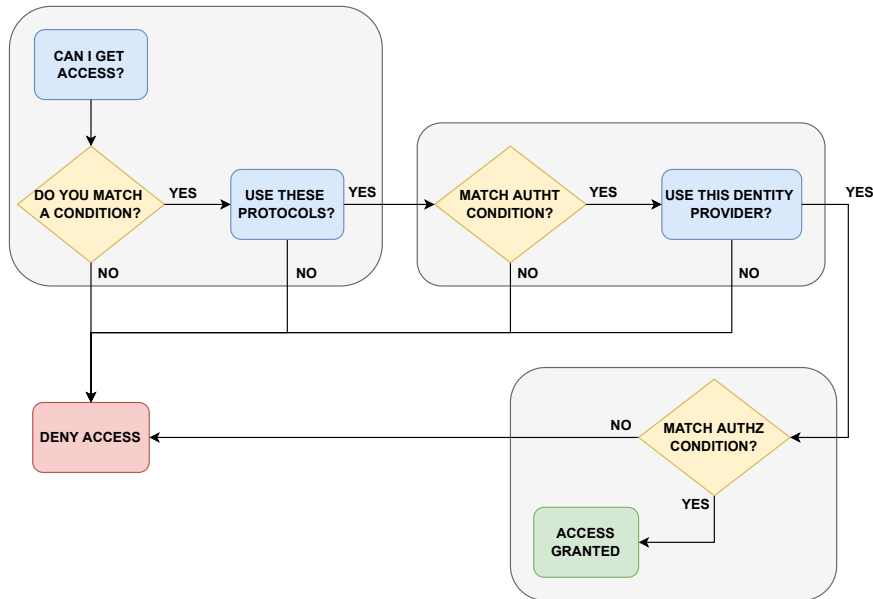


Figure 5.51: AAA flow between policies conditions.

Before reporting the configurations made, it is important to underline that the analyses and deployments concerns the wired infrastructure, leaving the wireless one as a future work. As an initial design phase, it has been decided to create two policy sets, one for 802.1X and one for MAB. The conditions (i.e., smart conditions) to use for these two sets are defined by default within ISE. Depending on the type of NAD vendor, these conditions use different comparison values (as described above). On the other hand, as regards the permitted protocols, the operating strategy chosen turns out to be different. While with 802.1X the default protocols are allowed (i.e., all authentication methods are allowed), with the MAB only the “*Host Lookup*” option is enabled. Remember that corporate PCs are provisioned through a GPO (shown in Figure 5.34). In the event that a device should authenticate itself using another method, this will be managed by the first policy set shown in Figure 5.52. Obviously, in the production environment, there will be others policy sets whose purposes will be different. The authentication

Policy Sets Reset Policyset Hitcounts Reset Save

+	Status	Policy-Set Name	Description	Conditions	Allowed Protocols / Server Sequence	Hits	Actions	View
		802.1X_Wired		Wired_802.1X	Default Network Access x +	2485		
		MAB_Wired		Wired_MAB	ONLY-MAB x +	2954		
		Default	Default policy set		Default Network Access x +	456		

Figure 5.52: Policy sets configuration.

policies relating to 802.1X and the MAB are presented respectively in Figure 5.53 and Figure 5.54.

Unlike 802.1X, MAB does not use a sophisticated authentication policy. In fact, being conceived to guarantee access (albeit limited) to each user/device

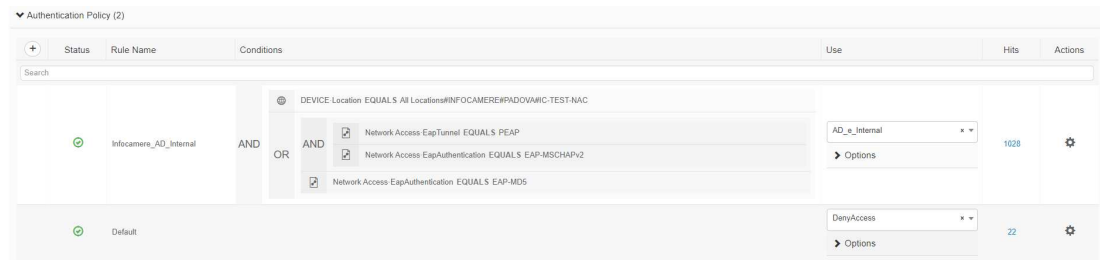


Figure 5.53: 802.1X authentication policies.

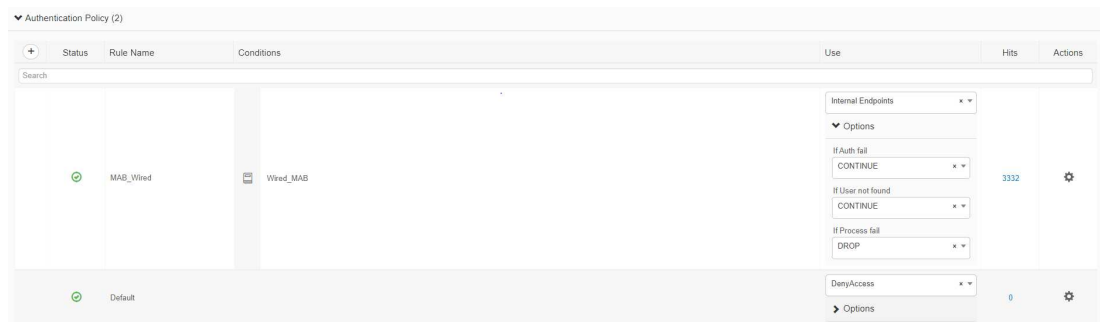


Figure 5.54: MAB authentication policies.

that connects, it cannot contain sophisticated rules. The same can also be seen from the options available for authentication (right side panel). Even a new device, potentially not managed by the company, can access the network thanks to the conditions of “*continue*”. However, through the authorization policies, it is possible to use the results of these checks to restrict access to the network. A certain policy can therefore be reserved only for those who have successfully passed the authentication (and therefore in the same way also one for those who have not provided the correct ones). As with 802.1X, the default rule is placed last in the list. Unlike the rules defined by the administrator, this cannot be deleted or disabled.

### Authorization profiles

Before carrying out the actual configuration of the authorization policies, to have a certain level of granularity, it is necessary to create authorization profiles. These profiles are associated with users/machines, who will be subject to the limitations related to that profile. An example of profile configuration is shown in Figure 5.55. The end device/user to which this authorization profile will be associated with, will be allocated in a specific VLAN (in this case obscured as part of the implementation of the production environment).

With reference to Figure 5.45, any device that is compliant with all 3 conditions reported would acquire the following profile. As will be shown in following chapters, conditions can contain attributes related to both profiling and posture. An example of attributes that can be used as conditions within policies is shown in Figure 5.56.

Authorization Profiles > [Set\\_VLAN\\_ArubaWired](#)

### Authorization Profile

\* Name:

Description:

\* Access Type:

Network Device Profile:

---

▼ **Common Tasks**

Security Group

VLAN      Tag ID:        ID/Name:

Figure 5.55: Authorization profile interface.

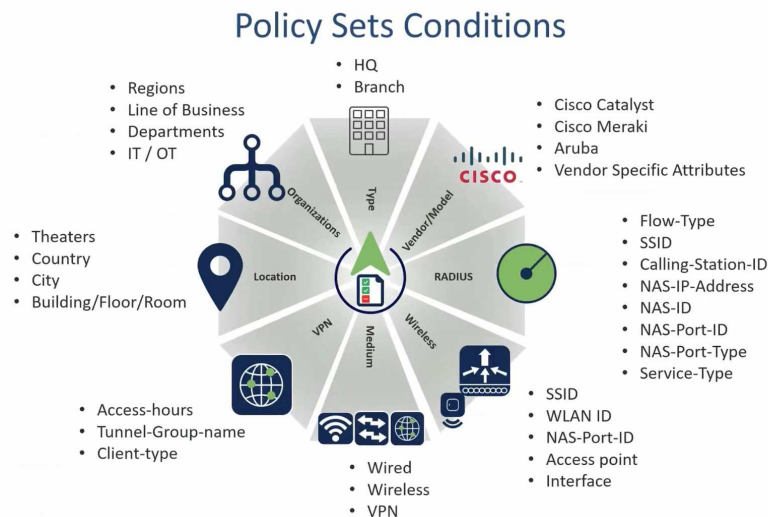


Figure 5.56: Example of policy conditions.

In Figure 5.57 the authorization policies related to 802.1X are shown. Therefore, considering that the various wired devices support an 802.1X configuration, different authorization profiles will be used. As mentioned previously, the authorization profiles are composed by various logical objects including dACLs. An example of these (for the case of IP phones) is shown below:

```

permit udp any host <PRIMARY-DNS-IP> eq 53
permit udp any host <SECONDARY-DNS-IP> eq 53
permit udp any <IP-SWITCHBOARD> 0.0.0.7 range <PORT RANGE>
permit udp any range <PORT RANGE> any range <PORT RANGE>
permit udp any eq 5060 any eq 5060
permit udp any eq 5061 any eq 5061

```

```

permit udp any eq 68 any eq 67
permit udp any any eq 123
permit tcp any eq 443 any established
permit tcp any <IP-SWITCHBOARD> 0.0.0.7 eq <PORT RANGE>
permit tcp any eq 80 any established
deny ip any any

```

In summary, DHCP, Session Initiation Protocol (**SIP**), HTTP, DNS, LDAP and Real Time Protocol (**RTP**) traffic is allowed. For visual feedback, referring to Figure 5.43, it is possible to see how DHCP traffic is allowed after having assigned the authorization profile. A critical aspect, initially not considered, is the management of traffic to the IP phone. In fact, in some cases, the IP phone must be accessible from any location within the company intranet. The VoIP technician may need to interface with the device GUI. In order to allow this operation, the second-last ACE has been introduced. The keyword *established* indicates that established HTTPS connections can only come from the outside (i.e., from the PC used by the VoIP technician). So the TCP traffic that is firstly generated by the *Polycom* IP phone to the outside will be dropped. Finally, remember that these ACLs are assigned to the session. Once completed, they are no longer present within the NAD.

Authorization Policy (6)				Results		Hits	Actions
Status	Rule Name	Conditions	Profiles	Security Groups			
✔	Employee	AND OR Network Access UserName CONTAINS host AD_IC_ExternalGroups EQUALS Cisco_Authn_Passed	Employee_Cisco	Select from list	539	⚙️	
✔	VoIP	AND Network Access UserName EQUALS ic Cisco_Authn_Passed	IP_Phone_Cisco	Select from list	0	⚙️	
✔	Printer	AND Network Access NetworkDeviceName EQUALS ic Cisco_Authn_Passed	Printer_Cisco	Select from list	0	⚙️	
✔	IP-Camera	AND Network Access NetworkDeviceName EQUALS ic Cisco_Authn_Passed	IP_Camera_Cisco	Select from list	0	⚙️	
✔	Multimedia	AND OR Network Access UserName EQUALS ic Network Access UserName EQUALS ic Network Access UserName EQUALS ic Cisco_Authn_Passed	Multimedia_Cisco	Select from list	0	⚙️	
✔	Default		DenyAccess	Select from list	25	⚙️	

Figure 5.57: 802.1X authorization policies.

Authorization Policy (2)				Results		Hits	Actions
Status	Rule Name	Conditions	Profiles	Security Groups			
✔	Contractor	AND InternalUser_IdentityGroup EQUALS User Identity Groups:GuestType_Contractor (default) Cisco_Authn_Passed	Contractor_Cisco	Select from list	443	⚙️	
✔	Default		Isolated_Cisco	Select from list	269	⚙️	

Figure 5.58: MAB authorization policies.

Regarding MAB, there are two authorization policies. The first, in which it



is checked whether the user is a contractor. The presence of a figure like this one must not be trivialized. For this reason, each consultant must be able to access the network with their devices to carry out their activities. This translates into creating an ad-hoc authorization profile. This profile was defined as follows:

- Assigning an ACL (described below) to define at the access level the services it does not need to access;
- Assignment to a reserved VLAN, in order to separate the related traffic flows from the production environment. However, it is important to remember that other security perimeters are implemented (i.e., NGFW) with which a more detailed analysis of the traffic is carried out.

The defined ACL consists of 3 entries. They do not allow traffic to respectively:

- Servers containing personal information of IC employees (therefore to the services available to the personnel administration);
- Databases containing information for statistical use;
- Servers/databases containing information on business registers. Particular attention should be paid to this latter aspect as they represent the core business of the company.

As for the policies relating to the 802.1X policy set, they contain 2 conditions:

- One with which it is verified that the authentication was successful (and therefore the user is registered);
- One with which the type of authorization is discriminated on the basis of the type of entity requesting it. An IP phone, authenticating with certain credentials, cannot be associated with an authorization profile of a printer.

In general, the configurations made were defined using a Cisco device as NAD. The configurations and commands presented throughout the study therefore present a slight difference with those used for example for Aruba devices.

## 5.4.2 NAD set-up

The configuration of the Cisco device relating to authentication is shown below. In addition, some configuration commands are described for a better understanding of the whole process.

### Authentication service set-up

The configuration implemented within the NAD is presented below.



```
switchport access vlan 200
switchport mode access
switchport voice vlan 201
device-tracking attach-policy DeviceTrackingPolicy
ip access-group WELCOMEACL in
authentication event fail action next-method
authentication host-mode multi-auth
authentication open
authentication order dot1x mab
authentication priority dot1x mab
authentication port-control auto
authentication periodic
authentication timer reauthenticate server
authentication timer restart 10
authentication timer inactivity server
mab
dot1x pae authenticator
dot1x timeout quiet-period 18
dot1x timeout tx-period 1
dot1x max-reauth-req 3
```

During the first phase of deployment, it was decided (following best practices) to adopt an open architecture using the following command:

```
authentication open
```

The messages generated by the devices will therefore not be subject to restrictions. However, after a comparison between the network group and the SOC, it has been decided to implement an ACL at the interface level. This ACL is called “*WELCOMEACL*” and its content has been modified during the various test phases to allow multiple troubleshooting operations. To cite an example, initially the ACL only contained an ACE like the following one:

```
permit ip any any
```

Also note how the ACL is implemented as inbound. Therefore only the traffic coming from the device is inspected. Furthermore, 2 VLANs (Data and Voice domain) have been configured to manage the first access of users. Initially these VLANs were not equipped with neither an L3 interface nor a DHCP relay. They were also not enabled on the trunk interface. This choice was made to contain all the messages coming from the devices just connected to the network. A Remediation VLAN (VLAN 400) was then created, to allow the devices to remedy the items found not to be compliant (especially for posture purposes). Further configurations can be added to this VLAN, which would increase the security level. This discussion is addressed at Section 8.2.3. In an advanced phase of analysis and

implementation, the ACL presented above must be redefined in such a way as to obtain a low-impact deployment. In case a closed mode architecture is requested, the command “*authentication open*” must not be used. It is important to specify, however, that in a deployment of this kind, some types of messages can transit even in the event of failure of authentication. The types are EAP, CDP, STP and LLDP. With the “*dot1x*” set of commands, network administrator can configure timers on which 802.1X authentication process is based. With the “*dot1x timeout tx-period 1*” command, switch sends Request/Identity messages every 1 second. The total number of retry with which NAD try to re-authenticate the device is configured by using “*dot1x max-reauth-req 3*” command. Furthermore, using the “*authentication host-mode multi-auth*” command, network administrator specify that for each interface will be connected multiple devices on Data domain and just 1 device in Voice domain. In fact, in the final implementation, only one IP phone needs to be connected to the NAD interface. On the contrary, there is the possibility that multiple devices such as PCs can connect to the same interface.

Some attributes will be downloaded from ISE, such as re-authenticate and inactivity timer. With the former, ISE specify the time period that it will use for re-authentication. This item does not be confused with the previous one. In fact this will be used once the device is authenticated. The inactivity timer instead is used to clear the session with a particular device which has not generated traffic for a certain period of time. As a final notation, note that although the NAD is an access level device there is no trace of the port-security function. This function allows to adopt a security level based exclusively on the device that connects to the interface. It can be considered as a basic NAC tool. However, the implementation of both these features is highly discouraged because together they represent a source of numerous issues.

### 5.4.3 Supplicant set-up

As previously said, the configurations of the devices (considering PCs) are subject to GPOs. This means that it is not possible to change configurations without being a user belonging to a special group within the ADs. Having said that, below is reported the configuration of a PC regarding the 802.1X supplicant. In Figure 5.59 it is possible to see how the 802.1X supplicant is enabled. Actually, it is necessary to set another option, configuring the authentication service and making it automatic (as shown in Figure 5.60). In this way, at each boot, the 802.1X process starts autonomously. In Figure 5.59 and Figure 5.61 the remaining configurations are shown. Note how the parameters agree with what is described in the Section 5.1.7.

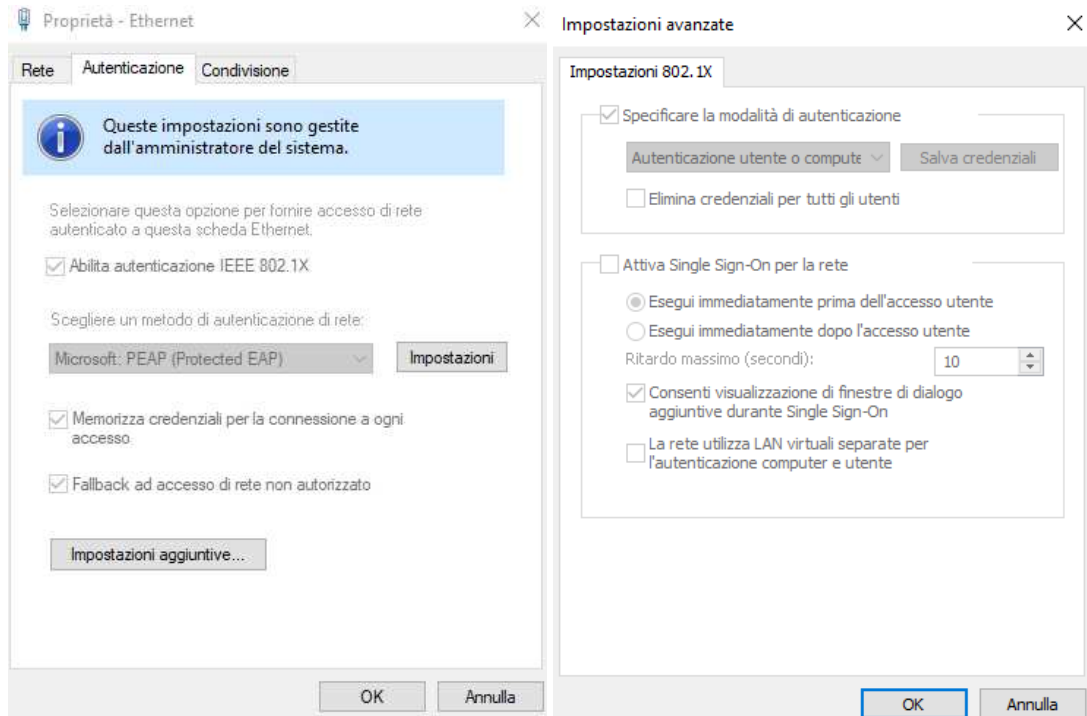


Figure 5.59: Windows 802.1X supplicant configuration.



Figure 5.60: Windows 802.1X supplicant configuration.

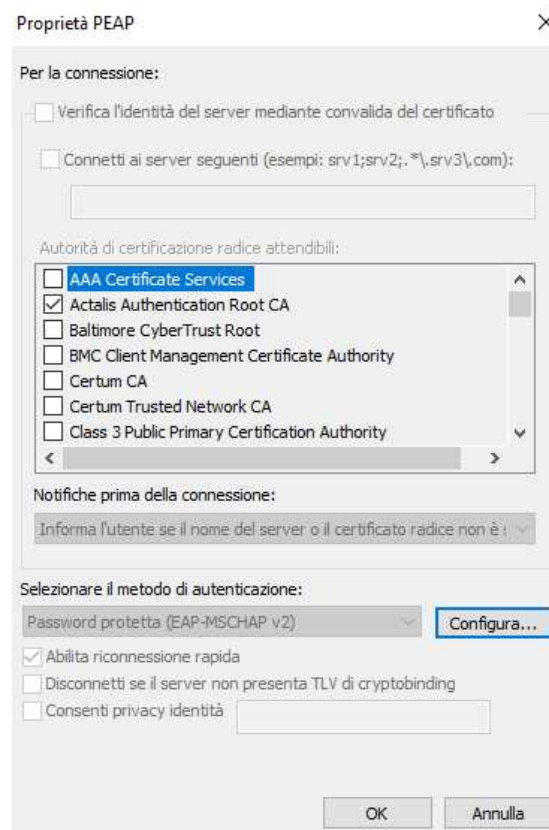


Figure 5.61: Windows 802.1X supplicant configuration.

# Chapter 6

## Profiling

One of the most important functions, on which the thesis work is based, is profiling. Cisco ISE has the ability to determine with some level of certainty the type of device attempting to access the network. This function is implemented by ISE monitoring the packets containing certain protocols. From these, for inspection, attributes are obtained from which the respective senders are classified. In a particular scenario (described in detail in the following sections) these attributes are obtained through tools called “*probes*”. The level of detail of the profiling function can be defined by the administrator. It is in fact possible to define different levels of granularity: for example, it is possible to pass from recognizing a generic device and cataloging it according to the vendor to which it belongs, or it can be defined down to the detail of the type of model. Figure 6.2 shows the example of an interface in which it is possible to see some of the thousands of profiling policies present in the ISE local database, while Figure 6.1 shows how to enable the profiling function from the configuration panel through the ISE GUI.

The profiling policies present in this list are partly provided by the vendor (therefore pre-existing and with the respective configurations), while the others are customized. The reason why it is necessary to create new ones can be for example to add a certain level of detail in the recognition of devices. In fact, the recognition in detail of some devices may be required in order to make the AAA process as granular as possible. Furthermore, it is possible that the devices to be profiled do not have an associated policy in the default list. The profiling function is therefore mainly used for 3 main purposes:

- Provide a more granular type of access;
- Context visibility of network infrastructure attached devices;
- Improved troubleshooting process.

In the first case, profiling can be used within the policy sets as a condition. In this way a user who tries to access with a certain device (e.g., a workstation or a laptop) is able to correctly access a portion of the internal network, while using another type of device (e.g., a tablet) can no longer access it. Regarding the second point, the company may need to know the number and type of devices

Deployment Nodes List >  
**Edit Node**

General Settings    Profiling Configuration

---

Hostname  
 FQDN **.net.infocamere.it**  
 IP Address  
 Node Type **Identity Services Engine (ISE)**

---

Role **STANDALONE** **Make Primary**

Administration

Monitoring

Role **PRIMARY**

Other Monitoring Node

Dedicated MnT

Policy Service

Enable Session Services

Include Node in Node Group **None**

**Enable Profiling Service**

Enable Threat Centric NAC Service

Enable SXP Service

Enable Device Admin Service

Enable Passive Identity Service

Figure 6.1: ISE node settings.

Identity Services Engine    Home    Context Visibility    Operations    Policy    Administration    Work Centers

Policy Sets    Profiling    Client Provisioning    Policy Elements

Profiling

Profiling Policies

Selected 0 | Total 645

Edit    Add    Duplicate    Delete    Import    Export

Show All

Profiling Policy Name	Policy Enabled	System Type	Description
<input type="checkbox"/> 2Wire-Device	Enabled	Cisco Provided	Policy for 2Wire-Device
<input type="checkbox"/> 3Com-Device	Enabled	Cisco Provided	Policy for 3Com-Device
<input type="checkbox"/> Aastra-Device	Enabled	Cisco Provided	Policy for Aastra-Device
<input type="checkbox"/> Aastra-IP-Phone	Enabled	Cisco Provided	Policy for Aastra-IP-Phone
<input type="checkbox"/> Aerohive-Access-Point	Enabled	Cisco Provided	Policy for Aerohive-Access-Point
<input type="checkbox"/> Aerohive-Device	Enabled	Cisco Provided	Policy for Aerohive-Device
<input type="checkbox"/> American-Power-Conversion-Device	Enabled	Cisco Provided	Policy for American-Power-Conv
<input type="checkbox"/> Android	Enabled	Cisco Provided	Policy for all Android SmartPhor
<input type="checkbox"/> Android-Amazon	Enabled	Cisco Provided	Policy for Android Amazon
<input type="checkbox"/> Android-Amazon-Kindle	Enabled	Cisco Provided	Policy for Android-Amazon-kind
<input type="checkbox"/> Android-Amazon-Phone	Enabled	Cisco Provided	Policy for Android Amazon Phor
<input type="checkbox"/> Android-Amazon-TV	Enabled	Cisco Provided	Policy for Android Amazon TV
<input type="checkbox"/> Android-Asus	Enabled	Cisco Provided	Policy for Android-Asus
<input type="checkbox"/> Android-Google	Enabled	Cisco Provided	Policy for Android-Google
<input type="checkbox"/> Android-Google-Glass	Enabled	Cisco Provided	Policy for Android-Google-Glass
<input type="checkbox"/> Android-HTC	Enabled	Cisco Provided	Policy for Android-HTC

Figure 6.2: Profiling policies interface.

currently connected to the network. This function can be compared to a kind of census. Through it, it is therefore possible to have a snapshot of the network definition. Last but not least is the improvement of the troubleshooting process. In fact, in the event that a device should encounter problems of any kind, having obtained its information, it is possible to immediately check what are the possible causes. The profiling feature (as far as ISE is concerned) is an advanced feature, which requires a certain type of licenses. The licensing issue is presented at Chapter 8.

---

Seeing the presentation of the profiling function, a very important question may arise: is it possible to use this function as an anti-spoofing mechanism while using MAB authentication? Let's consider the following scenario: an IP phone (not equipped with 802.1X supplicant) is connected to the access switch, ready to be used. The device therefore uses MAB to authenticate itself. At this point, an attacker could disconnect the device, connect a malicious one, spoof the MAC address and go unnoticed. Is it therefore possible, through the attributes collected by profiling, to identify this malicious device and isolate it? This is quite a complicated operation, as even the attributes can be modified by means of suitable tools. With the October 2020 release of ISE 2.2, Cisco introduced two new features that address this vulnerability:

- Anomalous Behaviour Detection;
- Anomalous Behaviour Enforcement.

Through anomalous behavior detection, ISE uses profiling together with analyses functions to identify if there have been any changes between before and after a certain event. Some of the characteristics verified by anomalous behavior detection are the following:

- DHCP Class-Identifier: in this scenario, 2 cases can take place. The first concerns devices characterized by static addressing. Often devices such as printers do not use the DHCP service, as administrators manually provide them the address. During the profiling phase this attribute should be null. However, if at a later time, a DHCP message is received from the same MAC address, containing a different value of that same attribute, it is reported as spoofed. In the second case, however, it consists in identifying 2 different values, always relating to the same attribute, at different times. For example, let assume that at first the attribute contained a value related to the “*Cisco-IP-Phones*”, while in a second moment one related to “*Microsoft-Workstation*”. Such a change would have indicated an anomalous event;
- Endpoint policy: as detailed in the previous chapter, each authenticating device is processed across various policies. In the event that a certain device were to be authenticated as a “*Cisco-IP-Phones*” and then pass instead as a “*Microsoft-Workstation*”, an alert would automatically arise. This would describe the scenario presented above where a computer connects instead of an IP phone;
- Port type: this parameter refers to the type of transmission medium used for communication. In fact, if it is identified that a certain MAC address first uses a wired type connection and later a wireless type, it is reported internally. This assumption is in fact legitimate as each NIC has its own MAC address, and therefore the same address cannot be visible even on 2 different media. It should be noted that all vendors now configure the various NICs in such a way as to use values generated via software rather

than those in hardware. This feature, if not known, can lead to large amounts of unnecessary effort during troubleshooting processes.

If an anomalous event is identified, the second function is that of anomalous behavior enforcement. However, to implement a change in access authorization levels, it is necessary to use another feature, closely related to that of profiling, namely “*CoA*” (described in Section 6.1). However, leaving out the functions suitable for identifying a possible threat, it must be remembered that during the phase of defining the policy sets it is necessary to apply a conservative strategy. This means that the possible authorization level must be associated with each requesting user/device. For example, it is useless to assign the same levels of an employee to a printer. This will not have to access the systems part of the data center. If a generic access is compromised, there will be a real threat to the services.

## 6.1 CoA

The RADIUS protocol, as described in the dedicated section, bases its implementation on a client/server architecture. It is therefore always the client who delivers requests to the server. With the creation of the new features (e.g., profiling and postures) a new need was issued by the server to initiate a request to the other node. To give a first example, using the profiling function, once a device has been profiled it is possible to assign it a specific authorization profile. This mechanism is guaranteed through a CoA message, which informs the NAD (the other node of the RADIUS session) to modify the session-related attributes of the end device authentication. Commonly known as CoA, within the respective RFC it is defined as Dynamic Authorization Extensions of the RADIUS protocol. In general, the RADIUS server can therefore force a re-authentication, disable/enable the port to which the device is connected, delete the session and so on. It is important to underline that the extension of the RADIUS is of fundamental importance, as well as for profiling, also for posture. In fact, they are both functions which, after an analysis phase, provide a result that corresponds to the assignment of the device to a certain level of authorizations [27] [28].

### 6.1.1 CoA messages

To force the modification of the attributes related to a session of a user/device, different types of CoA messages can be used. The following are the 2 messages that the RADIUS server is able to send (with the value of the *Code* field associated with the packet):

- 40 Disconnect-Request: request to end the client session;
- 43 CoA-Request: request to change the session attributes.

The clients can instead reply to the server with the following 4 messages:



- 41 Disconnect-ACK;
- 42 Disconnect-NACK;
- 44 CoA-ACK;
- 45 CoA-NACK.

The first type of message is used to end a session. Once the message has been received (sent to the destination UDP port 3799 or 1700), the NAD (e.g., switch) can reply with an ACK (successful operation) or with a NACK (in case the NAD has failed to end the session). Through CoA messages it is therefore possible to change the status of the active sessions. In particular, ISE offers the possibility to define customized CoA typologies. By default, 3 options are available:

- No CoA: in this case no actions are taken into consideration;
- Re-authentication: with this type of CoA the NAD initiates a new authentication process, thus sending a new EAPoL Start message. Once this message is received, the device will send the credentials again. It is important to note, however, that although the authentication sessions are different, their *Session ID* will be the same. This allows ISE to aggregate information relating to the same device;
- Port bounce: this option sets the NAD interface into shutdown state and then reactivates it again. From the point of view of the end device, this simulates the disconnection and connection of the NIC from the network cable. This causes a DHCP discover message request to be re-transmit for most devices. If this type of CoA is used to change the VLAN to a certain device, this operation solves the problems related to its addressing. In fact, it is possible that when a device is reassigned to another VLAN (and therefore to another subnet), it will not be able to request a new IP address, causing connectivity issues.

The generally recommended CoA method is that of re-authentication. In fact, the NAD, for each port, can have multiple connected devices. Then using the port bounce CoA, these devices would be disconnected and then reconnected. This would result in a disservice for end users. In ISE, however, there is a special function such that, if there is more than one MAC address associated with an interface, if the port bounce method is used, it is converted to re-authentication. In any case, the choice of which type must be defined during the design phase.

Some Vendor-Specific Attributes (**VSA**) used within CoA messages are shown in Table 6.1. Using the disable host port type, the interface associated with the session to be terminated is shut down (through the “*shutdown*” command). On the contrary, the type terminate session (to which the Disconnect-Request message corresponds) terminates the session without affecting the state of the interface on which the device is connected. The substantial difference between this method and the disabled port is that, while the first only terminates the

CoA Type	Cisco VSA
Bounce host port	Avpair="subscriber:command=bounce-host-port".
Disable host port	Avpair="subscriber:command=disable-host-port".
Re-authenticate host	Avpair="subscriber:command=reauthenticate" with Avpair="subscriber:reauthenticate-type=last" or Avpair="subscriber:reauthenticate-type=rerun".
Terminate session	Does not require a VSA.

Table 6.1: CoA VSA examples.

session, with the second in addition to terminating it the client is isolated, not allowing it to authenticate itself again. The latter method can be useful in case a device is the source of some problems. The term VSA shown in Table 6.1 indicates a special type of RADIUS attribute (n. 26) with which each vendor can send specific information relating to its devices. A VSA message, contained within an AVP, is made up of:

- Vendor type;
- Vendor length;
- Attribute-Specific.

When a VSA message is contained within an AVP, the *Data* field of the latter contains the unique ID attributed to each vendor. In the event that a RADIUS server does not support them (as specific extensions of the various vendors), it is obliged not to interpret them. As for the re-authentication method, the VSA containing the *Type* attribute serves to specify which behavior to implement during the process:

- last: use the last successful authentication method;
- rerun: the authentication process is initialized from scratch;
- none: if the type is not specified, the default behavior is that associated with the last type.

The CoA function can be applied in 2 different ways: manually or automatically. In the first case, through the live sessions panel, it is possible to carry out one of the following actions:

- Session re-authentication;
- Session termination with port bounce;
- Session re-authentication with rerun;
- Session termination;

- Session termination with port shutdown;
- Session re-authentication with last;
- SAnet Session Query.

For the second case, on the other hand, it is possible to configure a global CoA and one per profile within ISE. To globally enable the CoA function for profiling, it is needed to go through the profiling settings interface. It is also possible to specify the type of CoA based on the profile (e.g., IP phone). The configuration by profile obviously has a higher priority than the global one.

The services described up to now regarding the CoA are based on a fundamental piece of information: the *Session ID*. In fact, in order to perform operations on the sessions it is first of all necessary to identify them. Regarding ISE, this is done through the use of the following attributes:

31 Calling-Station-Id;

44 Acct-Session-Id;

VSA Audit-Session-Id.

Regarding attribute n. 44, note as per the RFC specification, this value is the same in the Accounting-Request-Start and Accounting-Request-Stop messages. This choice facilitates troubleshooting operations for a specific session.

In Figure 6.3 there is an example of re-authentication flow in case a CoA-Request message is used. In this case, through ISE, it was specified that the action corresponding to the CoA-Request was the port bounce. In fact, it is possible to see how after the CoA packets, there are 2 other RADIUS accounting packets. The port bounce operation is also visible from the NAD CLI as a service message (shown below) in which is shown that the link status has changed.

```
SwitchNAC#%LINK-5-CHANGED: Interface GigabitEthernet1/0/46, changed
state to administratively down
SwitchNAC#%LINEPROTO-5-UPDOWN: Line protocol on Interface
GigabitEthernet1/0/46, changed state to down
```

Once the ISE has been notified that the session has ended, the re-authentication process restarts from the beginning. If a re-authentication CoA type message has been used, the 2 accounting messages would not be present. In fact, these two are RADIUS Accounting-Request-Stop messages. In Figure 6.4 the content of the CoA request is presented.

## 6.2 Probes

The feature of profiling, to recognize devices, collects attributes from the various network protocols encapsulated within the packets. These are then analyzed to determine the type of connected device. Figure 6.5 shows an overview of

252624	437...	10.	10.	RADIUS	232 CoA-Request id=27
252625	437...	10.	10.	RADIUS	111 CoA-ACK id=27
252626	437...	Cisco_	LLDP_Multicast	LLDP	60 MA/00:2c:c8: IN/Gi1/0/ 0
252627	437...	Cisco_	CDP/VTP/DTP/PagP/UD...	CDP	108 Device ID: IC SwnAC1.net.infocamere.it Port ID: GigabitEthernet1/0/
252628	437...	10.	10.	RADIUS	745 Accounting-Request id=125
252630	437...	10.	10.	RADIUS	62 Accounting-Response id=125
253826	438...	YealinkX_	LLDP_Multicast	LLDP	60 NA/0.0.0.0 NA/80:5e:c0: 0
253827	438...	YealinkX_	CDP/VTP/DTP/PagP/UD...	CDP	114 Device ID: T40G805EC0 Port ID: WAN PORT
253828	438...	Cisco_	CDP/VTP/DTP/PagP/UD...	CDP	491 Device ID: IC SwnAC1.net.infocamere.it Port ID: GigabitEthernet1/0/
253829	438...	Cisco_	YealinkX_	EAP	60 Request, Identity
253830	438...	YealinkX_	Nearest-non-TPMR-br...	EAP	60 Response, Identity
253832	438...	10.	10.	RADIUS	303 Access-Request id=249
253833	438...	10.	10.	RADIUS	170 Access-Challenge id=249
253834	438...	Cisco_	YealinkX_	EAP	60 Request, Protected EAP (EAP-PEAP)
253835	438...	YealinkX_	Nearest-non-TPMR-br...	EAP	60 Response, Legacy Nak (Response Only)
253836	438...	10.	10.	RADIUS	379 Access-Request id=250
253837	438...	10.	10.	RADIUS	197 Access-Challenge id=250
253838	438...	Cisco_	YealinkX_	EAP	60 Request, MD5-Challenge EAP (EAP-MD5-CHALLENGE)
253839	438...	YealinkX_	Nearest-non-TPMR-br...	EAP	60 Response, MD5-Challenge EAP (EAP-MD5-CHALLENGE)
253841	438...	10.	10.	RADIUS	395 Access-Request id=251
253842	438...	10.	10.	RADIUS	339 Access-Accept id=251
253843	438...	10.	10.	RADIUS	194 Access-Request id=252
253844	438...	10.	10.	RADIUS	248 Access-Accept id=252
253853	438...	Cisco_	YealinkX_	EAP	60 Success

Figure 6.3: CoA work flow.

```

▼ Attribute Value Pairs
  > AVP: t=NAS-IP-Address(4) l=6 val=10.
  > AVP: t=Calling-Station-Id(31) l=19 val=80:5E:C0:
  > AVP: t=Acct-Terminate-Cause(49) l=6 val=Admin-Reset(6)
  > AVP: t=Event-Timestamp(55) l=6 val=Jul 11, 2022 12:02:04.000000000 ora legale Europa occidentale
  > AVP: t=Message-Authenticator(80) l=18 val=d91938be3bd61d7e8bd30fb212ad723a
  > AVP: t=NAS-Port-Id(87) l=23 val=GigabitEthernet1/0/
  ▼ AVP: t=Vendor-Specific(26) l=43 vnd=ciscoSystems(9)
    Type: 26
    Length: 43
    Vendor ID: ciscoSystems (9)
  > VSA: t=Cisco-AVPair(1) l=37 val=subscriber:command=bounce-host-port
  ▼ AVP: t=Vendor-Specific(26) l=49 vnd=ciscoSystems(9)
    Type: 26
    Length: 49
    Vendor ID: ciscoSystems (9)
  > VSA: t=Cisco-AVPair(1) l=43 val=audit-session-id=0A01209600000E6ECB4B0D4

```

Figure 6.4: CoA-Request packet sample.

the various steps that make up the profiling flow. Such mechanisms are defined “probes”. Each type of probe collects the corresponding traffic to then be analyzed by ISE. Below are reported the various probes that can be implemented:

- RADIUS
- DHCP
- DHCP SPAN
- NMAP
- SNMP QUERY
- SNMP TRAP
- HTTP
- DNS

- NETFLOW

Some of the probes listed above are detailed in the following sections. Figure 6.7

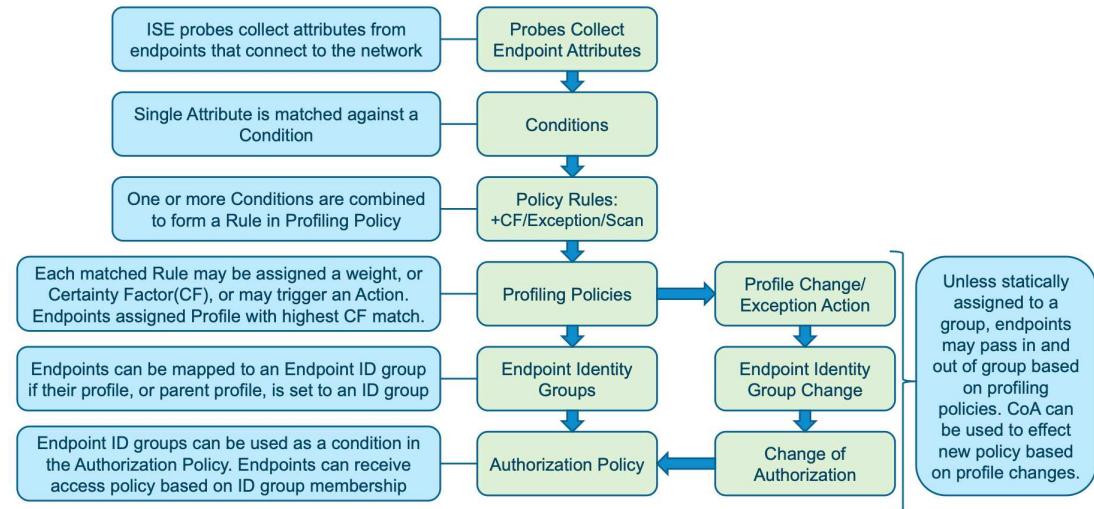


Figure 6.5: Profiling concept flow.

shows the interface with which to enable the various probes. Each probes is configurable with its own parameters which are reported in the following sections. On the other hand, Figure 6.6 shows the general configuration interface for profiling. Note that from this it is possible to configure the Global CoA mentioned in Section 6.1. It is important to underline that not all probes are enabled by default,

**Profiler Configuration**

\* CoA Type:

Current custom SNMP community strings: \*\*\*\*\*

Change custom SNMP community strings:  (For NMAP, comma separated. Field will be cleared on successful saved change.)

Confirm  community strings:  (For NMAP, comma separated. Field will be cleared on successful saved change.)

EndPoint Attribute Filter:  Enabled ⓘ

Enable Anomalous Behaviour Detection:  Enabled ⓘ

Enable Anomalous Behaviour Enforcement:  Enabled

Enable Custom Attribute for Profiling Enforcement:  Enabled

Enable profiling for MUD:  Enabled

Enable Profiler Forwarder Persistence Queue:  Enabled

Enable Probe Data Publisher :  Enabled

Figure 6.6: Profiling configuration interface.

but some of them are active even without having enabled the profiling function. In such scenarios, these services are used for context visibility functions. The probes corresponding to this scenario are RADIUS and HTTP probes. While the RADIUS protocol is used for communication between NAD and authentication server, the HTTP protocol is used to perform redirection operations to portals hosted by ISE. When user profiling is active, the administrator is able to check

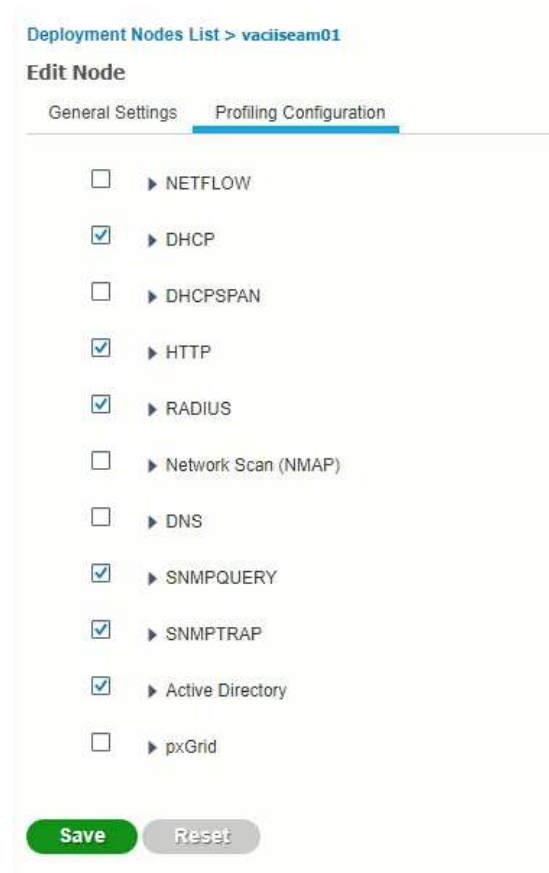


Figure 6.7: Profiling probes configuration.

which is the latest probe that has updated the attributes relating to a device. This check can be done through an attribute called “*EndPointSource*”.

### 6.2.1 RADIUS

The most common and effective probe among those described in this chapter is certainly the RADIUS probe. This protocol is used for communication between NAD and ISE. Communications between these two entities may contain highly relevant information in order to profile the device. Some of these in the form of RADIUS attribute, previously already explained are *Calling-Station-Id*, *Framed-IP-Address*, *Called-Station-Id*, *NAS-Port-Id* and *Service-Type*. An important note to make is to use the *Framed-IP-Address* attribute. In an enterprise environment, IP address allocation occurs dynamically via DHCP. In this scenario it is therefore possible that the same IP address is assigned in a first phase to a certain device, while at a later time to another (through specific configurable parameters within the DHCP server it is however possible to reserve an IP address for a device even after it has been disconnected for a certain period of time). This could cause ISE to incorrectly associate an IP address to a certain device. To deal with this problem, once the RADIUS Accounting-Request-Stop message

is received (RADIUS session termination), the IP is deleted from the ISE internal database. The configuration to implement this type of probe is trivial, having only to activate it from the ISE GUI as shown in Figure 6.8.

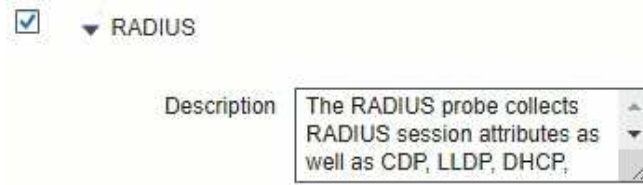


Figure 6.8: ISE RADIUS probe settings.

The use of this type of probe increases its effectiveness in combination with the device sensors (explained in the next section). This feature allows to collect a set of information (coming from multiple sources) and send them through RADIUS Accounting-Request packets to ISE.

### Device Sensor

The device sensors are a particular feature supported by the Cisco Catalyst switches and WLC, with which information about the device is collected and then transmitted to ISE. Devices information is collected through protocols such as CDP, LLDP, DHCP and HTTP. This information is then sent to ISE, through RADIUS Accounting-Request packets. Device sensors are composed of two principal components:

- Collector: component that deals with the collection of information from the end devices;
- Analyzer: component that processes and analyzes the information collected and determines the type of device.

Device sensors are logically connected to clients which can be internal or external. Some examples of internal clients are represented by embedded Device Classifiers (**DC** or local analyzer), Cisco EnergyWise (**EW**) and Cisco Auto SmartPorts (**ASP**). In Figure 6.9 an implementation example of a device sensor is shown. From the same figure it is possible to see how the information is sent both to the local analyzer and to ISE. Once the device has been profiled, the latter sends CoA messages to associate an authorization profile to the device. It is important to underline that this feature is highly dependent on the platform in which it is configured. In the project phase it is therefore necessary to verify that both the hardware and the OS versions of the devices are compliant. In the IC case study, a Cisco Catalyst 3650 switch (OS XE 16.5.1a version) has been used. Some of the advantages over other probes are as follows:

- Possibility of a high distribution of the information collection along the entire access perimeter (closest point to the user);

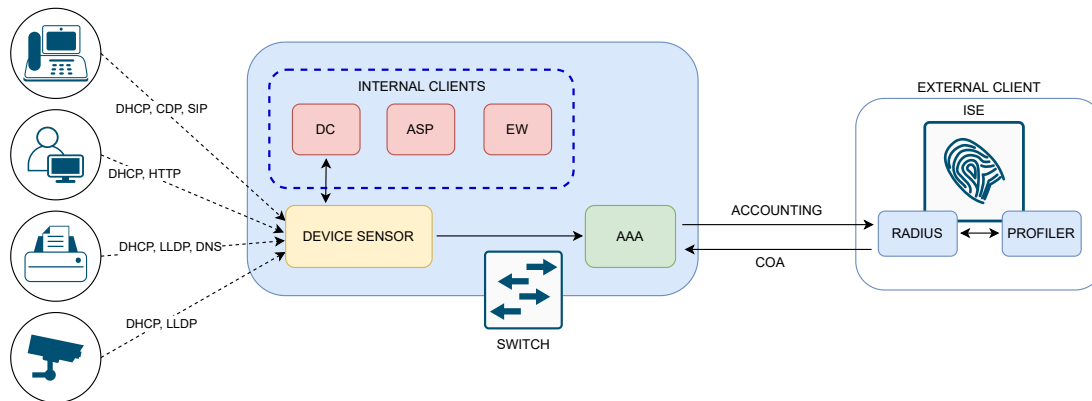


Figure 6.9: Device sensor architecture.

- Presence of pre-process and pre-analyses activities, which lighten the computational work of the ISE profiling module;
- By integrating the authentication and authorization of end devices via RADIUS, with the profiling functionality implemented with the same protocol, the efficiency of the entire device management process is increased. In fact, there will be no duplication processes of packets to other third-party systems as it is all integrated within a single system (ISE).

To successfully implement this feature, protocols such as CDP, LLDP, and DHCP must be enabled and configured correctly. It is also necessary to configure which attributes are to be collected to send to ISE. This operation takes place by creating some lists, within which the attributes to be collected are specified. Finally, as reported in the current and previous sections, the RADIUS accounting function must be enabled. In the case of ISE, this translates into enabling the RADIUS probe (shown in Figure 6.8). Using the following configuration, the lists of attributes to be used with the various protocols are specified.

```
SwitchNAC(config)#device-sensor filter-list dhcp list DHCP_LIST
SwitchNAC(config-sensor-dhcplist)#option name host-name
SwitchNAC(config-sensor-dhcplist)#option name class-identifier
SwitchNAC(config-sensor-dhcplist)#option name client-identifier
SwitchNAC(config-sensor-dhcplist)#option name parameter-request-list
SwitchNAC(config-sensor-dhcplist)#exit
SwitchNAC(config)#device-sensor filter-list cdp list CDP_LIST
SwitchNAC(config-sensor-cdplist)#tlv name device-name
SwitchNAC(config-sensor-cdplist)#tlv name platform-type
SwitchNAC(config-sensor-cdplist)#tlv name version-type
SwitchNAC(config-sensor-cdplist)#tlv name capabilities-type
SwitchNAC(config-sensor-cdplist)#tlv name address-type
SwitchNAC(config-sensor-cdplist)#exit
SwitchNAC(config)#device-sensor filter-list lldp list LLDP_LIST
SwitchNAC(config-sensor-lldplist)#tlv name port-id
SwitchNAC(config-sensor-lldplist)#tlv name system-name
```



```
SwitchNAC(config-sensor-lldplist)#tlv name system-capabilities
SwitchNAC(config-sensor-lldplist)#tlv name system-description
```

Two of the noteworthy parameters used to profile via the CDP/LLDP lists are *version-type* and *system-description*. These contain the firmware version of the software installed within the devices. It is therefore possible to create policy sets that take into account the devices OS version. In fact, if the firmware could not be updated in any way, and the current version should cause difficulties in deploying the system, it is possible to circumvent them by using ad hoc policies. Obviously, as suggested by best practices, it is good to use more and more recent firmware versions. After creating the lists, network administrator must first associate them with the device sensor and then configure the RADIUS accounting packets to transport them.

```
SwitchNAC(config)#device-sensor filter-spec dhcp include list DHCP_LIST
SwitchNAC(config)#device-sensor filter-spec lldp include list CDP_LIST
SwitchNAC(config)#device-sensor filter-spec cdp include list LLDP_LIST
SwitchNAC(config)#device-sensor accounting
SwitchNAC(config)#device-sensor notify all-changes
```

Last note, as previously mentioned, the device must support and implement the protocols used. To enable them, the following commands must be used.

```
SwitchNAC(config)#cdp run
SwitchNAC(config-if)#cdp enable
SwitchNAC(config)#lldp run
SwitchNAC(config-if)#lldp receive
SwitchNAC(config-if)#lldp transmit
```

In addition to the configurations presented above, in order for the DHCP information to be captured by the device sensors, it is necessary to enable the DHCP snooping functionality. In fact, together with the device-tracking functionality, it allows the switch to create an internal association between MAC and IP addresses. However, some issues with this functionality occurred during the testing phase. In fact, by default, when DHCP snooping is enabled, a special option (n. 82) is added within the DHCP packets. This option causes the switch to drop packets in this particular configuration. This additional option was therefore deleted using the following command.

```
SwitchNAC(config)#no ip dhcp snooping information option
```

Within the ISE GUI it is possible to check the values of the parameters collected through the device sensors under the “*Endpoints*” tab. An example is shown in Figure 6.10. Note that the parameters correspond to those present in the configuration shown above.

Before proceeding with the profiling methods, below is reported the problem

cdpCacheCapabilities	H,P
cdpCacheDeviceId	T40G805EC0
cdpCachePlatform	T40G
dhcp-class-identifier	yealink
dhcp-client-identifier	01:80:5e:c0:
dhcp-parameter-request-list	1, 2, 3, 4, 6, 7, 12, 15, 28, 42, 66, 67, 43, 100, 101, 120
host-name	SIP-T40G
ip	172.
lldpCacheCapabilities	B;T
lldpCapabilitiesMapSupported	B;T
lldpSystemDescription	76.84.0.15
lldpSystemName	SIP-T40G

Figure 6.10: Endpoint attributes collected.

and a possible resolution related to the addressing of *Polycom* IP phones. Previously, filter lists were defined with which the switch collects information to be sent to ISE for profiling. However, after some analyses it was discovered how the CDP protocol can be the source of problems with *Polycom* devices. If the CDP protocol is enabled in both 2 nodes (i.e., either within switch interface and within the IP phone), the information relating to the Voice VLAN is incorrectly communicated. Instead of including the one defined by ISE, switch inserts the one defined statically within the interface (i.e., VLAN 201) into the CDP packet. This causes the IP phone to remain blocked within the Welcome VLAN, obviously failing to contact the DHCP server. In Figure 6.11 and Figure 6.12 the contents of the incorrect and correct CDP packets are shown respectively. Notice how the *VLAN ID* field is different.

```

v Telecommunications Industry Association TR-41 Committee - Network Policy
 1111 111. .... = TLV Type: Organization Specific (127)
 .... ..0 0000 1000 = TLV Length: 8
 Organization Unique Code: 00:12:bb (Telecommunications In
 Media Subtype: Network Policy (0x02)
 Application Type: Voice (1)
 0... .. = Policy: Defined
 .1.. .. = Tagged: Yes
 ...0 0001 1001 001. .... = VLAN Id: 201
 .... ..1 01.. .... = L2 Priority: 5
 .... ..10 1110 = DSCP Priority: 46

```

Figure 6.11: Wrong packet content.

One way to solve this problem is to disable the CDP at the interface level. This operation must therefore be carried out on all ports in which this type of de-

```

  ▾ Telecommunications Industry Association TR-41 Committee - Network Policy
    1111 111. .... = TLV Type: Organization Specific (127)
    .... 0000 1000 = TLV Length: 8
    Organization Unique Code: 00:12:bb (Telecommunications In
    Media Subtype: Network Policy (0x02)
    Application Type: Voice (1)
    0... .. = Policy: Defined
    .1.. .. = Tagged: Yes
    ...0 0101 1000 011. .... = VLAN Id: 707
    .... .. 01.. = L2 Priority: 5
    .... .. ..10 1110 = DSCP Priority: 46
  
```

Figure 6.12: Correct packet content.

vices are connected. Another solution would result in disabling the CDP globally. This would cause the use of only LLDP (open standard), making the previously defined CDP filter useless. In particular, the protocol currently implemented is an extension of the LLDP, that is the LLDP Media Endpoint Discovery (**LLDP-MED**). Initially presented by Telecommunications Industry Association (TIA) TR-41.4 subcommittee, it introduces new features needed for managing VoIP devices. However, both solutions presented do not represent a feasible solution in terms of management and scalability. Another way to solve this issue (and the one actually used in the final deployment phase is reported at Section 6.4.1). As a last note, it is important to point out that through the use of device sensors, the flow described in Figure 3.1 is slightly different. In fact, the device-sensors exploit the RADIUS Accounting-Request messages, which are exchanged only after authentication. As mentioned in Section 3.1, the figure shows only the general flow schema.

### 6.2.2 DHCP

The DHCP probe is one of the most used among all. As its name indicates, this takes advantage of the information contained within DHCP messages.

This probe can collect attributes using one (or both) of the following modes:

- DHCP probe;
- DHCP SPAN probe.

It is however possible to use the device sensors, and therefore the RADIUS protocol, to collect information for profiling. Here are reported some attributes that can be collected through DHCP probes.

DHCP attributes	Option	Description
dhcp-class-identifier	60	Used to identify the type and configuration of a DHCP client. An example of a possible value is “ <i>MSFT 5.0</i> ”, used by <i>Windows</i> workstations.

dhcp-client-identifier	61	Provides the MAC address of the machine.
dhcp-message-type	53	Indicates the type of DHCP message. These are 4 and are divided into: <ul style="list-style-type: none"> <li>• DISCOVER;</li> <li>• OFFER;</li> <li>• REQUEST;</li> <li>• ACK.</li> </ul>
dhcp-parameter-request-list	55	Used to request values for specified configuration parameters. A <i>Windows 10</i> workstation uses the following exact sequence: 1, 3, 6, 15, 31, 33, 43, 44, 46, 47, 121, 249, 252.
dhcp-requested-address	50	IP address requested by the endpoint. Combined with the <i>dhcp-client-identifier</i> , it provides a binding between IP and MAC address.
host-name	12	Provides the name of the machine.
domain-name	15	Domain name used by the client to resolve host names via the DNS.
client-fqdn	81	Provides the Full Qualified Domain Name ( <b>FQDN</b> ) of the machine. In this way ISE server can update the IP-address-to-FQDN binding.

Table 6.2: Main RADIUS attributes.

## DHCP Probe

The following probe requires that the request made by the end device for an IP address be redirected to the ISE PSN. This is possible through the DHCP relay functionality. The layer 3 next hop with respect to the terminal device, in addition to sending the DHCP discover message to the correct DHCP server, will also send it to the PSN. By doing so, the device receives the IP address from the server, while the PSN can process the received message (without replying) and obtain useful information. With ISE it is possible to set multiple targets of this request so that it is received by multiple nodes. However, due to the traffic load, it is recommended to use only one node as a recipient. In the event that a certain redundancy is to be guaranteed, one of the following 3 solutions can be implemented:

- Load Balancers: the load balance devices allow to use a single Virtual IP (**VIP**) with which the different nodes can respond to requests. Multiple devices therefore use the same IP address. It is up to the load-balancer to

manage and route these requests to the servers for which it acts. Obviously, this consideration is valid when there are more active PSNs. In a standalone deployment, using a balancer would be a waste of resources and make no sense. In the context of IC, there are currently balancers implemented. However, they are not exploited both because they are still in the migration phase and because the load of requests is not yet so consistent;

- Anycast: this is special address which can be associated with multiple devices. The messages are delivered to the single device based on the routing operations carried out for that specific packet;
- Device Sensor: DHCP information is collected locally by the NAD which, through RADIUS accounting packets, are subsequently sent to the NAC server.

When using the DHCP probe it is important to know that the use of the relay functionality is not the only possible one. In fact, there is another solution, which is that of using a DHCP proxy. Configuring ISE as the first server, and the DHCP server as the second, leads to the same result. However, doing so, it slows down the addressing process. In fact, through a proxy, ISE would represent the primary address, while the others would represent the fallback. A timeout must therefore expire before delivering the request to the correct server.

### DHCP SPAN Probe

With this method, unlike the previous one, the traffic to the ISE interface is duplicated. Through the use of Switch Port Analyzer configurations (**SPAN**) and Remote SPAN (**RSPAN**), DHCP packets will also be delivered to the ISE PSN. However, as can be noticed, with the DHCP SPAN probe there is a not negligible increase in terms of traffic load (considering that requests can come from a large number of clients). It is therefore advisable to choose the first of the DHCP methods described and never both (this because doing so does not bring any improvement).

### 6.2.3 HTTP

As can be guessed, through this probe, HTTP messages are used to gather information. When an HTTP request is made, it contains some information such as software vendor, operating system and application type. This information is contained within the *User-Agent* field. As for the previous one, the HTTP probe can be divided into:

- URL Redirection;
- HTTP Probe with Direct Portal Access;
- HTTP SPAN Probe.

However, even in this case it is possible to use the features provided by the device sensors.

### URL Redirection

One of the most important functions of ISE is to manage access from guest users. In these cases, for authentication, web portals are created in which users can register. The user, then trying to access the network, is redirected to these web pages. In this way, since the traffic is redirected to ISE, the latter is able to inspect the *User-Agent* attribute and receive the desired information. Note how the redirect function can be configured in NADs. As discussed at the beginning, the HTTP probe is still active by default, and therefore in the case of redirects it is not necessary to explicitly activate the relative probe.

### HTTP Probe with Direct Portal Access

Unlike the previous one, the following probe allows network administrator to capture the traffic in these portals not involved in the redirection operation (Sponsor or My Devices). To establish an ISE association, the sender's IP line link with his MAC address. In ISE hosted web services it is important to know that encrypted traffic (HTTPS) can be decrypted.

### HTTP SPAN Probe

In the event that the web traffic is not addressed to ISE, it is necessary to use mirror functions to duplicate the traffic to an interface of a PSN. In such scenario, ISE is therefore a transparent node. This method is to be used only when the others (including the one related to the device sensor) are not available. While ISE-hosted traffic uses ports 80 (or 8080) to accept incoming requests, port 443 (HTTPS) is not used in this scenario as the traffic is encrypted. In terms of traffic management, the recommended methods to be used are those relating to device sensors and URL redirection. While no particular configuration is required for the traffic monitoring modes via web portals, when using the SPAN probe it is necessary to ensure that the ISE interface is active. From the ISE CLI it is possible to configure and activate an interface by means of a certain sequence of commands.

## 6.2.4 SNMP

The SNMP protocol allows the configuration, management and monitoring of network devices. SNMP is a layer 7 protocol and bases its operation on the UDP transport protocol (UDP port 161/162). The architecture necessary to collect the information from the systems can be broken down into 4 components:

- Manager: entity to which monitoring services are delegated. With NAC, this role is assumed by the authentication server (ISE);

- **Agent:** software modules installed within the network devices. They have the task of extracting information from the devices and saving them in a special internal database (MIB);
- **Management Information Base (MIB):** internal device database organized according to a certain hierarchy (not totally common among different vendors) in which all the information about the specific device are saved;
- **Management protocol:** protocol that allows the manager to obtain information from the MIBs through a *GET* command, and the agents to report the presence of particular events through a *TRAP* message.

There are different versions of SNMP (SNMPv1, SNMPv2 and SNMPv3). Of these versions, there are also more sub-versions that modify some implementation aspects. The 2 main differences to underline between these 3 versions are the increasing availability of greater types of messages (and therefore more available operations) and the increase of the security level (from a simple community string to the implementation of a more advanced authentication system). To take advantage of the management protocol, there are different types of messages through which one or more information is acquired or initialized in a different way. In particular, as regards the implementation of probes, 2 of these are used:

- **GET Request:** message used by the manager (ISE) to obtain information about the agent installed on the device. The corresponding response message is GET Response;
- **TRAP Message:** this type of message is generally configured and used to report environmental events or errors identified by the agent. Unlike the previous message, this is not sent directly by the agent without an explicit request being made. In the case study, it can be used when the NAD obtains new information with respect to the final device and communicates it to ISE.

The information to be obtained is, as previously mentioned, stored in the MIB (shown in Figure 6.13). Each specific parameter (object) is identified by an Object Identifier (**OID**). The OID corresponds to a sequence of numbers that trace the hierarchical tree shape shown in Figure 6.13. An example of an OID is *1.3.6.1.4*.

Figure 6.14 shows the interface with which the SNMP parameters are configured. Note that according to the version chosen (SNMPv1, SNMPv2c, SNMPv3), it is possible to configure only some parameters. In fact, if the first version is used (as in the figure), it is possible to specify only the password (SNMP Read-Only Community) as the authentication method.

After having presented a brief introduction of the protocol, the 2 types of probes that can be implemented in ISE are illustrated below. However, it is important to remember that, if possible, it is better to prefer the implementation of data collection through device sensors. One of the reasons behind this choice is the excessive traffic load generated by the SNMP protocol.

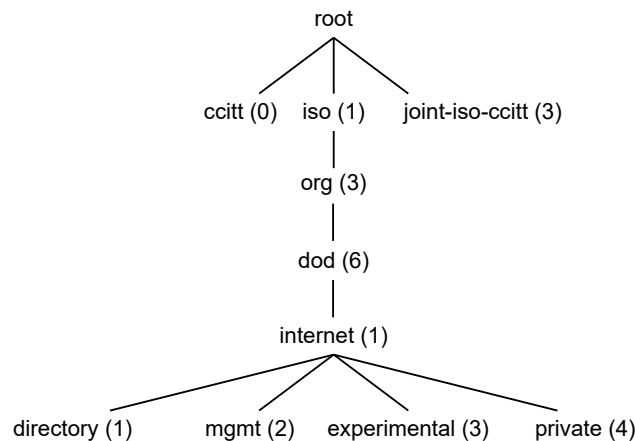


Figure 6.13: General SNMP MIB tree structure.

The image shows a configuration interface for SNMP settings. It includes several sections with expandable/collapsible headers:

- ▶ RADIUS Authentication Settings
- ▶ TACACS Authentication Settings
- ▼ SNMP Settings

Under the 'SNMP Settings' section, the following fields are visible:

- \* SNMP Version:
- \* SNMP RO Community:
- SNMP Username:
- Security Level:
- Auth Protocol:
- Auth Password:
- Privacy Protocol:
- Privacy Password:
- \* Polling Interval:  seconds (Valid Range 600 to 86400 or zero)
- Link Trap Query:
- MAC Trap Query:
- \* Originating Policy Services Node:

Figure 6.14: ISE SNMP probe settings.


## SNMPTRAP

The SNMPTRAP probe receives from the appropriately configured network device, information relating for example to the status of a general link (i.e., link-up or link-down) or relating to the MAC address table. This probe is closely related to the SNMPQUERY probe. For example, in case it is necessary to make a request through the SNMPQUERY probe, the SNMPTRAP allows this operation. It is therefore possible to confirm that, in order to have a correct operation of the SNMPTRAP probe, it is necessary to enable the SNMPQUERY one.

To implement this type of probe it is necessary to configure both the NAD and the authentication server. Therefore, starting from the configuration in ISE,



as a first step it is necessary to activate the probe from the GUI. As shown in Figure 6.15 it is possible to specify which UDP port to use (162 is the default), on which of the available interfaces to receive the probes (it is possible to select them all) and the active services (Link and MAC Trap query). This operation must be repeated for each PSN implemented. Once the service is activated, it is



**SNMPTRAP**

Link Trap Query

MAC Trap Query

Interface: GigabitEthernet 0

Port: 162

Description: This probe receives Linkup, Linkdown and MAC notification traps from network devices.

Figure 6.15: ISE SNMPTRAP probe settings.

necessary to configure the object that represents the NAD within ISE. Therefore, once chosen which device(s) it is necessary to configure, the correct configuration must be carried out using the fields shown in Figure 6.14. Once this is done, the single NAD must be configured directly from the CLI. The entire command set is shown below.

```

SwitchNAC(config)#interface GigabitEthernet 1/0/1
SwitchNAC(config-if)#snmp trap mac-notification change added
SwitchNAC(config-if)#snmp trap mac-notification change removed
SwitchNAC(config-if)#exit
SwitchNAC(config)#mac address-table notification change
SwitchNAC(config)#mac address-table notification mac-move
SwitchNAC(config)#snmp-server trap-source <Source-Interface>
SwitchNAC(config)#snmp-server enable traps snmp linkdown linkup
SwitchNAC(config)#snmp-server enable traps mac-notification change move
SwitchNAC(config)#snmp-server host <ISE-PSN> version 2c <RO-com-string>

```

As can be seen from the list of commands, in addition to the configuration relating to the role of ISE manager, the information to be sent to ISE has been configured. Indeed let's consider the scenario where only SNMPTRAP probes (Link Trap Query) are used. In this case, it would not be possible to determine who is connecting to the switch interface, as only information related to the link is reported. Therefore, the further use of MAC Trap Queries allows to trace the single device. However, to obtain more specific and detailed information it is necessary to use SNMPQUERY probes.

## SNMPQUERY

SNMPQUERY probes represents a very useful and effective tool for obtaining detailed information of end devices. These differ in 2 types:

- System: the queries are performed periodically, following the configuration shown in Figure 6.14. The default value is 28,800 seconds (8 hours);
- Interface: request that occurs in response to an SNMPTRAP query (requires SNMPTRAP probe enabled) or to a RADIUS Accounting-Request-Start packet (requires RADIUS probe enabled).

These 2 sub-typologies collect information of a different nature. The first collects information from the NAD MIB, while the second from the particular interface that triggered the alert. An example of information objects obtained from the first are *cdpCacheEntry*, *lldpLocalSystemData* and *lldpRemoteSystemsData*. Among these 3, the second one collects information relating to:

- Interface data;
- Port and VLAN data;
- Session Data (if Ethernet interface);
- CDP data (Cisco devices);
- LLDP data.

The last two protocols are widely implemented in IoT contexts, multimedia devices and also within operating systems. Thanks to these, it is possible to obtain a very high number of information (obviously if the end device supports them). The configuration in this scenario is similar to the previous one. In fact, after enabling the probe (shown in Figure 6.16) together with the SNMPTRAP probe, it is necessary to configure only one command within the NAD:

```
SwitchNAC(config)#snmp-server community <RO-com-string> RO
```

The command just reported allows the NAD to accept SNMP RO requests from the PSN. In case the NAD had been configured with a different SNMP version (e.g., SNMPv3), it would have been necessary to use other advanced commands. From Figure 6.16 it is possible to see how the network administrator can specify some parameters such as:

- Timeout (milliseconds): specifies the total time to wait for an SNMP response;
- Retries: in the event that the establishment of an SNMP session with an NAD should fail, ISE can retry a number of times. This number is specified in this field;

▼ SNMPQUERY

Retries:

Timeout:

EventTimeout:

Description:

Figure 6.16: ISE SNMPQUERY probe settings.

- EventTimeout (seconds): specifies the total time to wait before responding to a RADIUS Accounting-Request-Start or an SNMPTRAP message.

As reported in the previous section, regarding the configuration of the NAD within ISE, it is possible to configure the Polling Interval (Figure 6.14). This allows network administrator to specify how often ISE should send a *GET* Request to the NAD. This parameter is obviously valid only for System SNMPQUERY probes.

## 6.3 Profiling policies

As for the authentication and authorization process, the use of a policy set is also envisaged for the profiling function. In particular, being related to user profiling, this is called profiling policy set. Once the information has been obtained from one or more probes, ISE has the task of analyzing them and estimating the type of device. Each profiling rule consists of 2 main parts:

- Set of attributes that are compared with assigned values (either by default or by the administrator). A certain (editable) score called Certainty Factor (CF) is assigned to each comparison. If the value of the attribute taken into consideration equals the CF, the total score would be increased with the CF assigned;
- A minimum total score value (minimum CF), with which a certain device is profiled. In the event that, after comparing the attribute sent by the NAD with the configured ones, the total sum of the CF exceeds the minimum threshold then the device would be assigned the current profile.

Figure 6.17 shows an example of profiling policy (present by default in ISE) with which Apple devices are profiled. It is possible to see that, one of the conditions to be satisfied is given by the *device-platform* attribute which must be equal to “*apple-ios*”. The minimum certainty factor is set to 10 and the CoA configuration is set to global settings (so it is done when configured in the general settings). For a given profile it is therefore possible to specify the CoA action to be carried out.

**Profiler Policy**

\* Name:  Description:

Policy Enabled

\* Minimum Certainty Factor:  (Valid Range 1 to 65535)

\* Exception Action:

\* Network Scan (NMAP) Action:

Create an Identity Group for the policy:  Yes, create matching Identity Group  
 No, use existing Identity Group hierarchy

Parent Policy: \*\*\*NONE\*\*\*

\* Associated CoA Type:

System Type: Cisco Provided

Rules

Condition	Expression
If Condition: <input type="text" value="Apple-Device-Rule2-Check1"/>	ACIDEX:device-platform EQUALS apple-ios
If Condition: <input type="text" value="Apple-DeviceRule1Check1"/>	
If Condition: <input type="text" value="Apple-DeviceRule1-SCAN"/>	

Conditions Details (pop-up):

Name: Apple-Device-Rule2-Check1  
Description: Condition for Apple-Device based on ACIDEX:device-platform  
Expression: ACIDEX:device-platform EQUALS apple-ios

Buttons: Save, Reset

Figure 6.17: Example of profiling policy.

For example, it is possible to set the re-authentication action in case the device is assigned to this profile. In this way, in the event that there are conditions based on profiling within the authorization policies, the device would obtain a more specific authorization profile. Within the profiling rules, various comparison tools are available to compare the values received with the configured ones. Some of these are “*EQUALS*”, “*CONTAINS*”, “*MATCH*”. While they may appear to be equivalent, they are distinguished by some subtle differences. Indeed, the first can be used for straight forward comparison, the second for multi-value attributes, and the third for regular expression comparison. An important note to do concerns the hierarchical structure of the profiles. In fact, it is possible to configure child profiles of other profiles. For example, the “*Apple-Device*” profile assumes the role of parent to other devices such as “*Apple-iPhone*”, “*Apple-iPad*” and so on. This more accurate profiling allow to increase the level of general granularity. Obviously, as the level of detail increases more and more, it will be necessary to use increasingly specific values for attributes. With the evolution of ISE, new needs have arisen that provide for the possibility of grouping multiple devices (from different vendors but of the same type) within a single group. In fact, let’s consider the scenario in which profiling is used within the authorization policies to assign authorization profiles. It would be highly useless to insert a condition for each type of printer (e.g., differentiating them by model). The solution to this problem is the logical profiles concept. These can be considered as logical containers that group devices of the same type. The administrator is therefore able to create and manage (in case there are new unregistered devices) these containers.

## 6.4 Profiling service set-up

The configurations and related explanations related to the profiling service are shown below.

### 6.4.1 ISE set-up

In order to better integrate the profiling service, changes have been made to the configurations in all components compared to those presented in the previous chapters. To better understand the processes created below, two different sections are shown: the first, Section 6.4.1, which lists the policy sets containing conditions relating to profiling, while Section 6.4.1 shows the fundamental components for defining a profile.

#### Policy sets

Unlike what was shown in the previous chapters, configuring the profiling function revealed the criticality of the dynamic assignment of the Voice VLAN. If previously it was possible to disable the CDP to allow correct operation, now it is no longer possible. In fact, having to capture the information of the connected devices, it is necessary to use all the available resources. Eliminating the CDP would therefore result in a loss of tools that can be used to profile devices. A different implementation was therefore chosen with which, however, it is possible to achieve the same goal.

In this new architecture, the Voice VLAN is not downloaded from ISE to the NAD. In fact, it is configured within the switch, and therefore in a static way. Once an IP phone connects to the interface, it is immediately associated with it. So where does the authorization to exchange traffic comes from? Through ISE, it is possible to use a parameter called “*Voice Domain Permission*” through which permission is granted to generate voice traffic to the device. When the newly connected device is assigned the isolation profile, it is not allowed to communicate with any device. Only when it is associated with the “*IP-Phones*” logical profile it will be able to communicate correctly. The authorization profile it is associated with once the device is authenticated has been defined as “*IP\_Phone\_Cisco\_Domain\_Permission*”. Figure 6.18 and Figure 6.19 shows the Authorization Policies for 802.1X and MAB respectively.

For 802.1X, before a device can actually authenticate itself, it must be profiled correctly. In fact, it is not enough to provide the correct credentials, but the user must connect with a device appropriate to the authorization profile. As regards the MAB, in addition to the authorization profiles targeting devices, there is once again the policy associated with external contractors. Finally, the last policy, if matched, attributes the isolated authorization profile. This way, a device whose profile is unknown or whose identity has not been registered will have limited access. In the first phase of testing, the only guaranteed access is to the DHCP server with which an IP address can be obtained.

Authorization Policy (7)				Results		Hits	Actions
Status	Rule Name	Conditions	Profiles	Security Groups			
⊕	Search						
⊕	Employee	AND <ul style="list-style-type: none"> <li>OR               <ul style="list-style-type: none"> <li>Network Access UserName CONTAINS host</li> <li>AD_IC ExternalGroups EQUALS ic:extra.infocamera.it@SEC0GruppiPrivacy/NAVIGAZIONE-IC</li> </ul> </li> <li>Cisco_Authn_Passed</li> </ul>	Employee_Cisco	Select from list	624	⚙️	
⊕	VoIP	AND <ul style="list-style-type: none"> <li>Network Access UserName EQUALS ic</li> <li>Cisco_Authn_Passed</li> <li>EndPoint LogicalProfile EQUALS IP-Phones</li> </ul>	IP_Phone_Cisco_Domain_Permit	Select from list	19536	⚙️	
⊕	Printer	AND <ul style="list-style-type: none"> <li>Network Access NetworkDeviceName EQUALS ic</li> <li>Cisco_Authn_Passed</li> </ul>	Printer_Cisco	Select from list	0	⚙️	
⊕	IP-Camera	AND <ul style="list-style-type: none"> <li>Network Access NetworkDeviceName EQUALS ic</li> <li>Cisco_Authn_Passed</li> <li>EndPoint LogicalProfile EQUALS Cameras</li> </ul>	IP_Camera_Cisco	Select from list	0	⚙️	
⊕	Multimedia	AND <ul style="list-style-type: none"> <li>OR               <ul style="list-style-type: none"> <li>Network Access UserName EQUALS ic</li> <li>Network Access UserName EQUALS ic</li> </ul> </li> <li>Cisco_Authn_Passed</li> <li>EndPoint LogicalProfile EQUALS Mobile Devices</li> </ul>	Multimedia_Cisco	Select from list	0	⚙️	

Figure 6.18: 802.1X profiling policy set.

Authorization Policy (6)				Results		Hits	Actions
Status	Rule Name	Conditions	Profiles	Security Groups			
⊕	Search						
⊕	IP-Phone-Profiling	EndPoint LogicalProfile EQUALS IP-Phones	IP_Phone_Cisco_Domain_Permit	Select from list	238	⚙️	
⊕	Printer-Profiling	EndPoint LogicalProfile EQUALS Printers	Printer_Cisco	Select from list	0	⚙️	
⊕	IP-Camera-Profiling	EndPoint LogicalProfile EQUALS Cameras	IP_Camera_Cisco	Select from list	0	⚙️	
⊕	Multimedia-Profiling	EndPoint LogicalProfile EQUALS Mobile Devices	Multimedia_Cisco	Select from list	0	⚙️	
⊕	Contractor	AND <ul style="list-style-type: none"> <li>InternalUser IdentityGroup EQUALS User Identity Groups GuestType_Contractor (default)</li> <li>Cisco_Authn_Passed</li> </ul>	Contractor_Cisco	Select from list	443	⚙️	
⊕	Default		Isolated_Cisco	Select from list	424	⚙️	

Figure 6.19: MAB profiling policy set.

## Profiling policy sets

For demonstration purposes, the *Yealink T40G* IP phone is used as an example device. However, the same explanation applies to other types of devices such as IP cameras or printers. Two firmware versions were used during the test phases, namely *76.84.0.15* and *76.84.0.140*. Both versions support both CDP and LLDP. Depending on the type of device and software version, these 2 protocols may not be active by default. Together with the VoIP technician it was verified that for both versions, both the protocols were enabled by default. Furthermore, always considering the *Yealink* brand, a change has been made to the switchboard provisioning file, explicitly enabling the CDP. That said, in Figure 6.20 the profiling policy set for the model taken into consideration is shown. As can be seen from Figure 6.20, in the initial phase of creating a profile the minimum CF is specified. The choice of any profiling policy from which it derives is then proceeded. Since the *T40G* is a *Yealink* device, it is the child of the profiling policy called “*Yealink-Device*”. The profiling policy sets are made up of conditions that are defined separately. In Figure 6.21 are reported the conditions used to profile *Yealink* devices. Note how, to profile a device model in detail, it

Profiler Policy List > Yealink-Device

**Profiler Policy**

\* Name:  Description:

Policy Enabled:

\* Minimum Certainty Factor:  (Valid Range 1 to 65535)

\* Exception Action:

\* Network Scan (NMAP) Action:

Create an Identity Group for the policy:  Yes, create matching Identity Group  
 No, use existing Identity Group hierarchy

Parent Policy: \*\*\*NONE\*\*\*

\* Associated CoA Type:

System Type: Administrator Created

**Rules**

If Condition:  Then:

If Condition:  Then:

Figure 6.20: Yealink device profiling policy set.

is necessary to use more stringent specifications. In fact, to classify the *T40G* model, it was necessary to specify the presence of particular strings for the CDP and LLDP attributes. During the testing phase, it was decided to assign a fair CF for all conditions. The last option of remarkable importance is the CoA type.

Yealink-Device-DHCP-Check3	Administrator Created	dhcp-class-identifier EQUALS yealink
Yealink-Device-LLDP-Check1	Administrator Created	lldpPortId STARTSWITH 80:5e:c0
Yealink-Device-LLDP-Check6	Administrator Created	lldpPortId STARTSWITH 00:15:65
Yealink-Device-LLDP-Check7	Administrator Created	lldpPortId STARTSWITH 80:5e:0c
Yealink-Device-MAC-Check2	Administrator Created	MACAddress STARTSWITH 80:5e:c0
Yealink-Device-MAC-Check4	Administrator Created	MACAddress STARTSWITH 80:5e:0c
Yealink-Device-MAC-Check5	Administrator Created	MACAddress STARTSWITH 00:15:65
Yealink-IP-Phone-CDP-Check1	Administrator Created	cdpCacheCapabilities CONTAINS P
Yealink-IP-Phone-LLDP-Check2	Administrator Created	lldpCacheCapabilities CONTAINS T
Yealink-T23G-CDP-Check1	Administrator Created	cdpCachePlatform EQUALS T23G
Yealink-T23G-CDP-Check2	Administrator Created	cdpCacheDeviceId STARTSWITH T23G
Yealink-T23G-LLDP-Check3	Administrator Created	lldpSystemName EQUALS SIP-T23G
Yealink-T40G-CDP-Check1	Administrator Created	cdpCachePlatform EQUALS T40G
Yealink-T40G-CDP-Check2	Administrator Created	cdpCacheDeviceId STARTSWITH T40G
Yealink-T40G-LLDP-Check3	Administrator Created	lldpSystemName EQUALS SIP-T40G

Figure 6.21: Yealink device profiling conditions.

With this option can be specified which operation to perform once the device is profiled correctly. It is important to remember, however, that the CoA action does not take place systematically. In fact, it is possible that profiling takes place at the same time as authentication. For example, in Figure 6.22 the authentication (and profiling) process of the *Yealink T40G* is shown. To perform this test, the IP phone has been initialized. In this way, the deployment process in



a production environment was simulated. The process extends from the bottom up. The first entry identifies the first access attempt. Having no user configured, the NAD uses the MAB to authenticate it. Already during this first phase it is possible to see how the device has already been identified. As previously stated, a CoA action is not necessarily required. If the device had not been profiled, it would have been associated with an isolation profile. Since the device is already profiled, it is already assigned the authorization profile with which it will be operative. Remember how the entries containing only one component indicate the download of the ACLs relating to the associated authorization profile. Once the IP address is obtained, the IP phone is able to contact the TFTP server and download the provisioning file. From here, it reboots, and then attempts to authenticate itself with the 802.1X credential (sixth entry) which are contained within the provisioning file.

✓		#ACSACL#-IP-IP-...					
✓		ic	80:5E:C0	Yealink-T40G	802.1X_Wired >> VoIP	IP_Phone_Cisco_Domain_Permission	
✓		#ACSACL#-IP-IP-...					
✓		80:5E:C0	80:5E:C0	Yealink-T40G	MAB_Wired >> IP-Phone_profiling	IP_Phone_Cisco_Domain_Permission	
✓		#ACSACL#-IP-IP-...					
✓		80:5E:C0	80:5E:C0	Yealink-T40G	MAB_Wired >> IP-Phone_profiling	IP_Phone_Cisco_Domain_Permission	172

Figure 6.22: Yealink device profiling flow.

## 6.4.2 NAD set-up

As mentioned previously, some changes have been made within the interface level configuration. The interface configuration is reported below.

```

switchport access vlan 200
switchport mode access
switchport voice vlan 707
ip access-group WELCOMEACL in
device-tracking attach-policy DeviceTrackingPolicy
authentication event fail retry 3 action next-method
authentication host-mode multi-auth
authentication order dot1x mab
authentication priority dot1x mab
authentication port-control auto
authentication open
authentication periodic
authentication timer reauthenticate server
authentication timer restart 10
authentication timer inactivity server
mab
dot1x pae authenticator
dot1x timeout quiet-period 18
dot1x timeout tx-period 1
dot1x max-reauth-req 3

```



```
spanning-tree portfast
```

On the other hand, the configurations relating to profiling inserted within the NAD have been reported in Section 6.2.1 with the relative descriptions. However, it is important to pay attention to 2 previously not described details.

The first concerns the DHCP snooping functionality. To be able to activate it correctly it is necessary to use the following commands:

```
SwitchNAC(config)#ip dhcp snooping
SwitchNAC(config)#ip dhcp snooping vlan 200,501,601,707
SwitchNAC(config)#interface GigabitEthernet 1/0/48
SwitchNAC(config-if)#ip dhcp snooping trust
```

Remember how VLAN 501 is used to host external consultants. In addition to activating the snooping functionality on all the VLANs involved in the authentication process (even if the VLAN 200 has no impact on the routing functions), it is necessary to define which are the trust interfaces. The latter are interfaces in which the snooping feature checks are not carried out as they are considered trusted. The ports on which to apply this command are those of uplink, i.e., all those that point to the DHCP server. In this case, the *GigabitEthernet 1/0/48* interface represented the trunk interface to the production environment.

The second note concerns the configuration of the device-sensors. In the first instance, it was explained that the analysis of the data collected by these sensors is delegated to the central node (i.e., ISE). However it is also recommended within the NAD, to disable the internal analyzer. In order to do this, the following command has to be used.

```
SwitchNAC(config)#no macro auto monitor
```

## 6.5 Profiling related problem

During the analyses carried out, in addition to the problems relating to the Voice VLAN, which must be subject to particular attention, a problem related to profiling occurred with *Polycom* devices. When the IP phone is connected to the switch interface, it starts sending multicast LLDP packets. The switch, configured to use device-sensors, should collect the information specified within the “*LLDP\_LIST*” list. However, due to the abnormal behavior of the switch, only part of this information is sent to ISE. In this way the profiling function could not take place correctly. The current switch model which has been used is “*WS-C3650-48TD*”, while the initial version was the *16.5.1a*. After some troubleshooting steps it was seen how this obsolete version could be the cause of the problem. After having therefore performed an update to version *16.12.07*, the switch has returned to operate correctly. Figure 6.22 shows the process by which an IP phone, starting from the factory configuration, is authenticated and pro-

filed. To interpret the flow it is necessary to read the flow from the bottom up. In the first row the device, not yet recognized, is registered in the ISE database. It is then associated with the isolation profile (which assigns the device to VLAN 200). In fact, it acts as both a Welcome and Isolation VLAN. The next row refers to the ACL download. In fact, although the network is isolated, an ACL has been created in case it is expanded towards the other devices. The third row turns out to be of considerable importance. At this point the device is profiled and, thanks to this, in the fourth row it is associated with the authorization profile relating to the IP phones. At this point the second download of the ACL occurs. In this context, the device is able to contact the DHCP server and obtain an IP address. The NAD, by registering a new information about the device (i.e., the IP address) notifies it to ISE. Upon receiving new information, the latter sends a CoA message (for this reason the presence of the two authentication blocks on rows 4 and 7). The *Polycom* device, once obtained the IP address, then contacts the IC switchboard. After downloading the configuration file, it performs a reboot to apply the changes. At this point, having the 802.1X credentials configured, the authentication process shown in Figure 5.42 takes place.

#ACSACL#-IP-IP-...						
ic	64.16.7F:	Polycom-VVX201	802.1X_Wired >> Infocamere_AD_Internal	802.1X_Wired >> VoIP	IP_Phone_Cisco_Domain_Permission	
#ACSACL#-IP-IP-...						
64.16.7F:	64.16.7F:	Polycom-VVX201	MAB_Wired >> MAB_Wired	MAB_Wired >> IP-Phone-Profiling	IP_Phone_Cisco_Domain_Permission	172.
	64.16.7F:					
#ACSACL#-IP-IP-...						
64.16.7F:	64.16.7F:	Polycom-VVX201	MAB_Wired >> MAB_Wired	MAB_Wired >> IP-Phone-Profiling	IP_Phone_Cisco_Domain_Permission	
	64.16.7F:					
#ACSACL#-IP-Isol...						
64.16.7F:	64.16.7F:		MAB_Wired >> MAB_Wired	MAB_Wired >> Default	Isolated_Cisco	

Figure 6.23: AAA and profiling flow for Polycom-VVX201.

# Chapter 7

## Posture

In the previous chapters, the processes of authentication and profiling have been described. However, in both contexts, the problem relating to the state of health of the devices was never dealt with. As discussed in Section 3.1.4, a device affected by any kind of malicious entity has the potential to compromise the state of the network. To mitigate this type of scenario, the posture function has been introduced. Through a particular agent (whose type depends on the implemented posture mode) information is extracted from the endpoint and transmitted to ISE. At this point, the information obtained is subjected to policies (the same concept addressed for AAA and profiling). If the verified conditions were to report positive results, then the device would be defined *Compliant* and would gain access to the network. If it was not, ISE would be able to isolate the device by assigning it to a quarantine zone (better known as remediation zone). Once assigned to this network, the device has the opportunity (through several methods defined according to the implementation carried out) to resolve the errors that led it to not be compliant. Note that, from now on, the discussion made are based on *Windows* systems. In fact testing procedures and analyses are aimed at simulating the production environment. However, it is possible to carry out the exact same analyses with other OS environments such as *macOS*.

### 7.1 Posture architecture

The implementation of the posture through ISE is strictly related to the type of agent used. The types supported by the current version of ISE present in the IC production environment are listed below.

#### 7.1.1 Temporal Agent

In an enterprise environment, such as that of InfoCamere, to perform posture analyses it is possible to use mainly 2 methods. The first one bases its functioning on the concept of a temporal agent, which is an entity that during a limited period is used within the device to collect device information. This implementation is designed for figures present in the company for a limited period of time. This

method is also commonly referred to as “*dissolvable*” due to its characteristics. In fact, the software to proceed with the evaluation of the device is downloaded and executed within the user context (i.e., web browser). Once exited from the latter, the agent dissolves (i.e., it is no longer present within the device). A similar solution is strongly recommended in an environment where external figures (e.g., consultants) are usually quite present. This is because, taking into consideration the fact that external devices may not be accessible in privileged or admin mode, it is not possible to install applications/programs on devices not owned by the company. Therefore the flexibility that this functionality offers is extremely fundamental. However, it is important to underline what are the limiting aspects that this modality presents. In fact, by not relying on software installed on the device, the fullness of the features are not exploited. First of all, the conditions under which the assessments can be carried out are limited. This means that only some aspects of the device can be verified, thus leaving uncertainty about the others. An example of posture conditions includes:

- Anti-Malware installation;
- Firewall enabled;
- USB check.

The respective meaning, and that of the other components, is reported in the Section 7.2.1. As discussed earlier, posture functionality is closely related to the remediation one. It is therefore important to underline that, using temporal agents, remediation must be performed manually (ISE indeed doesn't support remediation for this posture mode). It is therefore not possible to trigger an automatic remediation process as it would with other types of agents. This behavior is to be considered in line with the operating model. Since no software can be installed to evaluate in detail the device, no action can be taken to update (make compliant) the device.

The second method, on the other hand, involves the download and installation of a software through which ISE dialogues to evaluate the posture conditions of the device. This scenario is therefore applicable for all those devices that belong to the environment in which NAC is being implemented. However, as described in Section 8.2.3, it is still possible to proceed with the installation during the device provisioning phase. Once in production, the posture process will only provide for the communication of the assessments, leaving out the process related to download and installation. This would imply a significant improvement in business productivity.

### 7.1.2 Agentless

Before moving on to the description of the next agent type, it is important to make a brief note on a new feature introduced in ISE. With the most recent releases (starting from releases 3.0 onwards), a new type of mode has been introduced: agentless. As can be deduced from the name, the assessment of the state

of the device with the conditions of postures is carried out without installing any software. The user experience therefore changes radically as the user is not aware of this process. The fact that there is no interaction between device and user is also due to the fact that there is no redirect to any portal. Once logged in, the user will gain access to the network after a short time (if judged compliant). So how does ISE communicate with the device? Communication takes place by establishing a session via SSH or PowerShell (TCP/UDP port 5985). One of the future works to pay attention to, is therefore the update to the new releases of ISE. The update would therefore introduce new features that can be used in multiple contexts. However, it is also necessary to consider the relative issues related to this operation: one among all the presence of bugs. In fact, each version introduces a non-negligible amount of bugs (most of these solved by the release of software patches) which can introduce compatibility issues with the previous releases.

### 7.1.3 AnyConnect Agent

In the event that the devices to be evaluated belong to the company, it is possible to install certain application software. Among these there is also one in particular, that is AnyConnect. Cisco *AnyConnect Secure Mobility Client*, also known as AnyConnect is a software application with which it is possible to establish Virtual Private Network (VPN) connections and perform posture functions by interacting with ISE. In the following scenario, requests from the device to the network would be intercepted by the switch and redirected to the ISE portal (from which AnyConnect can be downloaded). However, as mentioned above, it is possible that this software is already integrated within the provisioning configurations of PCs (e.g., via GPO). Another application provisioning option could be done through a software distribution method. The analyses carried out within the test environment involves the use of the AnyConnect agent.

In this mode, the application contains the device scan module which has the task of collecting all the information on the status of the device and sending it to the ISE PSN. In the event that the device fails verification, ISE allows it to obtain a limited period (configurable by ISE) of remediation. If the entity that sourced the issue is resolved, the device is defined as *Compliant*. At this point ISE assigns an authorization profile with which it is possible to access the network. Compared to the previous methods, using an application installed directly within the PC, the number of posture conditions available is higher. Some of these are:

- Application inventory;
- Disk encryption;
- Patch management.

It should be emphasized that with this mode, there is also the possibility of creating ad-hoc conditions. These types of conditions are useful in an environment where it is necessary to carry out specific controls. During the introduction it

was specified that only *Windows* machines are considered for the posture phase. Depending on the ISE release and the type of machine used, the number of available conditions is subject to change. For example, in case of *macOS* devices, the number of conditions is quite limited when compared to the *Windows* ones.

The remediation function mentioned above, in this mode, is made available in automatic. In fact, once the entities that do not make the device *Compliant* have been found, the posture scan module displays a screen where the user can start the automatic remediation. During this process AnyConnect is therefore in charge of all the operations to be performed, so that the device can be classified as *Compliant*. An example of AnyConnect agent is shown in Figure 7.1.

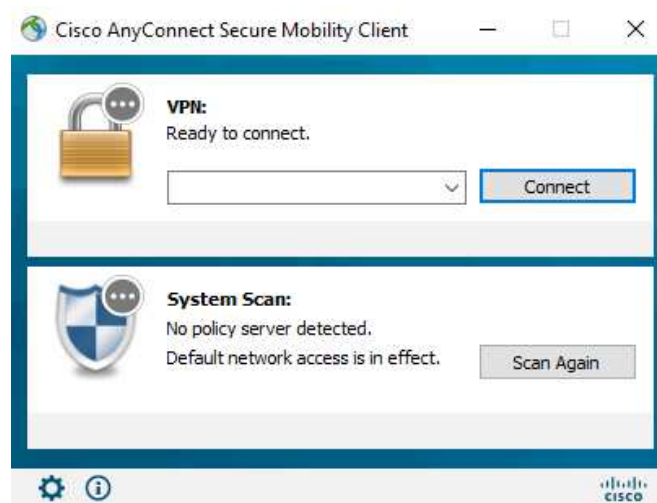


Figure 7.1: AnyConnect interface.

#### 7.1.4 Stealth AnyConnect Agent

The Stealth mode, both for the temporal agents and for AnyConnect, has the purpose of reducing the interaction between device and user. In this mode the user is not aware neither of the scan of postures nor of the communication between the end device and ISE. It is also possible, if AnyConnect has integrated only the module relating to posture, that its icon does not appear in the system tray. The application being installed behaves similarly to the non-stealth version. These equality can be summarized in 3 points:

- Admin credentials required for installation;
- The software upgrade does not require administrator-level credentials or permissions;
- Application executed within the User/Service space.

As in the previous mode, also in this case it is possible to modify the posture conditions by creating customized ones. The main difference in which it differs,

however, is in the modality of remediation. In this case, fully automatic remediation is not supported. To have a clear view of the whole posture functioning process, Figure 7.2 shows the steps to which the device is subjected. Figure 7.3 instead shows the posture flow in the event that a device would be compliant.

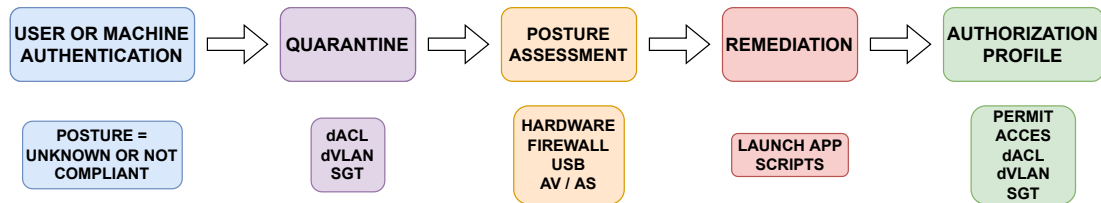


Figure 7.2: Generic posture assessment flow.

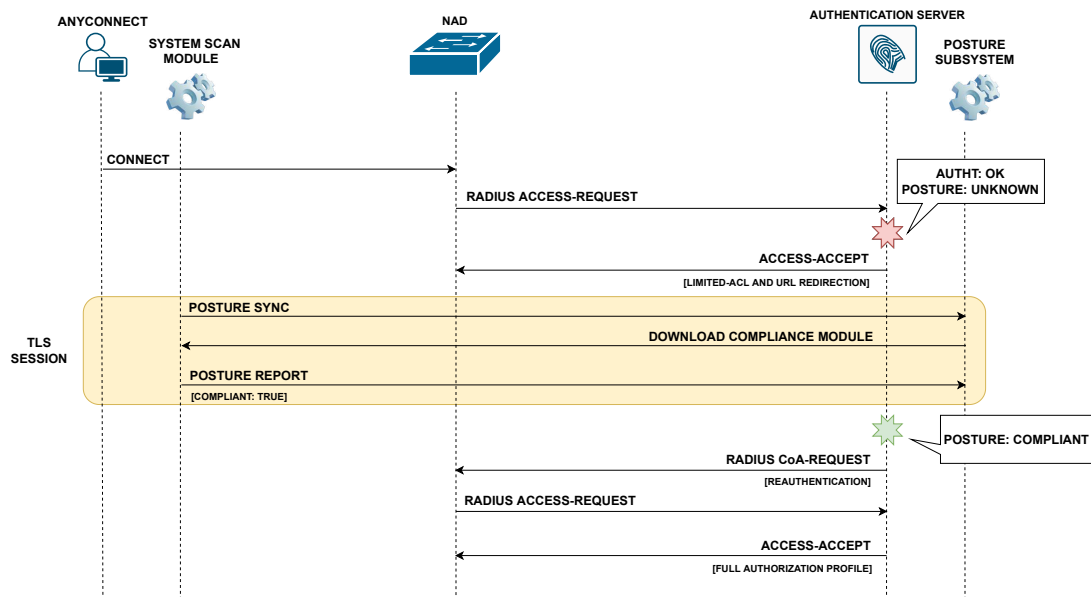


Figure 7.3: Posture process flow of a compliant device.

## 7.2 ISE posture flow

This section presents the various steps to follow to enable the posture functionality within ISE. Since both profiling and posture features are not native to ISE, it would be necessary to briefly illustrate the licensing model adopted by Cisco. For this discussion, reference is made to Section 8.2.1. Unlike the previous chapters (i.e., Chapter 5 and Chapter 6), in this case the NAD does not need any configuration. The changes to be made are all to be implemented through the ISE GUI. The configuration process is shown in Figure 7.4. Before carrying out the following 6 implementation steps, it is however necessary to carry out some important operations. The first thing to do is to enable posture and client provisioning features. It is therefore necessary to check the “*Enable Session*

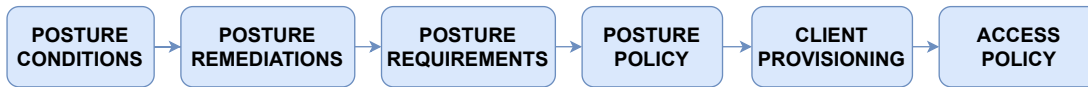


Figure 7.4: Posture configuration flow through the ISE GUI.

*Services*” service, as already shown in Figure 6.1. The posture functionality can only be enabled within nodes that assume the role of Policy Service persona. As for the test environment, there is only one node that assumes the role of all the personas. Secondly, is necessary to update the compliance module. This is the entity containing supported software, conditions and operating systems. This module is present both on the ISE side (inside the posture subsystem) and on the AnyConnect system scan module side. It therefore holds the responsibility for the correct functioning of the posture assessment. Figure 7.5 shows the tab with which it is possible to check the settings regarding the updates of the compliance module. It is possible to configure an online and offline update. To automate the process it is recommended to configure the online update every certain time period.

**Posture Updates**

Web  Offline

\* Update Feed URL

Proxy Address

Proxy Port  HH MM SS

Automatically check for updates starting from initial delay    every  hours

---

**▼ Update Information**

Last successful update on	2022/05/26 16:36:41 <input type="button" value="i"/>
Last update status since ISE was started	Last update attempt at 2022/05/26 16:36:41 was successful <input type="button" value="i"/>
Cisco conditions version	279375.0.0.0
Cisco AV/AS support chart version for windows	262.0.0.0
Cisco AV/AS support chart version for Mac OSX	180.0.0.0
Cisco supported OS version	71.0.0.0

Figure 7.5: Posture update interface.

Once the basic configurations have been made, it is possible to describe the various steps with which the assessment of posture is configured. In the following sections, different steps to carry out a correct configuration of the posture functionality are reported (considering a deployment using the AnyConnect agent).

## 7.2.1 Posture conditions

Posture conditions are those tools that the agent uses to evaluate the status of the device. There are a large number of conditions that can be used (depending



on the type of deployment used as mentioned in the previous section). Some of these are listed below:

- Hardware attribute condition: verification of some hardware parameters of the requesting device;
- Application: verification of the presence/execution of a specific application;
- Firewall condition: verification of the operability of the security application;
- Anti-Malware: similar to the Application one, it verifies the presence/execution of a specific Anti-Malware application;
- Disk encryption: check the encryption status of a storage drive;
- File: check for the presence of a particular file within a directory;
- USB: check if the device is connected, via USB port, with another device.

In general, the conditions can be of two types: simple or compound. Simple conditions provide for the definition only of the evaluation of a single element. An example of a simple condition is the following: “*A operand B*”, where *A* is the attribute to inspect and *B* is the value that *A* must assume. In Figure 7.6 an example is shown, relating to firewalls. Compounds, on the other hand, are a container of simple conditions. In fact, through them, multiple assessments can be carried out through only one rule. An example of a compound condition is the following: “*(A operand B) AND (C operand D) OR (E operand F)*” and so on. Some conditions are provided by default by ISE.

Name	Description
Default_Firewall_Condition_Win	Cisco Predefined Check for firewall
Default_Firewall_Condition_Mac	Cisco Predefined Check for firewall

Figure 7.6: Posture conditions interface.

However, it is possible to create conditions to satisfy specific needs. The interface to create one of these, again relating to firewalls, is presented in Figure 7.7. For each type of condition there are different aspects to configure. For simplicity, through Figure 7.7, some types of configurations are shown. As a first step it is best practice to assign a name to the condition that summarizes its role. Next step is to select the correct compliance module. Also in this case it is recommended to select the most recent one. It is then necessary to choose which operating system this condition refers to. In fact, depending on the type of OS, different vendor solutions are suggested. By way of example, “*AVAST Software a.s.*” was chosen as the vendor. In the lower part of Figure 7.7 there are all the

software (classified as firewall) and their available versions. Through the option “*Enable*”, the posture module is able to check if the software, at the specified version, is enabled in that device. To be specified as depending on the type of condition, several options are available. In fact, considering the condition for applications, the available options are “*Installed*” and “*Running*”. Furthermore, again for the scenario concerning the applications, it is possible to choose which one, between “*Application*” and “*Process*”, to verify. If “*Process*” is selected, it is then necessary to specify its name with its extension (e.g., *.exe*).

Firewall Conditions > New  
Input fields marked with an asterisk (\*) are required.

Name \*

Description

Compliance module \* 4.x or later

Operating System \* Windows All

Vendor \* AVAST Software a.s.

Enable

At least one product must be selected \*

<input type="checkbox"/>	Product Name	Version
<input type="checkbox"/>	Avast Business Security	10.x
<input type="checkbox"/>	Avast Business Security	12.x
<input type="checkbox"/>	Avast Business Security	17.x
<input type="checkbox"/>	Avast Business Security	18.x
<input type="checkbox"/>	Avast Business Security	19.x

Figure 7.7: Custom firewall condition configuration interface.

## 7.2.2 Posture remediations

Once condition have been configured, it is necessary to define the various types of remediation. Remember how remediation is the procedure by which a device tries to become *Compliant* from an *Unknown* or *Non-Compliant* state. The configuration of these tools is very similar to the previous one. The interface with which it is possible to view all types of remediation is in fact very similar to that shown in Figure 7.7. Also in this case, in addition to the remediation procedures present by default, it is possible to define more specific ones. An example of these is shown in Figure 7.8.

The first 4 options are the same as the one presented above. The fifth option, on the other hand, is entirely new. Through it, it is possible to specify the type of remediation to be used. The available options are “*Automatic*” and “*Manual*”. The “*Automatic*” option requires the AnyConnect agent to carry out the remediation operation autonomously (without the user interacting with the user interface). It is possible to specify the interval value every time the agent tries to perform the remediation operation. Similar concept for the “*Retry-Count*” field

The screenshot shows a configuration form for a custom firewall remediation. The fields are as follows:

- Name \***: An empty text input field.
- Description**: An empty text input field.
- Operating System**: A dropdown menu with "Windows All" selected.
- Compliance module**: A dropdown menu with "4.x or later" selected.
- Remediation Type \***: A dropdown menu with "Automatic" selected.
- Interval \***: A text input field containing "0", with a note "(in secs) Valid Range 0 to 9999" below it.
- Retry Count \***: A text input field containing "0", with a note "Valid Range 0 to 99" below it.
- Vendor Name \***: A dropdown menu with "AVAST Software a.s." selected.

At the bottom, there is a checked checkbox with the label "Remediation Options is to enable the Firewall".

Figure 7.8: Custom firewall remediation configuration interface.

with which the maximum number of attempts to carry out the remediation can be specified. The last available option is what is called “*Remediation Options is to enable the Firewall*”. Through it, it is possible to specify which remediation operation to perform. In this case, the remediation operation is intended to enable the firewall, if found disabled. As in the previous case, depending on the type of remediation, different options are available.

Figure 7.9 shows an example of an interface for patch management remediation. Note that 3 remediation options are available.

### 7.2.3 Posture requirements

Once the network admin has created the condition and posture remediations, it is necessary to configure the posture requirements to relate them to each other. Figure 7.10 shows the interface with which it is possible to define these tools. As a first option, the operating system that this requirement is for must be selected. In this option, several levels of granularity are available (e.g., starting from the generic *Windows* OS up to the *Enterprise*, *Business* and so on versions). Obviously, if the conditions and remediations refer to a certain OS, the related requirements must also be aligned with them. Following are reported the compliance module type and posture type. The latter makes AnyConnect and AnyConnect Stealth available as an option. Finally, in the last two fields, the conditions and remediation objects created in the previous sections are inserted.

Patch Management Remediations List > [New Patch Management Remediation](#)

### Patch Management Remediation

\* Name  ⓘ

Description

Operating System Windows

Compliance Module

Remediation Type

\* Interval  (Valid Range 0 to 9999)

\* Retry Count  (Valid Range 0 to 99)

\* Patch Management Vendor Name

Remediation Option  Enable  Install missing patches  Activate patch management software GUI

Check patches installed

---

▼ Products for Selected Vendor

Product Name	Version	Enabled Remediation Support	Update Remediation Support	Show UI Remediation Support
McAfee ePolicy Orchestrator Agent	4.x	NO	NO	YES
McAfee ePolicy Orchestrator Agent	5.x	NO	NO	YES
Seguridad Dispositivo	16.x	NO	NO	YES
Seguridad Dispositivo	17.x	NO	NO	YES

Figure 7.9: Custom patch management remediation configuration interface.

Note that each condition corresponds to only one remediation operation.

Name	Operating System	Compliance Module	Posture Type	Conditions	Remediations Actions
Any_AV_Installation_Win	for Windows All	using 3.x or earlier	using AnyConnect	met if ANY_av_win_inst	then Message Text Only <a href="#">Edit</a>
Any_AV_Definition_Win	for Windows All	using 3.x or earlier	using AnyConnect	met if ANY_av_win_def	then AnyAVDefRemediationWin <a href="#">Edit</a>
Any_AS_Installation_Win	for Windows All	using 3.x or earlier	using AnyConnect	met if ANY_as_win_inst	then Message Text Only <a href="#">Edit</a>
Any_AS_Definition_Win	for Windows All	using 3.x or earlier	using AnyConnect	met if ANY_as_win_def	then AnyASDefRemediationWin <a href="#">Edit</a>
Any_AV_Installation_Mac	for Mac OSX	using 3.x or earlier	using AnyConnect	met if ANY_av_mac_inst	then Message Text Only <a href="#">Edit</a>
Any_AV_Definition_Mac	for Mac OSX	using 3.x or earlier	using AnyConnect	met if ANY_av_mac_def	then AnyAVDefRemediationMac <a href="#">Edit</a>
Any_AS_Installation_Mac	for Mac OSX	using 3.x or earlier	using AnyConnect	met if ANY_as_mac_inst	then Message Text Only <a href="#">Edit</a>
Any_AS_Definition_Mac	for Mac OSX	using 3.x or earlier	using AnyConnect	met if ANY_as_mac_def	then AnyASDefRemediationMac <a href="#">Edit</a>
Any_AM_Installation_Win	for Windows All	using 4.x or later	using AnyConnect	met if ANY_am_win_inst	then Message Text Only <a href="#">Edit</a>

Figure 7.10: Posture requirements interface.

## 7.2.4 Posture policy

Since ISE was introduced, it has been said that it bases its operation on the definition of policy sets. As already verified for the profiling function, also for the posture this concept is strongly recalled. After the first 3 configuration steps, the network administrator must define what are the posture policies (Figure 7.11). The posture policy determines which requirements are mandated and for whom. It is important to remember that, to be defined *Compliant*, a device must satisfy all the requirements related to it. From Figure 7.11 can be seen that, not only some policies are already present by default, but these are also disabled. As in the other cases, it is possible to define custom policies. As far as the available options are concerned, in this case there is a new option called “*Grace period settings*”. With this option it is possible to define the period of time for which

a device, which is *Non-Compliant*, can access the network (thus obtaining the authorization).

It is also possible to configure an automatic warning message, whose purpose is to notify the user of the progress of this period. Once this period has expired, the device gains access to the network only if it is evaluated as *Compliant*. Subsequently it is possible to define which identity group this policy is to be associated with. Usually (and also following best practices) this field should be set to the value of *any*. The fields in the central part of the figure have the same meaning as those described in the previous sections. The last field instead determines which requirements are associated with this policy. Remember, as in the previous case, the attributes relating to the OS must correspond to each other. It is therefore not possible to define a requirement for a certain OS, and then insert it within a policy addressed to another type of OS.

Posture Policy  
Define the Posture Policy by configuring rules based on operating system and/or other conditions.

Status	Policy Options	Rule Name	Identity Groups	Operating Systems	Compliance Module	Posture Type	Other Conditions	Requirements	
⊕	Policy Options	Default_AntiMalware_Policy_Mac_temporal	Employee	Mac OSX	4.x or later	AnyConnect	(Optional) Dictionary	Any_AM_Installation_Mac_temporal	Done
⊕	Policy Options	Default_AntiMalware_Policy_Win_temporal	Any	Windows All	4.x or later	AnyConnect	and	Any_AM_Installation_Win_temporal	Edit
⊕	Policy Options	Default_AntiMalware_Policy_Win_temporal	Any	Windows All	4.x or later	Temporal Agent	and	Any_AM_Installation_Win_temporal	Edit
⊕	Policy Options	Default_AppVis_Policy_Mac	Any	Mac OSX	4.x or later	AnyConnect	and	Default_AppVis_Requirement_Mac	Edit
⊕	Policy Options	Default_AppVis_Policy_Mac_temporal	Any	Mac OSX	4.x or later	Temporal Agent	and	Default_AppVis_Requirement_Mac_temporal	Edit
⊕	Policy Options	Default_AppVis_Policy_Win	Any	Windows All	4.x or later	AnyConnect	and	Default_AppVis_Requirement_Win	Edit

Figure 7.11: Posture policy interface.

## 7.2.5 Client provisioning

Once the network administrator has finished configuring the assessment options, it is needed to define the client provisioning process. In fact, behind the communication between the agent and ISE there is a deployment of a series of elements to be defined. The first step to follow has already been introduced and discussed at Section 7.2. The second step is to download the AnyConnect agent from the official *software.cisco.com* website (shown in Figure 7.12). In fact, it is not natively integrated within ISE. However, it is not the AnyConnect lightweight software that needs to be loaded, but its respective headend. This software, integrated with ISE, allows the central management of configurations and updates of the clients located within the devices. At this point, the focus goes on to the configuration of the client provisioning resources (whose interface is shown in Figure 7.13). It is necessary to upload the file that has just been downloaded. From the interface shown in Figure 7.13, the AnyConnect agent is added (by choosing the “*Agent Resources from Local Disk*” option). At this point, the compliance module must be downloaded (again from the *software.cisco.com* website). In the case study, as already pointed out initially, only applications for *Windows* OS are used.

Once the previous 2 steps have been completed, it is now necessary to create a posture profile. Following what is shown in Figure 7.14 different options are available. It is therefore possible to define different profiles, which can then be






























Language localization transform Pre-Deployment (Windows) 	09-Jun-2022	0.67 MB	 
anyconnect-win-4.10.05111-core-vpn-lang-predeploy-k9.zip			
<a href="#">Advisories</a> 			
Language localization transform Headend Deployment (Windows) 	09-Jun-2022	0.67 MB	 
anyconnect-win-4.10.05111-core-vpn-lang-webdeploy-k9.zip			
<a href="#">Advisories</a> 			
AnyConnect Pre-Deployment Package (Windows) - includes individual MSI files 	09-Jun-2022	66.60 MB	 
anyconnect-win-4.10.05111-predeploy-k9.zip			
<a href="#">Advisories</a> 			
Application Programming Interface [API] (Windows) 	09-Jun-2022	142.33 MB	 
anyconnect-win-4.10.05111-vpnapi.zip			
<a href="#">Advisories</a> 			
AnyConnect Headend Deployment Package (Windows) 	09-Jun-2022	79.20 MB	 
anyconnect-win-4.10.05111-webdeploy-k9.pkg			
<a href="#">Advisories</a> 			
AnyConnect Pre-Deployment Package (Windows 10 ARM64) - includes individual MSI files 	09-Jun-2022	36.77 MB	 
anyconnect-win-arm64-4.10.05111-predeploy-k9.zip			
<a href="#">Advisories</a> 			
AnyConnect Headend Deployment Package (Windows 10 ARM64) 	09-Jun-2022	48.37 MB	 
anyconnect-win-arm64-4.10.05111-webdeploy-k9.pkg			
<a href="#">Advisories</a> 			

Figure 7.12: Cisco AnyConnect download page.

Resources

Selected 0 | Total 10 





  Add  Duplicate  Delete	Name	Type	Version	Last Update	Description
<input type="checkbox"/>	Agent resources from Cisco site				
<input type="checkbox"/>	Agent resources from local disk				
<input type="checkbox"/>	Cisco Native Supplicant Profile	CiscoTemporalAgentWindows	4.8.176.0	2019/11/14 02:14:29	With CM: 4.3.838.6145
<input type="checkbox"/>	Cisco AnyConnect Configuration	CiscoTemporalAgentOSX	4.10.6024.0	2022/05/26 16:53:47	Cisco Temporal Agent for OSX ...
<input type="checkbox"/>	Mac AnyConnect Posture Profile	MacOxSPWizard	2.7.0.1	2019/11/14 02:14:28	Supplicant Provisioning Wizard f...
<input type="checkbox"/>	Win AMP Enabler Profile	WinSPWizard	2.7.0.1	2019/11/14 02:14:28	Supplicant Provisioning Wizard f...
<input type="checkbox"/>	CiscoTemporalAgentWindows 4.10.0...	CiscoTemporalAgentWindows	4.10.6024.0	2022/05/26 16:53:41	Cisco Temporal Agent for Windo...
<input type="checkbox"/>	CiscoTemporalAgentOSX 4.8.00176	CiscoTemporalAgentOSX	4.8.176.0	2019/11/14 02:14:32	With CM: 4.3.761.4353
<input type="checkbox"/>	Cisco-ISE-Chrome-NSP	Native Supplicant Profile	Not Applicable	2016/10/06 22:01:12	Pre-configured Native Supplicant...
<input type="checkbox"/>	AnyConnectComplianceModuleWind...	AnyConnectComplianceMo...	4.3.2815.6145	2022/05/26 16:53:56	AnyConnect Windows Complian...
<input type="checkbox"/>	Cisco-ISE-NSP	Native Supplicant Profile	Not Applicable	2016/10/06 22:01:12	Pre-configured Native Supplicant...
<input type="checkbox"/>	AnyConnectComplianceModuleOSX ...	AnyConnectComplianceMo...	4.3.2431.4353	2022/05/26 16:54:00	AnyConnect OSX Compliance M...

Figure 7.13: Client provisioning resources interface.

associated with different agents to be made available to the devices. It should be noted that the available options are numerous, while in Figure 7.14 only a part of them are shown. Among these, those on which to pay more attention are the following:

- Remediation timer: this is the time period after which the device is classified as *Non-Compliant* after being initially classified as *Unknown*. The value associated with it has been left at the default (i.e., 4 minutes);
- Stealth mode: through this option, the end user is not shown any message or notification regarding the status of the device. In fact it is possible that in certain enterprise environments, the user does not have enough privileges to be able to remedy to these problems;
- Enable agent IP refresh: highly useful option that allows the device to detect a change in VLAN. Through it, the device will therefore be able to



ISE Posture Agent Profile Settings > Posture\_Profile\_Windows

\* Name:   
 Description:

Agent Behavior

Parameter	Value	Notes	Description
Enable debug log	<input type="text" value="No"/>		Enables the debug log on the agent
Operate on non-802.1X wireless	<input type="text" value="No"/>		Enables the agent to operate on non-802.1X wireless networks.
Enable signature check	<input type="text" value="No"/>	OSX: N/A	Enables signature checking of executables before the agent will run them.
Log file size	<input type="text" value="5"/> MB		The maximum agent log file size
Remediation timer	<input type="text" value="4"/> mins	Default Value of global setting - 4. Acceptable Range between 1 to 300. Accept only integer Values.	The time the user has for remediation before they will be tagged as non-compliant
Stealth Mode	<input type="text" value="Disabled"/>		AnyConnect can act as either clientless or standard mode. When stealth mode is enabled, it runs as a service without any user interface.
Enable notifications in stealth mode	<input type="text" value="Disabled"/>		Enables error notifications in stealth mode. Disabled by Default.
Parallel Check Optimization	<input type="text" value="Enabled"/>		
Enable Rescan Button	<input type="text" value="Enabled"/>		Enables 'Rescan' button on System Scan tile. This allows users to force a rerun of posture policies as well as posture module to ISE discovery from the endpoint.
Disable UAC Prompt	<input type="text" value="No"/>	Windows only. Applicable if user has administrator privileges.	By turning off UAC Prompt, AC posture uses system process for privilege escalation instead of 'Run as administrator'. Please validate your posture policies on machine where users have local admin rights prior to disabling UAC prompt.
Periodic probing	<input type="text" value="3"/> x 10 mins	Supported range is between 0 - 30. '0' disables periodic probing.	Enable/Disable periodic discovery probes in AnyConnect after back-off timer crosses back-off timer limit. AnyConnect will send periodic probes with the given interval continuously till valid ISE is found.
Automated DART Count	<input type="text" value="3"/>		Set the number of automated dart bundles to be collected during failure scenarios.

Figure 7.14: Posture profile interface.

obtain a correct IP addressing. In our case it is extremely useful as the device, if considered *Compliant*, will be assigned to a new VLAN.

As the second to last step regarding the configuration of the client provisioning process components, it is necessary to define a configuration of the AnyConnect agent. From Figure 7.15 it is possible to see how through this procedure the AnyConnect agent is linked to the compliance module. In the central part it is possible to define which of the modules to enable. In the case study, only the module relating to posture has been enabled. Instead, within the option below it is possible to specify which posture profile to use (relative to the corresponding AnyConnect module).

The last step consists in defining the Client Provisioning Policy (**CPP**). By default, as in the previous cases, the concept of policy set is re-adapted. The CPP interface is shown in Figure 7.16. The matching rules on which the various policies are based, are the Identity Group to which they belong, and its OS. There is also the possibility of inserting additional conditions to increase the granularity of the policies.

At the end of these steps, it remains only to use the posture checks within the policy sets. The next section therefore illustrates how to integrate the posture conditions with the policy sets already defined in the previous sections.

As a last note, remember that before scanning the device for postures, the user must download the agent through the ISE client provisioning portal. The latter can be modified both from the point of view of the user interface and from the point of view of operation. However, these types of configurations are not strictly necessary for the case study. For this reason the default configurations were used. Figure 7.17 shows the 3 components (discussed in the previous sections) that have been installed within the *Windows* workstation.

AnyConnect Configuration > AnyConnect Configuration

\* Select AnyConnect Package: AnyConnectDesktopWindows 4.10.5111.0

\* Configuration Name: AnyConnect\_Configuration\_WN

Description:

**DescriptionValue**

\* Compliance Module: AnyConnectComplianceModuleWindows 4.3.2815.614

**Notes**

---

**AnyConnect Module Selection**

ISE Posture

VPN

Network Access Manager

AMP Enabler

ASA Posture

Network Visibility

Umbrella Roaming Security

Start Before Logon

Diagnostic and Reporting Tool

---

**Profile Selection**

\* ISE Posture: Posture\_Profile\_WN

VPN

Network Access Manager

AMP Enabler

Network Visibility

Umbrella Roaming Security

Customer Feedback

Figure 7.15: Posture AnyConnect configuration interface.

**Client Provisioning Policy**

Define the Client Provisioning Policy to determine what users will receive upon login and user session initiation.  
 For Agent Configuration: version of agent, agent profile, agent compliance module, and/or agent customization package.  
 For Native Supplicant Configuration: wizard profile and/or wizard. Drag and drop rules to change the order.

Rule Name	Identity Groups	Operating Systems	Other Conditions	Results
<input checked="" type="checkbox"/> IOS	If Any	and Apple IOS All	and Condition(s)	then Cisco-ISE-NSP
<input checked="" type="checkbox"/> Android	If Any	and Android	and Condition(s)	then Cisco-ISE-NSP
<input checked="" type="checkbox"/> Windows	If Any	and Windows All	and Condition(s)	then CiscoTemporalAgentWindows 4.8.00176 And WinSPWizard 2.7.0.1 And Cisco-ISE-NSP
<input checked="" type="checkbox"/> MAC OS	If Any	and Mac OSX	and Condition(s)	then CiscoTemporalAgentOS X 4.8.00176 And MacOsKSPWizard 2.7.0.1 And Cisco-ISE-NSP
<input checked="" type="checkbox"/> Chromebook	If Any	and Chrome OS All	and Condition(s)	then Cisco-ISE-Chrome-NSP

Figure 7.16: Client provisioning policy interface.

Cisco AnyConnect ISE Compliance Module	Cisco Systems, Inc.	01/08/2022	26,4 MB	4.3.2815.6145
Cisco AnyConnect ISE Posture Module	Cisco Systems, Inc.	01/08/2022	5,13 MB	4.10.05111
Cisco AnyConnect Secure Mobility Client	Cisco Systems, Inc.	01/08/2022	5,85 MB	4.10.05111

Figure 7.17: AnyConnect files installed within the workstation.

## 7.2.6 Access policy

Once all aspects concerning configurations and provisioning have been defined, all that remains is to integrate the concepts of posture within the policy sets. Within the policy sets there are some conditions related to posture. One



of these is the “*Session-Posture-Status*”. This attribute can only take 3 values: *Unknown*, *Compliant* and *Non-Compliant*. When a device connects for the first time, its posture status is *Unknown*. In fact, ISE is unable to verify its status via AnyConnect. At this point the user is redirected to the ISE portal where he can download the AnyConnect agent (in case this is the first time he connects to it). The first operation that the AnyConnect agent does, after being downloaded and installed, is called “*Discovery Phase*”. During this phase ISE sends some discovery packets to some of the following destinations listed below:

- HTTP packet to Port 80 (if configured);
- HTTPS to Port 8905 (if configured);
- HTTP to Port 80 on the default gateway;
- HTTP to Port 80 on *enroll.cisco.com*.

AnyConnect then tries to check the posture status of the device and communicates it to ISE. Once the process is complete, the device is registered as *Compliant*.

Figure 7.18 shows the process through which a device goes through (in case the AnyConnect agent is already installed). This situation turns out to be the ideal one to apply in the production environment. Remember how the inspection of the logs must be done from the bottom up. First of all it is specified once again that the device, being a corporate asset, is configured with an 802.1X agent. During this authentication phase, the policy set used is therefore that relating to this protocol. In the first row, can be seen how the device is associated with the *Non-Compliant* state. Being in this one, the provisioning profile “*Employee\_Unkn\_NoCompli\_Cisco*” is associated with it. In fact, from the second row, it is possible to see the download of the ACL (i.e., “*DeployAgent*” ACL) related to the client provisioning. It is important to underline at this point that devices associated with the states *Unknown* and *Non-Compliant* are associated with the same profile. This choice was dictated by the fact that there are no particular requirements to draw a distinction between these two states. At Section 8.2.3, however, a solution is presented for which devices with different states are transparent to each other (and therefore cannot communicate).

On the third row can be seen how the communication between AnyConnect and ISE was successful, evaluating the device as *Compliant*. On the fourth row the device is profiled as “*Microsoft-Workstation*”. The profiling process is similar to the one illustrated in Chapter 6. Then another authentication process takes place due to new information received from the NAD through the device-sensors. This new information relates to IP addressing. In fact, remember that there are 2 networks associated with the *Compliant* and *Non-Compliant* states. If at the beginning the device had obtained an addressing of the type *X.X.242.50*, then changing subnet it received another one of type *X.X.7.191*. Since the flow was captured during the test phase, the authorization profile associated with the employees included a simple ACL whose task was to block any ping attempt (already reported in the previous chapter). The last row instead indicates the device session.

Status	Details	Repeat...	Identity	Endpoint ID	Endpoint Profile	Authenticatio...	Authorization Policy	Identity Group	Posture St...	Auth Method	Authentication Protocol
		0	IC	F4:30:B9:	Microsoft-Workstation	802_1X_Wired ...	802_1X_Wired ==> Employee_WN_Compliant	Workstation	Compliant	dot1x	PEAP (EAP-MSCHAPV2)
			#ACSACL:IP-Deny-Ping...	F4:30:B9:	Microsoft-Workstation	802_1X_Wired ...	802_1X_Wired ==> Employee_WN_Compliant	Workstation	Compliant	dot1x	PEAP (EAP-MSCHAPV2)
			IC	F4:30:B9:	Microsoft-Workstation	802_1X_Wired ...	802_1X_Wired ==> Employee_WN_Compliant	Workstation	Compliant	dot1x	PEAP (EAP-MSCHAPV2)
			#ACSACL:IP-DeployAge...	F4:30:B9:					Pending	dot1x	PEAP (EAP-MSCHAPV2)

Figure 7.18: Posture assessment flow.

## 7.3 Posture service set-up

The configuration concerning the posture service, as previously mentioned, only relates to ISE. However, some configuration changes have been made within the NAD. The final configurations made in the test environment are illustrated below.

### 7.3.1 NAD set-up

During the presentation of the current chapter, it was said that the NAD has not undergone any configuration changes. However, minor changes have been made in order to simulate the functioning of the final solution to implement in production environment. Before introducing them, however, it is necessary to make a premise. In the previous chapters some VLANs have been defined (in particular VLAN 400 for remediation procedures and 501 for welcoming consultants). In order to finalize the configuration of the posture functions, however, it was necessary to make changes to the numbering of the VLANs as they were already defined within the production environment (i.e., within the campus network and ACI). The VLANs 400 and 501 have therefore been redefined in 599 and 598. Furthermore, unlike what is shown in the previous chapter (Section 5.1.7), a closed mode approach has been applied to the deployment. In fact, the following 2 configurations have been deleted:

```
SwitchNAC(config)#no authentication open
SwitchNAC(config)#ip access-group WELCOMEACL in
```

In this scenario, it is therefore not possible for a user to exploit network services without first authenticating. Another change to note, and of great importance, is the definition of a new ACL. Initially named as “*POSTUREACL*” and then renamed as “*REDIRECT\_CPP*”, it has the purpose of capturing the traffic and redirecting it to the ISE portal. Not all the traffic generated by the device must be redirected though. In fact, in the test environment, the Welcome VLAN was not associated with a DHCP server (and therefore with a DHCP scope). The device that results *Unknown* or *Non-Complaint* must be able to contact a DHCP server to obtain an IP address in order to connect to the ISE Client Provisioning Portal. It also needs to be able to contact a DNS server in order to resolve FQDN. The following ACL, has been statically defined within the NAD, while its identifier

(i.e., name of the ACL) has been specified in ISE.

```
deny udp any eq bootpc any eq bootps
deny udp any any eq domain
deny udp any host <ISE-PSN> eq 8905
deny udp any host <ISE-PSN> eq 8906
deny tcp any host <ISE-PSN> eq 8443
deny tcp any host <ISE-PSN> eq 8905
deny tcp any host <ISE-PSN> eq www
permit ip any any
```

With the deny statement, the ACL define how the traffic that match this rule must not be redirected. On the other hand, for the permitted statement, the opposite is true. This ACL is therefore not used to block or allow the flow of traffic, but to redirect it. To also allow redirection to an external portal it is necessary to enable the HTTP/HTTPS server functions using the following 2 commands:

```
SwitchNAC(config)#ip http server
SwitchNAC(config)#ip http secure-server
```

It is important at this point to make a brief note on the architectural choice. A device can be associated with a remediation authorization profile in 2 different scenarios. The first, when the device has not carried out any posture checks and is therefore marked as *Unknown*. The second is when a device is defined *Non-Compliant*. The choice that best suits InfoCamere's needs is to use a single authorization profile for both cases. However, a different and possible, implementation turns out to be that of creating 2 different authorization profiles having different VLANs.

## 7.3.2 ISE set-up

### General settings

While the first part of the configuration was described above, this section shows the configurations related to the definition of the authorization profile and policy sets. Regarding the first, following the same logic of the previous chapters, an ad-hoc profile was created. The profile (i.e., "*Employee\_Unkn\_NoCompli\_Cisco*") is first associated with the dACL shown below.

```
permit udp any eq bootpc any eq bootps
permit tcp any host <ISE-PSN> eq 8443
permit tcp any host <ISE-PSN> eq 8905
permit udp any host <ISE-PSN> eq 8905
permit udp any host <ISE-PSN> eq 8906
permit tcp any host <ISE-PSN> eq 80
permit tcp any any eq 80
```

```

permit tcp any any eq 443
permit udp any any eq domain
permit icmp any any

```

This ACL, is applied (inbound) to the session related to the user. Always based on the previously reported reachability concepts, the device needs to communicate with some fundamental services (e.g., DHCP and DNS). However, since the device is in an intermediate state, it must have extremely limited visibility. Hence the creation of this ACL. As can be seen from the list, the traffic flow to ISE is initially allowed. However, it was also necessary to define the two *permit* statements to allow all traffic to ports 80 and 443. This is because the traffic is initially subject to the dACL and consequently to the redirect ACL. Therefore, if these 2 statements were not present, the traffic would not be redirected as it would be immediately blocked. However, it should be emphasized that this consideration differs according to the type of platform used. During the analyses, considerations were made with the “*Network Management*” business unit relating to the definition of some policies within the NGFW. In fact, within them it is possible to filter the traffic coming from the Remediation VLAN. Also note the structure of the ACL: the first ACEs defined are the most specific ones, i.e., those that intercept specific types of traffic.

Another option present in the profile is the one relating to redirection (reported in Figure 7.19).

▼ Common Tasks

Web Redirection (CWA, MDM, NSP, CPP) ⓘ

Client Provisioning (Posture) ▼      ACL      REDIRECT\_CPP      Value      Client Provisioning Portal (defa ▼

Static IP/Host name/FQDN

Suppress Profiler CoA for endpoints in Logical Profile

Figure 7.19: Web redirection for remediation authorization profiles.

This option specifies the type of provisioning to be carried out. In the case study, it is the *Client Provisioning Posture* option. Next, network administrator needs to specify which ACL to use for interception and redirection of traffic. This ACL was defined in the previous section. Finally, it is needed to select which portal to use for the user interface. As specified previously, a default portal was used in this case. However, ISE offers tools with which it is possible both to customize the graphical interface of the portal and to modify its configuration. Examples of the latter are:

- Portal settings (which includes allowed interfaces, HTTPS port, Authentication method);
- Login page settings (which includes auto login and AUP settings);

- Change password settings.

In addition to these, there are also options for re-authentication and session expiration (already presented in the previous chapters). Looking also at the value of the attributes that are forwarded to the NAD via the Access-Accept RADIUS messages it is possible to see the presence of an URL (Figure 7.20). Within the

```
Access Type = ACCESS_ACCEPT
Tunnel-Private-Group-ID = 1:599
Tunnel-Type = 1:13
Tunnel-Medium-Type = 1:6
DACL = DeployAgent
cisco-av-pair = url-redirect-acl=REDIRECT_CPP
cisco-av-pair = url-redirect=https://ip:port/portal/gateway?sessionId=SessionIdValue&portal=44fd6796-4ebf-40d3-a24d-afbbdd3fb10&action=cpp
Session-Timeout = 3600
Termination-Action = RADIUS-Request
Idle-Timeout = 3600
```

Figure 7.20: RADIUS AVP pair for posture redirection.

link shown in the figure it is possible to see how the session identifier is present (i.e., the same value shown if the “*show authentication sessions interface GigabitEthernet x/x/x*” command is used within the NAD CLI). As a last field instead, it is possible to notice how the specified action is “*CPP*”, that is the Client Provisioning Portal. As a last configuration to mention there is the PRA. It is in fact possible to define a PRA profile to which a device is associated. By defining a certain interval, the device is then scanned, thus allowing the device to be tracked throughout the duration of the session.

### Policy set and posture conditions

For the case study, it has been chosen to modify and add the conditions relating to the posture, only for devices that authenticate via 802.1X. In fact, devices that authenticate via MAB (e.g., printers, IP phones, printers, etc.) do not have the ability to support an agent like AnyConnect. Furthermore, the latter is not designed for devices belonging to the group of IoT devices or similar. The only change introduced therefore within the policy set (more precisely within the authorization policies) of 802.1X concerns the corporate devices of employees (i.e., PCs, tablets, etc.). For simplicity, only the policy set concerning the devices that have *Windows* OS is shown in Figure 7.21. For the other devices it is sufficient only to change the value of this attribute.

For the definition of the posture conditions, on the other hand, following a discussion with the SOC team, 2 functions have been introduced: in the former, a check is made towards the operating status of the native *Windows* firewall (i.e., Windows Defender Firewall), while in the latter the installation of any Anti-Malware software is verified. As the device is configured through a GPO, it is highly unlikely that these two checks will fail. However, it is possible to define conditions (and therefore remediation operations) to verify the presence of other software, or patches installed. This in-depth analysis, to be conducted together with the SOC, may be the objective of future work. The flow is shown in Figure 7.22 (similar to that of Figure 7.18). The only difference is that in Figure 7.22 the

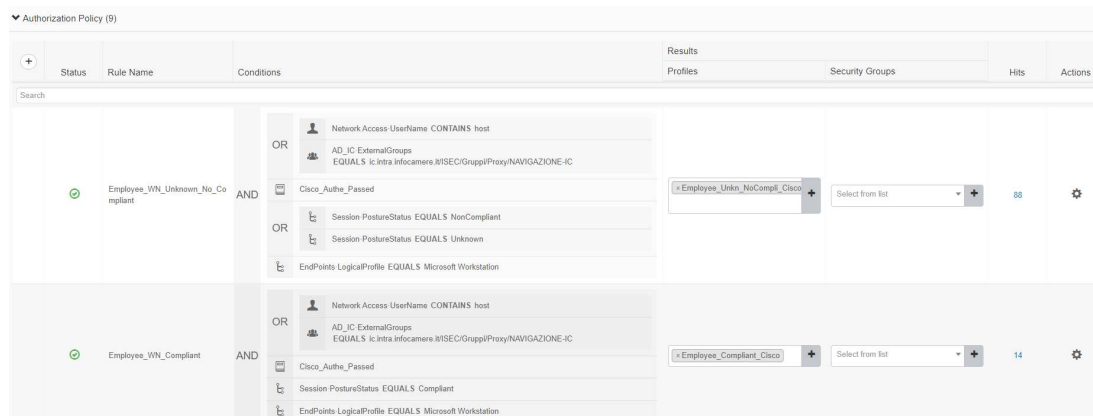


Figure 7.21: Authorization policy for posture assessment.

device is defined as *Non-Compliant* at a certain time. This behavior was in fact deliberately introduced. In addition to the 2 conditions mentioned above, a third one has been added (i.e., USB check, default condition) only for demonstration purposes. At the time AnyConnect scanned the device there was an USB external storage drive connected to the PC. This combination triggered an alarm from the agent, who promptly informed ISE. The latter then assigned the device to the remediation zone. As a last note, remember to keep the posture module

Identity	Endpoint ID	Endpoint Profile	Authenticatio...	Authorization Policy	IP Address	Identity Group	Posture ST...	Auth Method
Identity	Endpoint ID	Endpoint Profile	Authentication Pt	Authorization Policy	IP Address	Identity Group	Posture Status	Auth Method
IC	F4:30:B9	Microsoft-Workstation	802.1X_Wired ...	802.1X_Wired >> Employee_WN_Compliant	10		Compliant	dot1x
#ACSACL#-IP-Deny-Ping...								
IC	F4:30:B9	Microsoft-Workstation	802.1X_Wired ...	802.1X_Wired >> Employee_WN_Compliant	10	Workstation	Compliant	dot1x
	F4:30:B9						Compliant	
#ACSACL#-IP-Deny-Ping...								
IC	F4:30:B9	Microsoft-Workstation	802.1X_Wired ...	802.1X_Wired >> Employee_WN_Compliant	10	Workstation	Compliant	dot1x
	F4:30:B9						Compliant	
#ACSACL#-IP-DeployAge...								
IC	F4:30:B9		802.1X_Wired ...	802.1X_Wired >> Employee_WN_Unknown_No_Compliant			Pending	dot1x
#ACSACL#-IP-DeployAge...								
IC	F4:30:B9	Microsoft-Workstation	802.1X_Wired ...	802.1X_Wired >> Employee_WN_Unknown_No_Compliant	10	Workstation	NonCompliant	dot1x
	F4:30:B9						NonCompliant	
#ACSACL#-IP-DeployAge...								
IC	F4:30:B9	Microsoft-Workstation	802.1X_Wired ...	802.1X_Wired >> Employee_WN_Unknown_No_Compliant	10	Workstation	NonCompliant	dot1x
	F4:30:B9						NonCompliant	
#ACSACL#-IP-DeployAge...								
IC	F4:30:B9		802.1X_Wired ...	802.1X_Wired >> Employee_WN_Unknown_No_Compliant			Pending	dot1x

Figure 7.22: Posture assessment flow with remediation.

constantly updated. In case it wasn't, the creation of particular conditions could be a limiting factor as all the software and updates may not be present.

# Chapter 8

## Solution analysis

During the analyses of the main functionalities (i.e., AAA, profiling and posture) various solutions have been presented. With the addition of new features, as already explained, it has been necessary to make some changes (i.e., changing the numbering of the VLANs). In Figure 8.1 are reported 2 diagrams relating to the final solutions tested to be implemented also in the production environment (for simplicity in the production environment, only the VLANs relating to *Windows* workstations and IP phones have been considered). The next sections show

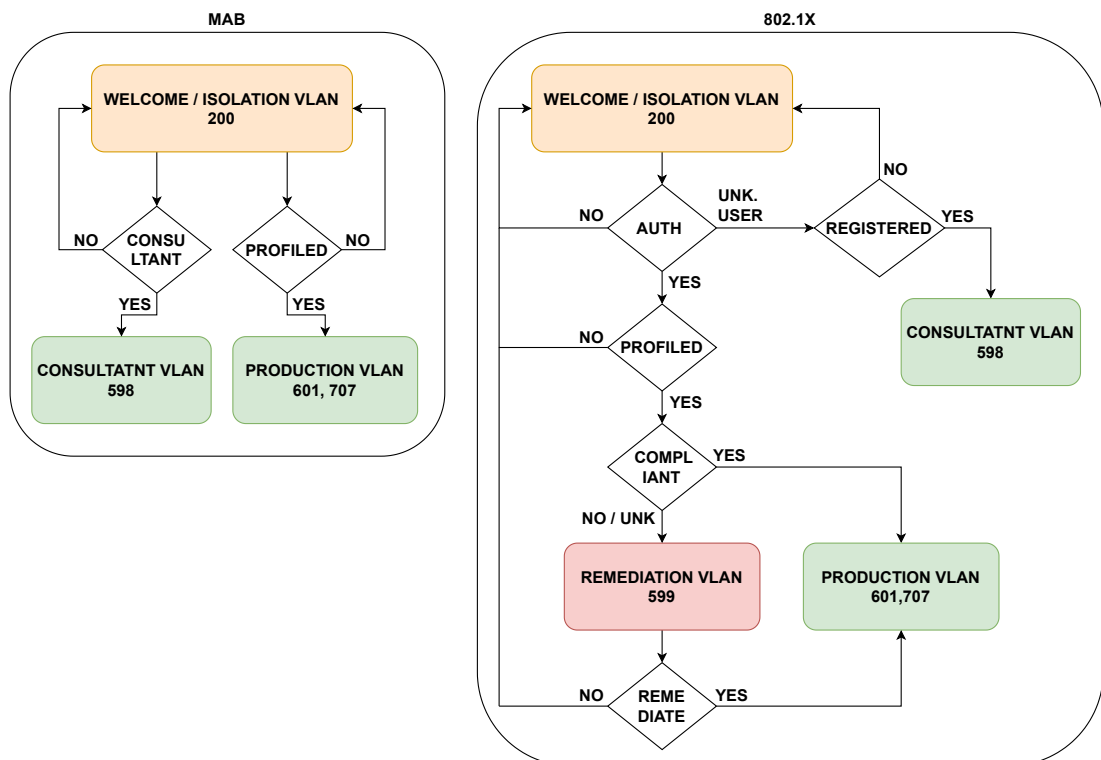


Figure 8.1: Final implementation schema.

the comparison analyses between the Cisco and Aruba NAC solutions (i.e., ISE and ClearPass) and the feasibility analyses of implementation in the production environment.

## 8.1 Short-list comparison

The analysis of the solution is divided into two sub-analyses. The first (discussed in the current section) in which the solutions proposed by Cisco and Aruba are compared (Table 8.1). The second instead deals with the analysis of the implementation within the context of the case study (InfoCamere). Within the list, the 3 functions of AAA, profiling and posture are first compared. In addition to these 3, some considerations regarding the management of the NAC system are illustrated. A more in-depth analysis of the latter is proposed at the Section 8.2.1.

It is important to underline how the comparison between the 2 products is carried out on a theoretical basis (without therefore having compared their respective implementations). The score given for each single item has the following value range: [0 – 5]. The value 0 indicates non-support of the specific feature. On the other hand, the maximum value of 5 indicates full support to it. Intermediate values (i.e., integer following range [1 – 4]) indicate partial support. The higher the score, the more well integrated the functionality is.

#	Feature	ISE	ClearPass
<b>AAA</b>			
1	Support for RADIUS	5	5
2	Support for TACACS+	5	5
3	<b>EAP</b>		
3.1	EAP-PEAP	5	5
3.2	EAP-TLS	5	5
3.3	EAP-TTLS	5	5
3.4	EAP-TEAP	5	5
3.5	EAP-FAST	5	5
3.6	EAP-FASTv2	5	0
3.7	EAP-LEAP	5	5
4	<b>External Identity Source Support</b>		
4.1	LDAP	5	5
4.2	AD	5	5
4.3	ODBC - MySQL	5	5
4.4	ODBC - Oracle	5	5
4.5	ODBC - PostgreSQL	5	5
4.6	ODBC - MicrosoftSQL	5	5



#	Feature	ISE	ClearPass
4.7	ODBC - Sybase	5	0
4.8	ODBC - MariaDB	0	5
4.9	Radius Token	5	2
5	Radius attribute within Authorization profile	5	4
6	Identity source sequence object	5	2
7	Login management of individual profiles	0	5
8	Troubleshooting tools support (tcpdump)	5	5
9	Allowed protocols source sequence object	5	3
10	Multiple Authorization profile association support	5	5
11	DTLS support	5	3
12	IPSEC communication support	4	4
13	Smart-Join cluster nodes to a domain	5	2
14	Certificate management options available	4	2
<b>Profiling</b>			
15	<b>Support for probe</b>		
15.1	RADIUS	5	5
15.2	DHCP	5	5
15.3	HTTP	5	5
15.4	DNS	3	0
15.5	NETFLOW	4	0
15.6	pxGRID	3	0
15.7	SNMP	5	5
15.8	NMAP	3	5
15.9	ActiveSync plugin	0	5
15.10	MDM	0	4
16	CoA support	5	5
17	Profile source sequence object	5	3
18	Profile hierarchy support	5	5
19	Automatic profiling info update	4	4
20	Profiling summary report	3	0

#	Feature	ISE	ClearPass
21	Automatic scan option within profile configuration	4	3
22	Evaluate Configuration Validator	3	0
<b>Posture</b>			
23	Provisioning portal customizable	5	5
24	Automatic posture compliance module update	3	3
25	Support for PRA	4	1
26	<b>Posture conditions</b>		
26.1	Application	5	5
26.2	Firewall	5	5
26.3	Anti-Malware	5	5
26.4	Disk encryption	3	0
26.5	File	3	5
26.6	Patch management	4	5
26.7	USB	2	0
26.8	Registry keys	0	5
27	Posture troubleshooting tool support	3	2
<b>System management</b>			
28	GUI responsive time	2	5
29	User-friendly interface	5	3
30	Redirect URL within the dashboard	2	5
31	Configuration via CLI	1	1
32	Configuration via GUI	5	4
33	Rest API	4	4
34	System already in use in InfoCamere	5	0
35	Node switchover function support	5	5

Table 8.1: NAC comparison between Cisco ISE and Aruba ClearPass.

So the final score  $S_v$  for a given vendor  $v$  (between “ISE” and “CP”) is calculated as follows (weighted average):

$$S_v = \frac{\sum_{g=1}^4 w_g \sum_{i=1}^{F_g} K_{giv}}{\sum_{g=1}^4 w_g \sum_{i=1}^{F_g}} \quad (8.1)$$

where:

- $w_g$  = weight (within range  $[0 - 5]$ ) assigned to the group  $g$ ;
- $F_g$  = total number of features corresponding to group  $g$ ;
- $v$  = ID of the vendor;
- $K_{giv}$  = score associated with the feature  $i$  of group  $g$  belonging to vendor  $v$ .

The groups denoted by  $g$  (with their respective weights) are the following:

- 5 AAA;
- 3 Profiling;
- 2 Posture;
- 5 System management.

Note how, by attributing to the variable  $k_{giv}$  a range value equal to  $[0 - 5]$ , it also performs the role of indicator function. The final results are therefore  $S_{ISE} = 4.14$  and  $S_{CP} = 3.68$  thus electing Cisco ISE as the solution to choose.

Therefore, taking into account the purely theoretical comparison, InfoCamere would obtain greater benefits by choosing to continue developing and improving the solution already present within the IT infrastructure. An important note to make in support of the result but linked to the case study is the following: the IC infrastructure, at the time of writing, has about 1000 Cisco switches and 200 Aruba switches (both of access type). These data corroborate the result found, since having a significantly higher number of Cisco switches compared to the Aruba ones, allow to have fewer interfacing issues between NADs and ISE (thanks to belonging to the same vendor).

## 8.2 InfoCamere case study analysis

After repurposed the general schemes on which the NAC bases its operation, it is necessary to carry out an analysis on the impact of this solution applied to the InfoCamere infrastructure.

### 8.2.1 Value, effort and risk analyses

At the same time as the test phases, it is necessary to conduct an analysis on the implementation in the production environment of the tested features. This analysis is divided into 3 micro-categories, such as Value, Effort and Risk.

#### Value

A service is considered of value if and only if the real performance and associated costs are feasible. The value associated with services such as profiling and posture lies in the type of impact they have within the company. The issue

of cybersecurity today holds a prominent importance, especially in those environments where services are provided and sensitive data are stored. In a reality like that of InfoCamere, it is of primary importance to ensure that the devices accessing the network are compliant. With the implementation of a similar solution there is therefore a significant increase in the effectiveness of the security perimeter. Through the function of AAA and access control, in addition to being effective, can be arranged with a level of microscopic granularity. Therefore, depending on the BU (device type) to which the employee (device) belongs and also on the authorization level, access can be totally redesigned. Through the profiling function, two types of innovative solutions are introduced (and improved): automation and context visibility.

The first relates to the fact that it is not necessary, for each new device, to be manually provisioned in the production environment and to create (or modify) access rules. ISE is able, through policies defined only in the initial deployment phase, to recognize the device and associate it with the correct authorization group. This allows network administrators not to have to manually manage each new device deployment. Another aspect that has not been discussed in this thesis, but which is increasingly important is programmability automation. Through the use of REST APIs (in ISE called Extensible RESTful Services, **ERS**), it is possible to automate, by means of scripts, a large part of the deployment and management of the entire NAC system. It is also possible to implement interfacing with other systems (e.g., Domain Controller), to improve the management of user traffic flows.

Context visibility, on the other hand, allows the company to carry out analysis on the type of devices connected to the network with extreme simplicity. This analysis can be used for purely statistical purposes, or to keep the visibility of the devices belonging to the internal network constantly updated. An application example of this function in a real scenario is that of a chamber of commerce. In fact, by obtaining some information from this feature, it is able to use them for commercial purposes.

Even if outside the scope of this thesis, it is important to underline how a function (used also for posture) is extremely useful in the case study of InfoCamere. Through ISE in fact it is possible to manage portal-based access. In a reality where guest-type networks are widely spread, their centralized management is of significant value.

## **Effort**

Effort analysis can be broken down into 2 parts: management effort and costs. The management effort determines the total effort that network administrators must devote to managing the NAC system. As a first note, it should be noted that the implementation of the profiling and posture features, even only in a test environment, require the supervision of a specialized employee. In fact, during the various phases of deployment, transversal knowledge proved to be necessary. Both functions are very useful features in general but require a non-

trivial management process. Since access control is based on user assessment, it is also necessary to design (or in any case manage) external identity sources. Therefore, in addition to the management of the system itself, it is necessary to make the interface between NAC and external storage (e.g., AD) as functional as possible. This request can therefore represent a time-consuming effort. As for the effort to dedicate, in the event that the system has been implemented in a production environment, it would be of a much higher complexity. In fact, having an integration of so many technologies makes managing the interfacing between them much more complex. In addition to this, there is also the issue of configuring the same functionalities within different switches, WLCs and APs vendors (see Section 6.5). As reported in Chapter 5, not all vendors support the same configurations. This fact represents another effort index, as it is necessary to test the same operation on multiple platforms using different CLI syntax.

Now let's consider the NAC system as a service that IC can provide, for example, to the chambers of commerce. In this case, each chamber has potentially different network devices (which are supplied through CONSIP). Since these devices are different (with maybe different software releases), they offer different functionalities. It may therefore not be possible to implement the same type of services. Another important note, closely related to what has been said before, is the operational deployment phase. In fact, a good knowledge of commands is required for multiple vendors.

When analyzing the expected costs of a deployment, there are two main aspects to consider: training costs and licensing costs. The first point is linked to what was said before, and refers to the fact that previous training is required to provide a service if a certain technology is deployed. Furthermore, to better understand the real functioning of the technology, a hands-on training phase is necessary. The second point, on the other hand, is also of primary importance. Currently, the licensing model defined by Cisco for its products (and therefore also for ISE) is called "*Smart Licensing*". As of ISE 3.X this is the only method available. In the case of InfoCamere (ISE 2.7), the licensing model called "*Product Authorization Key*" (**PAK**) is still available. This model uses the concepts of packages. These ones are a set of functions that ISE is able to implement. These packages are defined as *Base*, *Plus* and *Apex*. For the former, the functions of AAA, 802.1X and guest service are made available. For the second, instead, the functions of profiling, context sharing and BYOD, while the latter contains the posture and Mobile Device Management (**MDM**) functions. Depending on the business need, different choices can be made. Once the corporate has purchased 1 or more packages, the only costs related to them are those referred to technical support. The company is therefore required to pay an initial expense, and then an annual expense equal to approximately 15% of the initial one. In smart mode, on the other hand, the delivery method is transformed into a subscription model. The services provided are also in this case 3, namely *Essentials*, *Advantage* and *Premier*. Below is the correspondence between the previous 2 models:

- *Base* - *Essentials*;

- *Base & Plus - Advantage*;
- *Base & Plus & Apex - Premier*;

For this model, it is necessary to make a subscription (lasting 1, 3 or 5 years) to the service that the corporate needs. The amount to be paid at each deadline turns out to be the full amount. In the face of an increase in the price of the service, it should in any case be emphasized that in this second model a greater number of features are made available than in the previous model. For example, by subscribing to the *Essentials* option, there will be more features than purchasing the *Base* package. At the time of writing, IC has chosen to adopt the release of ISE 2.7 and therefore to use the first licensing model. As mentioned later in the Section 8.2.2, the posture function is not of primary importance (i.e., it does not introduce fundamental improvements, especially in relation to business productivity). Therefore, by choosing not to adopt it, the cost of the *Apex* package would be canceled.

## Risk

The management and monitoring of the NAC system is a rather critical operation due primarily to the fact that user access to the network is related to it. An anomaly in the behavior of the system could result in a partial, or in the worst case a total, restriction on access to the network. It is easy to understand how this disservice would be projected, without any filter, towards the end user/device. For this reason, as mentioned in the previous paragraph, it is necessary to devote a qualified figure to this type of service who is able (in all cases) to respond positively to any incidents.

Furthermore, when this technology is offered in the form of a service (as mentioned in the previous paragraph), it is necessary to consider that the coverage area of the disservice expands from the InfoCamere site to only all the entities that have purchased it. Also in this case, the timeliness of incident response is fundamental.

### 8.2.2 Feasibility analysis

The feasibility analysis of a Proof Of Concept (**POC**) is the basic component of the project itself. The final solution to be implemented in a production environment is therefore analyzed below.

Numerous critical issues emerged during the various configurations and deployments. The latter mainly related to posture functionality. It is true that this feature allows the network administrator to know exactly the compliance status of the device, but is it really so essential? In fact, the scanning of a device must be strongly justified as the management of this system is not so trivial. In addition to the management for which a monitoring activity must be required, the user experience is also of vital importance. It is necessary to identify the correct trade-off between the level of security and usability. Along with SOC,

posture functionality has been classified as unnecessary. Current security tools allow to provide good coverage, making a posture client not strictly necessary. Different consideration regarding the profiling functionality have been made. It requires more attention during the first phase of configuration and deployment. Once adopted, the attention to devote to it turns out to be less than that paid to the posture.

Therefore, taking into account the need to prepare a specialized figure for monitoring it, and the non-obligatory nature of implementing posture function, it is possible to consider the final POC presented as a feasible solution.

### 8.2.3 Future work

The implementation described in the previous chapters aims to improve the one currently in production in InfoCamere. However, during the analysis carried out for the deployment in the test environment, innovative solutions were identified that can be applied in the future. These features are illustrated below.

#### DTLS

As shown several times in the previous chapters (e.g., from the captures made using Wireshark) the RADIUS packets are transmitted in clear text (except for the password attribute) from the NAD to the NAC server. Taking into consideration the types of attacks described in Chapter 2 it is possible to understand how leaving the traffic in clear text is not a good solution. Before illustrating a possible solution to this problem, however, it is necessary to analyze the context of analysis. First, RADIUS traffic is exchanged between the access switch (NAD) and the machine where ISE (PSN) is installed. As regards the latter, it is located within the DC (like all the machine that provide services). From this point of view it is highly unlikely that an attacker will be able to enter the DC, passing all the security checks and apply any modification. The attention must therefore be placed on the other side, which is the NAD. The latter is certainly more vulnerable than the previous one, at least from the point of view of physical access. The attacker could try to perform 2 types of attacks:

- Modify the configuration by creating a SPAN interface;
- Groped to carry out a MITM attack by placing an ethernet tap (or similar appliance) in the uplink connection towards the core layer.

While for the second case only a change of state of the uplink interface would occur (thus causing a disservice to all users connected to it), in the first case there is an important impediment: to access the device's console it is necessary to know both the *login* and *enable* password. Although the unencrypted transmission of sensitive information is obviously a bad practice (consider the increasingly limited use of the Telnet protocol for remote management), the complexity of accessing such information is not trivial. The solution of using the DTLS protocol is therefore not as necessary as it might seem at first sight [29]. Furthermore,

the configuration of this feature introduces an additional level of complexity to management of the entire system. Its use is therefore recommended but not necessary.

To enable it, it is necessary to follow 2 steps: one within the ISE, while the other within the NAD. Figure 8.2 shows the interface to enable DTLS when defining the NAD within ISE. Note how the L4 ports change depending on the

The screenshot displays the 'RADIUS Authentication Settings' configuration page. It is divided into three main sections:

- RADIUS UDP Settings:**
  - Protocol: RADIUS
  - \* Shared Secret: [Redacted] [Show]
  - Use Second Shared Secret:  [i]
  - [Redacted] [Show]
  - CoA Port: 1700 [Set To Default]
- RADIUS DTLS Settings [i]:**
  - DTLS Required:  [i]
  - Shared Secret: radius/dtls [i]
  - CoA Port: 2083 [Set To Default]
  - Issuer CA of ISE Certificates for CoA: Select if required (optional) [i]
  - DNS Name: [Empty field]
- General Settings:**
  - Enable KeyWrap:  [i]
  - \* Key Encryption Key: [Redacted] [Show]
  - \* Message Authenticator Code Key: [Redacted] [Show]
  - Key Input Format:  ASCII  HEXADECEIMAL

Figure 8.2: DTLS configuration interface.

protocol used. Subsequently it is necessary to create a certificate to be imported within the NAD. In fact, to create a secure tunnel, both the NAD and ISE must exchange certificates with each other. Therefore, from the access network device it will be necessary to generate a certificate to be imported into ISE. Moving on to NAD, the rest of the configuration of the RADIUS settings remains similar, apart from a few commands needed to enable the DTLS. By performing a packet capture it will be possible to see how the establishment of a TLS tunnel takes place at the beginning, while afterwards the actual exchange of encrypted RADIUS packets takes place.

## TEAP authentication

All authentication methods have been illustrated in Chapter 5. The main problem with these methods lies in the correlation between machine and user authentication. In fact, to apply this condition it was necessary to use Cisco AnyConnect NAM with EAP-FAST. There was no native support for this feature. In the implementation present in production, and the one developed within this



thesis, the assessment of the device and of the user at the same time has not been used. This means that a user could use a personal device to log into the network, using his domain accounts. With the release of *Windows 10 2004* in May 2020, for all machines with this OS installed, the TEAP method is natively supported. As for ISE, it is supported from version *2.7* onwards. Through TEAP method and the use of the feature called “*EAP chaining*” it is therefore possible to increase the level of security. Through this implementation, only users defined within the Domain Controller who use a company device can access the network correctly. Furthermore, the fact that this method is natively present within the *Windows* OS, opens up the possibility of creating a GPO that integrates it. The conceptual steps to follow to perform a deployment based on this method are as follows:

- 1 Create a GPO such that the selected authentication method is “*EAP-TEAP*”;
- 2 Properly configure the TEAP method within ISE.

As regards the second point, the main steps to be carried out are 2: enable the TEAP method within the list of permitted protocols, and consequently redefine the authorization policies. In Figure 8.3 the interface that satisfies the first point is shown. Notice how the “*Enable EAP Chaining*” option is enabled. The second step is instead shown in Figure 8.4. Within the policy set it is therefore possible to use the condition “*EapChainingResult*”. This condition can assume the 3 indicated values. To obtain a maximum authentication level it is recommended to select “*User and machine both succeeded*”. This results in the behavior described above. It should also be specified how the possibility of using different values

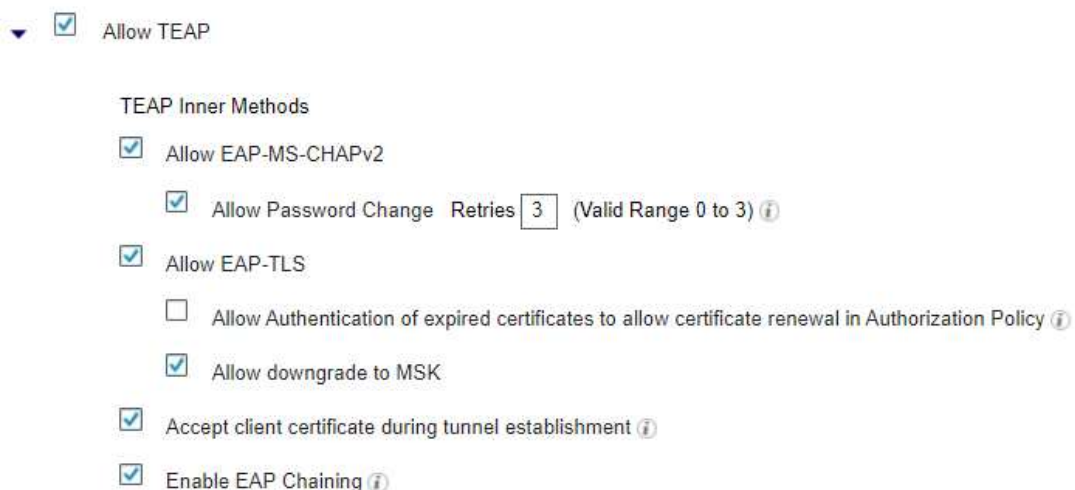


Figure 8.3: EAP-TEAP configuration in ISE.

for this condition allows the network administrator to manage multiple scenarios with different solutions. The authorization level, for example, can therefore be

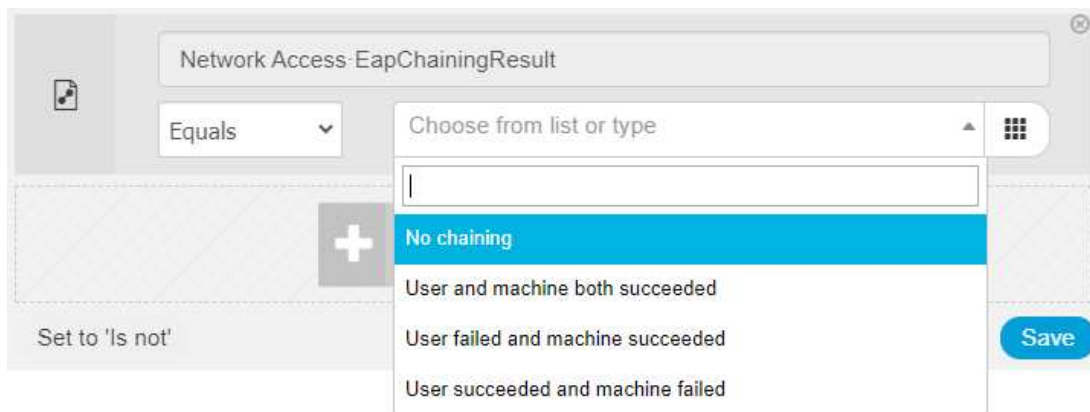


Figure 8.4: EAP-TEAP configuration within policy set.

different depending on whether the device used to access the network is registered within the domain (i.e., it is owned by the company) or not.

### Profiling applied to corporate devices

At Chapter 6 the profiling functions were described in detail, and their implementation through the policy set. To maximize the safety level, also in this case it is necessary to apply an additional feature. In fact, the main problem of the solution presented lies in identifying whether the device belongs to the corporate or not. While with 802.1X it is possible to use the TEAP method, with MAB this solution is not available. In case an attacker connect a device to the IC internal network, it would be considered as a corporate device. This is because it would authenticate via MAB and would therefore be considered as a legitimate device. A solution that can solve this problem is the census of corporate devices. Each device, purchased through a commercial channel (e.g., CONSIP) is initially registered within an external AD to which ISE is joined. Subsequently, within the policy set, more precisely within the MAB authorization policies, this AD is used to confirm that the devices belong to the corporate asset. when registering the MAC addresses of the devices (used as ID of the devices), it is recommended to adopt a hierarchical organization. In this scenario it would be possible to use a higher level of granularity within the policy set.

### Private VLAN

An extension of the VLAN concept presented in the initial chapter are the private VLANs [30]. Before explaining their use cases, their behaviour is shown below (refer also to Figure 8.5). Private VLANs are basically composed of 2 components, namely primary and secondary VLANs. The primary VLAN is the container VLAN, which contains all other VLANs. The secondary VLAN is instead the one that is located within the primary. These secondary VLANs are therefore associated with a single primary VLAN. An addressing is associated with the primary, e.g., the subnet 192.168.1.0/24 is associated with VLAN 100).

This therefore means that all the secondary VLANs associated with the primary one, share the same addressing space. Therefore, referring to Figure 8.5, a device associated with VLAN 300 has the same network address as one associated with VLAN 400. At this point, 2 types of secondary VLANs are defined: isolated and community. While for the first type, each port associated with it cannot communicate with anyone outside the dock interface, in the second one the devices can communicate with each other (but always within the same VLAN community). So, if a device belonging to the Isolated VLAN 200 wants to communicate with a device of the Community VLAN 300, it cannot. The previously mentioned dock interface is the only interface that can be reached by any other one, regardless of which VLAN it is associated with. The dock therefore consists of an uplink interface to the gateway.

Therefore, the devices that are destined for the Remediation VLAN can be associated with an Isolated VLAN. This way *Non-Compliant* devices cannot communicate with each other, removing the possibility that they can spread any kind of malicious entity.

This solution could therefore be implemented through dynamic templates with ISE. Another solution that leads to a similar result is the use of Cisco Trust-Sec feature. However, both of these 2 solutions require additional configurations which however, correspond to careful troubleshooting analysis during the testing phase.

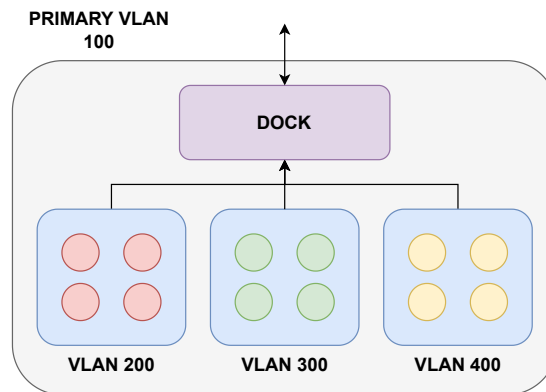


Figure 8.5: Private VLAN diagram.

### Posture agentless

With the 3.X release of ISE, a new type of agent posture has been introduced, namely “*agentless*”. This agent can be used on all devices where the installation of an application is not feasible. However, to allow ISE to evaluate the posture status it is necessary to enable, within the Windows Defender Firewall, the TCP connection towards TCP port 5985 to access the PowerShell. It is therefore necessary to enable the *PowerShell Remoting* feature on the endpoint. Port 5985 is defined by default for remote PowerShell access by Microsoft. It is however possible to define another port to be used within ISE to execute posturing scripts.

In environments such as InfoCamere, such configurations can once again be made using GPOs.

### **8.3 Conclusion**

In this thesis the 3 main functions related to a NAC system have been illustrated. Various types of deployments were presented, starting first from an open approach up to the exact opposite (i.e., closed mode). After an in-depth analysis of the protocols, an analysis was carried out with respect to which of the most widely adopted solutions in the enterprise environment is better. From this comparison it emerged that the solution, currently already implemented, is Cisco ISE. Subsequently, analyses were carried out with respect to the feasibility of the deployment of the proposed final solution. From this, it has been jointly established with the SOC that not all features studied are strictly necessary. The posture, despite increasing the level of security, introduces a not negligible management complexity. Furthermore, its deployment is strictly related to the user interface. Profiling, on the other hand, introduces automation concepts with which it is possible to exploit context visibility also for statistical purposes. In fact, it is possible (also on behalf of the chambers of commerce) that information relating to the types of devices that connect to the network is requested. With the several advanced features introduced in Section 8.2.3, the NAC system can be further improved.

# Index

AAA, 27  
ACE, 15  
ACI, 46  
ACK, 5  
ACL, 15  
AD, 27, 36  
AES, 80  
AM, 24  
API, 4  
ARP, 21  
ASIC, 28  
ASP, 123  
AUP, 22  
AVP, 53  
  
BGP, 4  
BYOD, 12  
  
CA, 44  
CCIAA, 42  
CERVED, 44  
CF, 135  
CHAP, 74  
CLI, 32  
CoA, 33  
CP, 61  
CPP, 155  
  
dACL, 17  
DC, 9, 123  
DDOS, 9  
DES, 76  
DHCP, 4  
DLP, 9  
DNS, 4  
DR, 44  
  
DSA, 82  
DTLS, 57  
DTP, 67  
  
EAP, 21  
EAPoL, 64  
EAPoR, 70  
ECDH, 82  
ECDSA, 82  
ERS, 168  
EW, 123  
  
FAST, 79  
FQDN, 128  
FTP, 4  
FW, 8  
  
GPO, 78  
GRE, 56  
GTC, 72  
GUI, 32  
  
HA, 45  
HIPS, 15  
HTTP, 4  
  
IC, 29, 42  
IDS, 9  
IPS, 9  
IPv4, 5  
IPv6, 5  
IS-IS, 6  
ISE, 29  
ISO, 3  
ISP, 9  
  
LACP, 45

- LAN, 9
- LLC, 6
- LLDP, 67
- LLDP-MED, 127
  
- MAB, 65
- MAC, 6
- MDM, 169
- MIB, 131
- MITM, 10
- MnT, 30
- MPLS, 45
  
- NAC, 3
- NACK, 5
- NAD, 25
- NAS, 25
- NBA, 15
- NGFW, 8
- NIC, 48
- NIPS, 14
  
- OID, 131
- OSI, 3
- OSPF, 4
- OTP, 72
- OUI, 68
  
- PAC, 79
- PACL, 17
- PAE, 61
- PAK, 169
- PAN, 30
- PCI, 4
- PDU, 4
- PEAP, 77
- PKI, 80
- PMS, 25
- PNAC, 21
- POC, 170
- PPTP, 56
- PRA, 20
- PRF, 80
- PSN, 30
  
- PUE, 44
- PXE, 98
- PXG, 30
  
- RADIUS, 29
- RIP, 6
- RSA, 82
- RSPAN, 129
- RTP, 107
- RTT, 36
  
- SD-WAN, 45
- SDU, 4
- SHA, 80
- SIP, 107
- SNMP, 4
- SNS, 30
- SP, 15
- SPAN, 129
- SPIF, 33
- SSH, 4
- SSL, 4
- STP, 67
  
- TACACS+, 57
- TCP, 4
- TCP/IP, 4
- TEAP, 80
- TFTP, 13
- TLS, 4
- TLV, 53
  
- UDP, 5
- UP, 61
  
- VIP, 128
- VLAN, 9
- VPN, 145
- VSA, 117
- VTP, 56
  
- WAF, 13
- WAN, 9
- WIPS, 15
- WLC, 27

# Bibliography

- [1] Jon Postel. *Internet Protocol*. RFC 791. Sept. 1981. DOI: 10.17487/RFC0791. URL: <https://www.rfc-editor.org/info/rfc791>.
- [2] Andrew S. Tanenbaum. “Network Protocols”. In: *ACM Comput. Surv.* 13.4 (Dec. 1981), pp. 453–489. ISSN: 0360-0300. DOI: 10.1145/356859.356864. URL: <https://doi.org/10.1145/356859.356864>.
- [3] Gerald A Marin. “Network security basics”. In: *IEEE security & privacy* 3.6 (2005), pp. 68–72.
- [4] Merike Kaeo. *Designing network security*. Cisco Press, 2004.
- [5] Eric Cole. *Network security bible*. John Wiley & Sons, 2011.
- [6] Omar Santos. *CCNP and CCIE Security Core SCOR 350-701 Official Cert Guide: Implementing and Operating Cisco Security Core Technologies*. Cisco Press, 2020.
- [7] Andy Richter and Jeremy Wood. *Practical Deployment of Cisco Identity Services Engine (ISE): Real-world Examples of AAA Deployments*. Syngress, 2015.
- [8] Craig Hys. *Designing ISE for Scale & High Availability*. Cisco LIVE, 2018. URL: <https://www.ciscolive.com/c/dam/r/ciscolive/emea/docs/2018/pdf/BRKSEC-3699.pdf>.
- [9] Allan Rubens et al. *Remote Authentication Dial In User Service (RADIUS)*. RFC 2865. June 2000. DOI: 10.17487/RFC2865. URL: <https://www.rfc-editor.org/info/rfc2865>.
- [10] Craig Finseth. *An Access Control Protocol, Sometimes Called TACACS*. RFC 1492. July 1993. DOI: 10.17487/RFC1492. URL: <https://www.rfc-editor.org/info/rfc1492>.
- [11] “IEEE/ISO/IEC International Standard-Telecommunications and exchange between information technology systems—Requirements for local and metropolitan area networks—Part 1X:Port-based network access control”. In: *ISO/IEC/IEEE 8802-1X:2021(E)* (2021), pp. 1–292. DOI: 10.1109/IEEESTD.2021.9650828.
- [12] “IEEE/ISO/IEC International Standard-Telecommunications and exchange between information technology systems—Requirements for local and metropolitan area networks—Part 1X:Port-based network access control”. In: *ISO/IEC/IEEE 8802-1X:2021(E)* (2021), pp. 1–292. DOI: 10.1109/IEEESTD.2021.9650828.

- [13] John Vollbrecht et al. *Extensible Authentication Protocol (EAP)*. RFC 3748. June 2004. DOI: 10.17487/RFC3748. URL: <https://www.rfc-editor.org/info/rfc3748>.
- [14] Taesub Kim et al. “Designs of a Secure Wireless LAN Access Technique and an Intrusion Detection System for Home Network”. In: Oct. 2008, pp. 318–324. ISBN: 978-0-7695-3322-3. DOI: 10.1109/NCM.2008.46.
- [15] Daniel Simon, Ryan Hurst, and Bernard Dr. D. Aboba. *The EAP-TLS Authentication Protocol*. RFC 5216. Mar. 2008. DOI: 10.17487/RFC5216. URL: <https://www.rfc-editor.org/info/rfc5216>.
- [16] Glen Zorn and Steve Cobb. *Microsoft PPP CHAP Extensions*. RFC 2433. Oct. 1998. DOI: 10.17487/RFC2433. URL: <https://www.rfc-editor.org/info/rfc2433>.
- [17] Glen Zorn. *Microsoft PPP CHAP Extensions, Version 2*. RFC 2759. Jan. 2000. DOI: 10.17487/RFC2759. URL: <https://www.rfc-editor.org/info/rfc2759>.
- [18] William A. Simpson. *PPP Authentication Protocols*. RFC 1334. Oct. 1992. DOI: 10.17487/RFC1334. URL: <https://www.rfc-editor.org/info/rfc1334>.
- [19] Alan DeKok. *Extensible Authentication Protocol (EAP) Session-Id Derivation for EAP Subscriber Identity Module (EAP-SIM), EAP Authentication and Key Agreement (EAP-AKA), and Protected EAP (PEAP)*. RFC 8940. Oct. 2020. DOI: 10.17487/RFC8940. URL: <https://www.rfc-editor.org/info/rfc8940>.
- [20] Vivek Kamath, Ashwin Palekar, and Mark Wodrich. *Microsoft’s PEAP version 0 (Implementation in Windows XP SP1)*. Internet-Draft draft-kamath-pppext-peapv0-00. Work in Progress. Internet Engineering Task Force, Oct. 2002. URL: <https://datatracker.ietf.org/doc/draft-kamath-pppext-peapv0/00/>.
- [21] Ashwin Palekar et al. *Protected EAP Protocol (PEAP) Version 2*. Internet-Draft draft-josefsson-pppext-eap-tls-eap-10. Work in Progress. Internet Engineering Task Force, Oct. 2004. URL: <https://datatracker.ietf.org/doc/draft-josefsson-pppext-eap-tls-eap/10/>.
- [22] Hao Zhou et al. *Tunnel Extensible Authentication Protocol (TEAP) Version 1*. RFC 7170. Apr. 2014. DOI: 10.17487/RFC7170. URL: <https://www.rfc-editor.org/info/rfc7170>.
- [23] Eric Rescorla. *The Transport Layer Security (TLS) Protocol Version 1.3*. RFC 8446. Aug. 2018. DOI: 10.17487/RFC8446. URL: <https://www.rfc-editor.org/info/rfc8446>.
- [24] Eric Rescorla and Tim Dierks. *The Transport Layer Security (TLS) Protocol Version 1.2*. RFC 5246. Aug. 2008. DOI: 10.17487/RFC5246. URL: <https://www.rfc-editor.org/info/rfc5246>.



- 
- [25] Jon Postel. *Transmission Control Protocol*. RFC 793. Sept. 1981. DOI: 10.17487/RFC0793. URL: <https://www.rfc-editor.org/info/rfc793>.
- [26] Aaron Wolandm and Katherine McNamara. *CCNP Security Identity Management SISE 300-715 Official Cert Guide*. Cisco Press, 2020.
- [27] Gopal Dommety et al. *Dynamic Authorization Extensions to Remote Authentication Dial In User Service (RADIUS)*. RFC 5176. Jan. 2008. DOI: 10.17487/RFC5176. URL: <https://www.rfc-editor.org/info/rfc5176>.
- [28] Murtaza Chiba et al. *Dynamic Authorization Extensions to Remote Authentication Dial In User Service (RADIUS)*. RFC 3576. July 2003. DOI: 10.17487/RFC3576. URL: <https://www.rfc-editor.org/info/rfc3576>.
- [29] Eric Rescorla and Nagendra Modadugu. *Datagram Transport Layer Security Version 1.2*. RFC 6347. Jan. 2012. DOI: 10.17487/RFC6347. URL: <https://www.rfc-editor.org/info/rfc6347>.
- [30] Marco Foschiano and Sanjib HomChaudhuri. *Cisco Systems' Private VLANs: Scalable Security in a Multi-Client Environment*. RFC 5517. Feb. 2010. DOI: 10.17487/RFC5517. URL: <https://www.rfc-editor.org/info/rfc5517>.