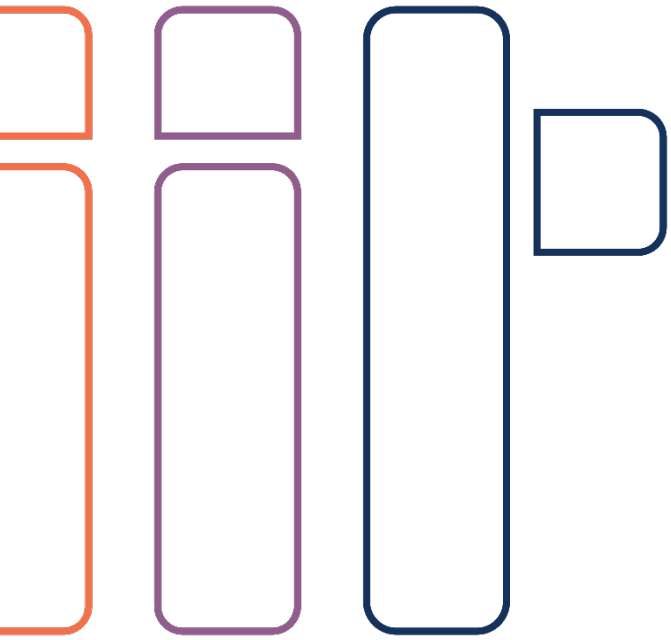

TECHNICAL REPORT

IIT TR-03/2022

A mechanism to monitor user access control on complex wireless network systems

C. Porta

ISTITUTO DI INFORMATICA E TELEMATICA

Consiglio Nazionale delle Ricerche

A mechanism to monitor user access control on complex wireless network systems

claudio.porta@iit.cnr.it

Computer and Communication Networks
Istituto di Informatica e Telematica - Consiglio Nazionale delle Ricerche
via G. Moruzzi, 1 - 56124 Pisa, Italy

Abstract

This document presents a solution for monitoring user access on wireless networks, combining the native protocolos and functionalities of the principal network devices that normally compose a complex wireless telematics infrastructure (access points, controller Wi-Fi and firewall). The result achieved is the correlation of network access data, mainly IP address and login name related to users authenticating on a given WLAN [1]. This derived information is useful especially for network administrators, because it enables the management of high level-application security policies related to IDS/IPS [2] that are suitable for today's evolving network security requirements.
Finally, the document describes a practical representation of the proposed solution, which has been implemented and it is still used to monitor the user's network accesses in the wireless networks of the CNR Research Area in Pisa.

Keywords: Network monitoring, Wireless Network, User Access Control, Intrusion detection system, intrusion prevention system.

Introduction

The constant evolution of telematic networks and their increasing pervasiveness in the daily routine, arouse a better attention concerning their security aspects. They have to be rigorous in all aspects, regarding the protection of network traffic and its related data, but also monitoring the users that access in a network. In fact, knowing the identity of those who login and navigate in a network is useful for several reasons, such as avoiding unauthorized access, understanding if someone is using illicit applications, detecting someone that may have transmitted a threat or simply to refine the IDS/IPS security policies [2] of the system.
The work described in this paper has to be classified in this context. It describes a solution to identify user access on wireless networks, combining the information and functionalities offered by devices that are typically part of a wireless telematics infrastructure, such as Access Points (APs) ,the related Wireless Controller [3] for their centralized management, and a perimeter Firewall, usually of Next Generation (NG) type [4] capable of monitoring both intranet and outbound traffic.
In order to understand the proposed solution, it is necessary to keep in mind how a typical telematics network generally works and its main components. The traffic coming from the Internet transits in a gateway router and is forwarded to a perimeter firewall, which monitors IPv4 [5] and IPv6 [6] pass-through traffic.
The data flow is then forwarded to the internal network, where other components such as switches or internal routers route the traffic to users end devices.

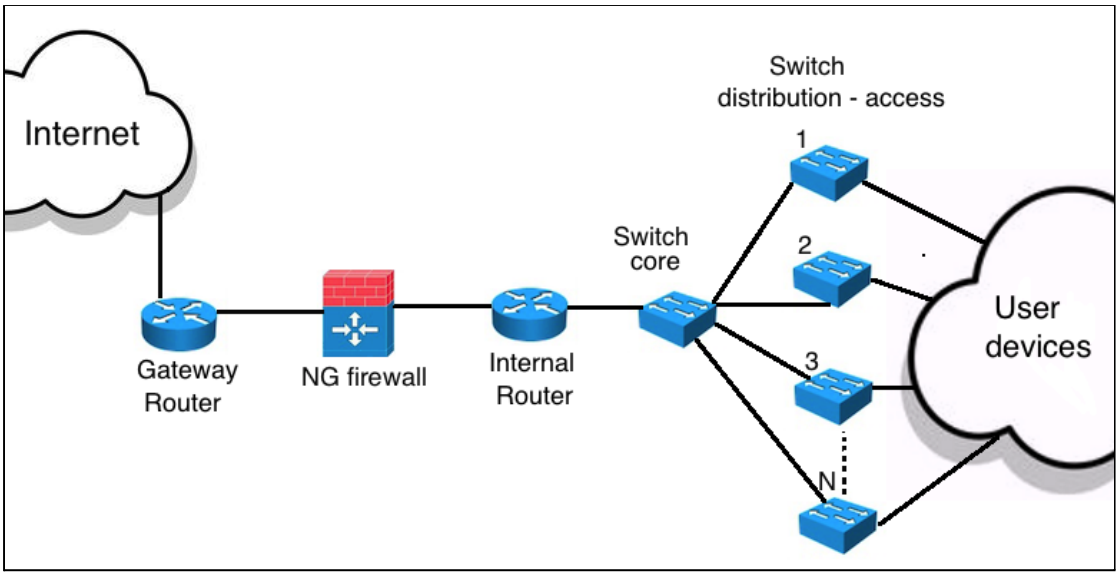

Figure 1: simplified diagram of a typical telematic network

All the network traffic incoming and outgoing transits normally through a firewall, which is used to detect threats or vulnerabilities identifying and classifying the affected devices with their IPv4 address.
In fact, when a user completes the authentication procedure to access a network, his device gets an IPv4 address that can be used for its identification. Things are a bit more complicated when using IPv6, because a device can have multiple addresses, but the concept remains the same.

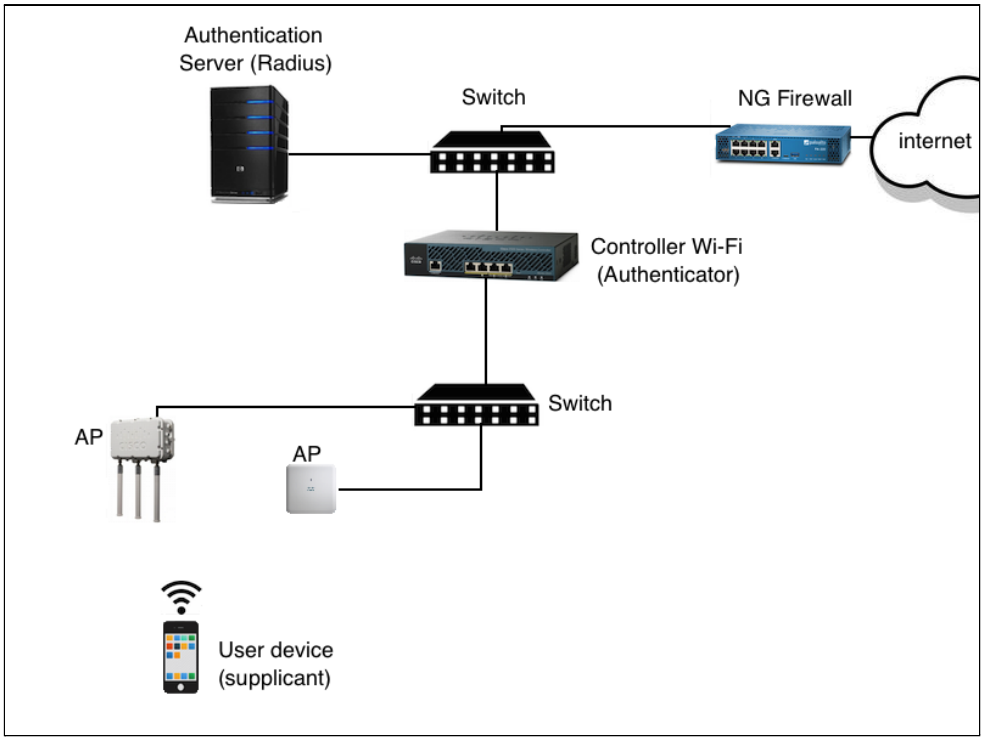

Figure 2: A network topology with the main components used in the authentication process to access the network.

Wireless network access monitoring

The main difference between cabled and wireless network is that in the latter case users do not have sockets and cables to plug in, but they login and navigate on a network through wireless access points.

Overview of 802.1X authentication mechanism

Generally , medium and large non-domestic wireless networks have distributed Access Points that cover an area with wireless signal, and these access points are centrally managed by a Wi-Fi Controller. The network authentication process to access can be implemented in different ways according to the chosen implementation.
One of the most common solutions refers to the IEEE 802.1X standard [7], which is based on port management and uses a server to store user credentials, in which a generic user can access the network through the use of personal credentials, typically username and password. The whole authentication procedure can be synthetically summarized as follows:

- a user device, which can be a smartphone, a tablet or a laptop, tries to connect to a WLAN [1] via the SSID [8] announced by a network access point. The SSID [8] represents the name of the WLAN and contains information needed to access the network, including the type of credentials required.
- The access point, managed by the Wi-Fi controller, sends a request to the client and sets up a secure channel on a port that allows the client to communicate only with an authentication server installed on the network, usually a RADIUS server [9], while all other traffic is blocked.
- The client sends authentication credentials to the RADIUS server [9] over the secure channel. If the authentication is successful, the RADIUS server [9] sends the configuration parameters to the client. After the procedure succeeds , the client can start to navigate on the network, using an encrypted communication segment between access point and client.

The problem

Once the authentication process is completed, the device's traffic flow is monitored by the perimeter next generation (NG) firewall, which detects possible threats or vulnerabilities identifying the devices according to their IP address.
From this assumption arises the problem discussed regarding the monitoring of network traffic: we know the IP address that identifies a device, but we do not know the association between a user and a device, and also its position.
Providing an example, if a device is infected by malware, in order to solve the problem as well as the IP address we also need to know who is using that device and in some cases even its position.
The user credential data is stored in the authentication server, while the IP and MAC address of the device's data flow is monitored by the NG Firewall. How to correlate the two information, in order to detect which IP address corresponds to which user, and moreover to create in the firewall rules based also on user credentials and monitor in real time the wireless user access control on network?

The proposed solution

A possible and cheap solution, using the features provided by the previously cited networking devices, is to exploit the SNMP protocol functionalities [10], configuring it on the Wi-Fi controller, and using in combination the remote communication interfaces API [11] normally used in a next generation firewall to interact with other applications.
In fact, the controller is capable of detecting events on a network, including if a user is accessing a network and its related information, (e.g. its login and the device's assigned IP).
In addition, most of the new generation firewalls use APIs for integrating external data and functionalities with other devices, including receiving information about users

accessing the network, useful for the correlation of data already available at a lower level, like device's IP and MAC address [12].
SNMP is an application layer protocol that provides the exchange of management information between network devices. The system defined by SNMP protocol consists of three main elements:

1. an **SNMP manager**, a software that communicates with SNMP agents of the same community configured on other networking devices. The SNMP manager can query SNMP agents, modify their variables, receive responses to queries, and detect asynchronous events sent by the agents;
2. one or more **SNMP agents**, a program configured in a networking device, which if enabled store all the device's status information in a dedicated database, or send to the SNMP manager of the same community the occurrence of a particular event. The type of message used for events notification is called **TRAP** [13];
3. the **Management Information Base** (**MIB** [14]), a database consisting of objects in which the SNMP agent stores information about the managed device. The MIB [14] is also used by the SNMP manager to request information from the SNMP agents.

Summarizing, a networking device with an SNMP agent configured has the ability to intercept events and to serialize the information of the significant ones into particular messages called **TRAP**s [13], which can then be sent to the SNMP Manager in the same community of the SNMP agents.
TRAPs [13] are composed of nodes, implemented as objects identified by **Object IDentifiers** (**OID**s) [14], sequences of integers and dots that uniquely identify managed objects that are defined in the MIB. The communication between agent and manager SNMP is established using 161 and 162 UDP ports. The first is used by the manager to send requests to agent, and by the agent to respond to requests, while the latter is used by the manager to receive TRAP messages or notifications sent by the agent.

Bringing these notions back to the case of user access control on wireless networks, it is possible to configure and enable an SNMP agent on the network Wi-Fi controller and use it to detect the user's access network events, with the aim to forward the TRAPs related to those events to another device where an SNMP manager is configured.
The SNMP Manager does not make requests to the SNMP agents, but simply remains waiting for receiving TRAP messages from the SNMP agent activated on the Wi-Fi controller. When a TRAP is received, the programm parses the information contained and serializes it in another message that can be sent to the Firewall using its exposed APIs.
In fact, the SNMP protocol messages differ from that used for communication via API interfaces, which is generally in XML format [15].
Then, after the activation and configuration of an SNMP agent in the Wi-Fi controller , it is moreover necessary implement a software in a network device (e.g a server) that perform the following operations:

- acts as an SNMP manager, receiving using a secure connection the TRAPs sent by the SNMP agent of the Wireless controller when a user access on network occurs;
- parses sequentially all the received TRAP messages, with the aim to retrieve the following information:
    - The login name of the user who accessed the network. .
    - The IP address assigned to the user's device.
    - The network domain that has been accessed.

  The part of MIB that deals with the network user access information is not standard, but implemented by different manufacturers privately. Consequently, the OIDs that identify the objects within the TRAP message are different and based on the brand of the device on which the SNMP agent is installed. Therefore, it is necessary to implement different message parsing modes, based on the MIB of the device on which the SNMP agent is installed.

- The information retrieved is then serialized in an XML message and forwarded through remote API invocation to the next generation firewall.

When the firewall receives the XML message sent by the API, it correlates the information retrieved with the data already available that are IP and MAC addresses. The new set of correlated information can then be used to create ad hoc high-level rules on applications, or simply to monitor user access on the network.

The below image represents an overview of the whole process, distinguishing three main steps:

1. a user device logs in on network;
2. the event is detected by the SNMP agent configured on the Wi-Fi controller, which sends a TRAP message with the user's information using a secure connection to communicate with the SNMP Manager installed on a network server;
3. The SNMP manager receives the TRAP, containing the username of the user who logged on a network, his device's IP address and the network domain he logged in. The information is put into an XML message and sent to the firewall through an API request.

The Firewall receives the information and automatically performs a data correlation with the data already available gathered from the network traffic analysis. After this operation, all the IP addresses used by wireless devices through users authentication are associated with the username credential and the WLAN accedeed. This correlation can be viewed on the graphical Firewall management interface.

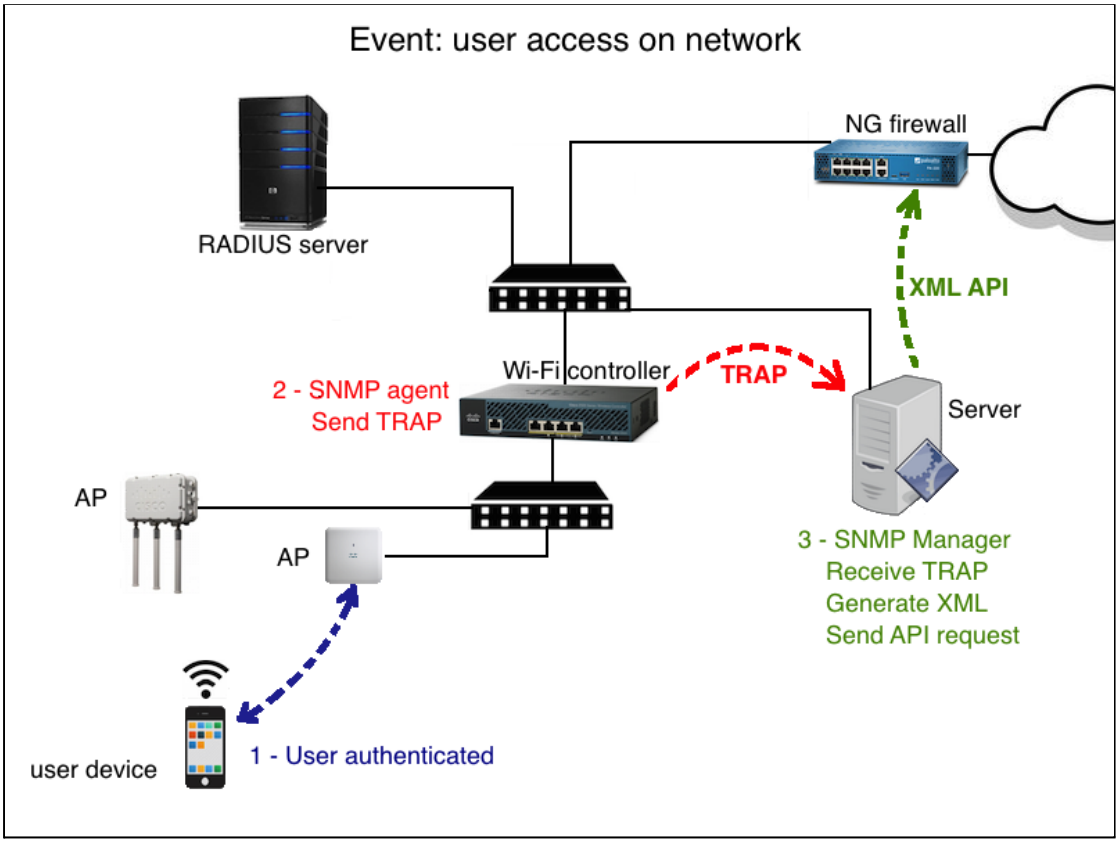

Figure 3: Graphical representation of the detection of a user access on a wireless network.

Implementation of the solution: a practical case

An example that describes the effectiveness of the solution is its practical implementation to monitor the user access on the wireless telematics network of the Pisa Research Area Campus, an infrastructure shared by 15 organizations and composed of:

- about 160 internal and external Access Points of various Cisco technology models;
- two Cisco WLC 5520 wireless controllers for centralized AP management;
- a couple of next generation perimeter firewalls model Palo Alto PA-5520.

In the research area campus different WLANs are announced. Those reserved for area's staff use the 802.1X authentication protocol [7] to grant a secure connection with custom credentials. To provide internet connection to guests, there is also a dedicated WLAN, which implements a Web-Based authentication. In this WLAN, when a user guest tries to access the network, a web page in his device's browser appears, and it is redirected to an authentication page managed by a captive portal. More information about this implemented authentication type can be found in [16] but in the context of this work, there is only to keep in mind that there is a captive portal [17] based on MikroTik technology [18], used for the management of the authentication mechanism for guest users of the research area. .

The first operation carried out to implement the wireless access control system had been the configuration and activation of the SNMP agents on the Area WI-Fi controllers and on the captive portal, giving attention to specify the correct SNMP community and the IP address of the machine that would have host the SNMP manager that will have received the SNMP TRAP messages.

Once the SNMP agents were configured in the two controllers and in the captive portal, we also configured and activated the program that acted as SNMP Manager with the aim to parse the TRAPs received from the SNMP agents. The program is implemented as a background service of the host's operative system. In this way, it can be automatically restarted after a device reboot. It listens on 162 UDP port, waiting to receive SNMP TRAP messages sent from the agents configured on the WiFi controllers and captive portal each time that a user accesses the network. Every message is parsed in order to retrieve information about the users accessing the network, i.e. the network WLAN accessed, the user's login, his IP and the time of access.

The data parsed is then serialized in XML format and sent to the network firewall through the use of Web APIs made available by its operating system.

The connection to the firewall is established the first time sending through the web authentication API the credential access (login and password) of a custom profile created exclusively for the user-id management. After the first authentication succeeds, next authentications are performed using an encrypted key, generated and stored in a hidden folder during the first connection. A new connection occurs every time that a simple TRAP message information is parsed and ready to be sent to the firewall.

The program is also provided with a logging and email notification mechanism, so as to communicate any malfunctions and to have a report for debugging purposes.

Looking the overall program procedure, it is possible classify three different main phases:

1. The receival of SNMP TRAP messages
2. The analysis and parsing of the TRAP messages
3. The dispatch of the retrieved user's data to the network firewall

1. Receival of SNMP TRAP messages

TRAPs are received in input by a receiver configured as a passive listener on UDP port 162. Once received, they are serialized using a python open source library (pysnmp) [19] into an object-and-attribute structure suitable to be parsed using OID identifiers.

2. Analysis and parsing of TRAP messages

Once received in input and serialized, the TRAP message is ready to be parsed according to OIDs identifiers, selecting only those related to network access and authentication information. The OIDs related to this type of information differ according to the brand of the device sending the TRAPs, so different parsing must be adopted according to the brands of the devices. In detail, Cisco devices have a dedicated OID for each field IP address, username and access network, while the MikroTik device allows the custom insertion of all information in a unique field identified by specific OID for sending information.

From the TRAPs parsing the following information is obtained:

- The type of device that is sending the TRAP (Cisco or MikroTik), useful to differentiate the subsequent user access data parsing;
- The login ID of the user that is accessing the network;
- The IP address that have been assigned to the user after the authentication;
- The WLAN network that the user has accessed.

3. Sending the obtained data to the network firewall

Once the data is parsed, it is then forwarded to the network NG firewalls, using the APIs exposed by the firewall operating system. Another python open source library, PanXapi [20], has been used to implement the secure communication mechanism between our program and the Palo Alto firewall.

Another task performed for security reasons, has been the configuration on the firewall of a custom account with restricted permissions, used exclusively for receiving and memorizing through API invocation and after authentication, the information about the accesses of the users in the various Area wireless networks.

As mentioned above, the connection to the firewall is made by the application the first time by entering the account credentials, and once this operation is performed, an encrypted key is received and stored in a hidden folder. This key is then used for the next communication with the aim to send API messages in secure mode without using the authentication credentials again. In fact any further communication, related to a new user access on the network, will take place using the encrypted key and not the credentials of the profile configured on the firewall.

Conclusions

It is a given fact that monitoring user access on the network is important for several reasons, that start from the prevention and detection of unauthorized access on the network to the definition of high-level security rules based on applications classified by user groups.

After briefly summarizing a widely used authentication mechanism on wireless networks, 802.1X authentication, the document has introduced the problem that led to the implementation of this solution: how to integrate a user's information within network monitoring, correlating his access user-id with his IP address?

The proposed solution uses the functionality provided by firewall and network controller, integrating the user information available to the two devices through the use of SNMP protocol and Web API interfaces to share the datas.

Then it has been demonstrated how this solution can be implemented in practice at zero cost, providing a use case for monitoring user access in wireless networks in the Research Area of Pisa, successfully actually used for this purpose

A possible and similar future expansion of this work could also concern the correlation of user data in cabled data networks, integrating the available data already obtained from other monitoring applications recently developed, that are used to localize the devices inside the Research Area starting from their IP address and the structured cabling datas.

References

[1] - WLAN - Wireless Local Access Network
https://en.wikipedia.org/wiki/Wireless_LAN

[2] - IDP/IPS - Intrusion Detection System
https://en.wikipedia.org/wiki/Intrusion_detection_system

[3] - Wireless LAN Controller
https://en.wikipedia.org/wiki/Wireless_LAN_controller

[4] - Next Generation Firewall
https://en.wikipedia.org/wiki/Next-generation_firewall

[5] - Information Sciences Institute - University of Southern California. (1981).
RFC 791 - Internet Protocol - https://tools.ietf.org/pdf/rfc791.pdf

[6] - S. Deering, R. Hinden (1998). RFC 2460
Internet Protocol, Version 6 (IPv6) specification - https://www.ietf.org/rfc/rfc2460.txt

[7] - IEEE 802.1X - Remote Authentication Dial In User Service (RADIUS) -
RFC 3580 - https://datatracker.ietf.org/doc/html/rfc3580#ref-IEEE8021X

[8] - SSID - Service Set IDentifier
https://en.wikipedia.org/wiki/Service_set_(802.11_network)#SSID

[9] - RADIUS - Remote Authentication Dial In User Service
RFC 2865 - https://datatracker.ietf.org/doc/html/rfc2865

[10] - Case, J., Fedor, M., Schoffstall, M., and J. Davin, "Simple
Network Management Protocol", RFC 1157, SNMP Research,
Performance Systems International, Performance Systems
International, MIT Laboratory for Computer Science, May 1990 -
https://datatracker.ietf.org/doc/html/rfc1157

[11] - API- Application Programming Interface
https://en.wikipedia.org/wiki/API

[12] - next generation firewall, user-id API
https://docs.paloaltonetworks.com/pan-os/9-0/pan-os-admin/user-id

[13] - TRAP - Rose, M., "A Convention for defining traps for use with
the SNMP", RFC 1215, March 1991. - https://datatracker.ietf.org/doc/html/rfc1215

[14] - MIB - McCloghrie K., and M. Rose, "Management Information Base for
Network Management of TCP/IP-based internets", RFC 1155, Hughes
LAN Systems, Performance Systems International, May 1990.
https://datatracker.ietf.org/doc/html/rfc1155

[15] - XML - EXtensible Markup Language - https://www.w3.org/XML/

[16] - A. Mancini, A. De Vita, A. Gebrehiwot, M. Marinai IIT TR-02/2016
Wireless network federation for Smart Cities: an example of implementation in the Research Area of Pisa https://intranet.cnr.it/servizi/people/prodotto/download/i/110828

[17] - Captive Portal - https://en.wikipedia.org/wiki/Captive_portal

[18] - MikroTik RouterOS - http://download2.mikrotik.com/what_is_routeros.pdf.

[19] - pysnmp - https://pysnmp.readthedocs.io/en/latest/

[20] - PanXAPI - http://api-lab.paloaltonetworks.com/pan-python.html