

Port-Based Authentication using 802.1x

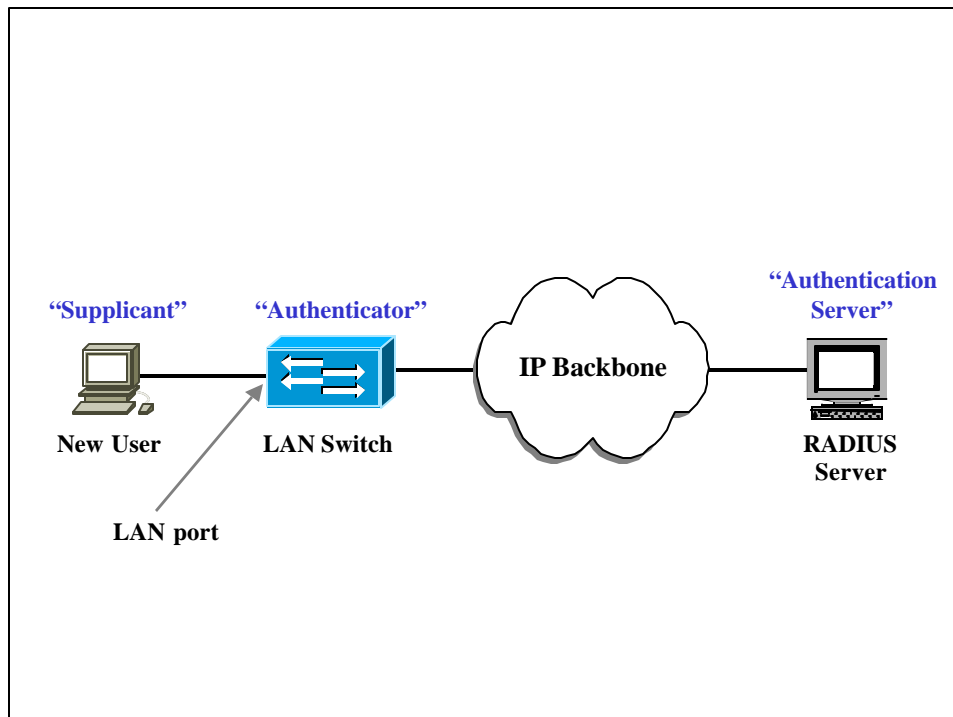
Authenticating Users at the LAN Switch Port using 802.1x

802.1x is a new standardized port-based authentication protocol defined by the IEEE “802 task-force” in 2001. (The “x” is part of the name of the standard) It is a method in which a LAN switch queries all new connections as soon as they are made, but prior to any IP connectivity being allowed through the switch.

This method of authentication is commonly used to control access to Wireless Access Points, but 802.1x is also supported on “wired” networks as well.

Cisco supports 802.1x on their Catalyst LAN switches as of CatOS 7.1 and Switch IOS 12.1. Network communication between the switch and client is made possible, prior to any IP connectivity, by the fact that the LAN switch encapsulates the queried credentials from the new computer within a special Ethernet frame, then marks that frame with a special “Ethertype” label, and forwards it to the RADIUS server over the network using normal IP routing. The packets sent between the switch and RADIUS server are encrypted. So 802.1x essentially acts as a special “tunnel” between the newly-connected computer and the RADIUS server, with the tunnel only partially-opened for the first phase of network-access.

802.1x breaks down the authentication process into 3 distinct elements, and gives each element its own name. Visually, the breakdown looks like this:



The 3 parts of this process are given the names “Supplicant”, which refers to the new computer trying to connect to the network, the “Authenticator”, which refers to the LAN switch that the computer is trying to connect to, and the “Authentication Server”, which refers to the RADIUS server located on the network-backbone.

The process basically involves the Authenticator (the LAN switch) asking each newly-connected PC for a set of credentials in the form of an “EAP Type”. EAP packets come in several different types, which must be defined on the RADIUS server:

- EAP-TLS
 - o TLS stands for “Transport Layer Security”. This uses Certificates to authenticate identity. The certificates must be managed on both the client and server side, with certificates installed on each workstation in order to maintain a PKI infrastructure. This is secure, but can be a challenge to manage as a network scales.

- EAP-TTLS
 - o TTLS stands for “Tunneled Transport Layer Security”. It was developed as an extension of EAP-TLS. It also uses Certificates, but are sent to the RADIUS server through an encrypted channel, or "Tunnel". Unlike EAP-TLS, EAP-TTLS requires only server-side certificates. Since the Certificate is tunneled, there is no need for a certificate on the client-side.

- EAP-LEAP
 - o LEAP stands for “Lightweight Extensible Authentication Protocol”. This is Cisco’s proprietary version of EAP, developed for use with their Aironet wireless products. It encrypts packets using dynamically-generated WEP keys, and supports mutual authentication. Although it was originally a proprietary format, Cisco has licensed LEAP to other manufacturers.

- EAP-PEAP
 - o PEAP stands for “Protected Extensible Authentication Protocol”. It supports legacy password-based protocols. PEAP tunnels between network-clients and the RADIUS server, like the competing standard TTLS. Also like TTLS, PEAP authenticates clients using only server-side certificates. It was developed by Microsoft, Cisco and RSA Security.

Required Client Software

The “Supplicant” is required to be running 802.1x client-software in order to reply to the query by the LAN switch. Without any such software, the client will be unable to respond to the query by the “Authenticator” and no connection will be possible.

Most current operating systems support either add-on 802.1x client-software, or have 802.1x-support integrated into the OS itself:

- Windows XP comes with support for 802.1x built into its networking software, and Windows 2000 systems support 802.1x if they are running at least Service Pack 3. This link at Microsoft.com describes how to use the 802.1x client software, using either the client built into Windows XP, or the add-on software to Windows2000:

www.microsoft.com/windows2000/server/evaluation/news/bulletins/8021xclient.asp

- This is a link for adding 802.1x support for Windows98:

http://downloadfinder.intel.com/scripts-df/Detail_Desc.asp?agr=N&ProductID=763&DwnldID=5763

- This link points to Open Source 802.1x client software for Unix/Linux clients:

www.open1x.org

- This is a link showing how to use the 802.1x client built into MacOS X “Panther”:

www.uic.edu/depts/accc/network/wireless/macx.html

Authentication Steps

Assuming a newly-connecting computer is running one of the above-mentioned 802.1x clients, the following describes what happens:

1. Once the client's NIC has established contact with the Switch-port, the client initially sends an "EAP Start" message, which initiates a series of message-exchanges between the client and the LAN switch.
2. The LAN switch then replies with an "EAP Request Identity" message, basically asking the client to identify itself. This will usually be in the form of either a system-name or a MAC address.
3. The client then replies with its identity, which the LAN switch encapsulates into an IP packet and forwards on to the RADIUS server. At this point, the LAN switch still does not allow any IP connectivity by the client, such as DHCP requests. The LAN-port is basically in a semi-blocked state, waiting for a reply from the RADIUS server.
4. The RADIUS server then runs some specific algorithm to verify the new client's identity. This will usually be some sort of digital certificate, or other similar EAP authentication type.
5. The RADIUS server will then send a message back to the LAN switch, in the form of either an "Accept" or "Reject" reply.
6. The LAN switch then forwards this reply back to the new client, in the form of an "EAP Success" or "EAP Reject" message. If it was accepted, the LAN-port is fully opened and DHCP requests, along with all other IP activity, is allowed. If it was rejected, the port is either shut down, or it is mapped to a specific, usually external, VLAN.

802.1x is basically just a delivery mechanism. It doesn't enforce any specific dynamic key management or EAP types, leaving this up to the RADIUS server and the department managing it.

This all takes place prior to the client making a DHCP request for an IP address. This requires that the RADIUS server reply in a timely manner. It is possible for the reply from the RADIUS server to delay enough that the client DHCP request times out. We would need to simulate this to establish what the tolerable wait-time would be.

Configuring 802.1x on Cisco Catalyst Switches

The following describes how to configure 801.x port-based authentication on Cisco switches, using either Switch IOS or CatOS.

Note: Cisco refers to 802.1x as "dot1x".

This is the base configuration required to support 802.1x on Cisco switch ports:

IOS Switch Config:

```

config terminal
aaa new-model (This enables AAA/Radius)
aaa authentication dot1x default group radius
  (The above creates an 802.1x "method list")
ip radius source-interface FastEthernet 0/0
  (The above defines the RADIUS source-address)
radius-server host 10.1.240.43 (This defines the RADIUS server address)
radius-server key MyKey123 (This defines the RADIUS authorization key)

dot1x system-auth-control (This "turns on" 802.1x authentication on the switch. This
  command is required as of Switch IOS 12.1(14), not prior IOS. )

```

interface range FastEthernet 0/0 – 48
switchport access vlan 5 (Defines the VLAN the specific ports belongs to)
switchport mode access (This turns off VLAN “dynamic” mode. No trunking allowed)
dot1x port-control auto (This enables 802.1x on specific ports)

dot1x guest-vlan 666 (Optional: This defines what VLAN to assign to clients that fail authentication 3 times. Otherwise, port closes.)

CatOS Switch Config:

set radius server 10.1.240.43 (This defines the RADIUS server address)
set dot1x system-auth-control enable (This “turns on” 802.1x authentication)
set port dot1x 0/1-48 port-control auto (This enables 802.1x on specific ports)

set port dot1x 0/1-48 auth-fail vlan 666 (This defines what VLAN to assign to clients that fail authentication 3 times)

The last line in the configurations defines what to do when a client fails authentication. The switch will allow 3 attempts before declaring the connection a failure. The switch will then prevent access to the default VLAN, and instead re-map the connection to a default VLAN, which standard policy usually defines as an external VLAN with limited network access.

The time between the 3rd authentication failure and VLAN re-assignment is **3 minutes**.

Show commands:

IOS Switch:
show dot1x ?
show dot1 statistics

CatOS Switch:
show port dot1x ?
show port dot1x auth-fail-vlan (Shows who failed authentication)

Cisco switches have the ability to periodically refresh 802.1x authentication, if required. This is a global configuration and cannot be defined on a per-port basis. It is enabled like this:

IOS Switch:

config terminal
dot1x reauthentication (This activates re-authentication)
dot1x timeout re-authperiod 4000 (This defines the timeout value, in seconds.
The default is 3,600 seconds, or 1 hour. The range of values is 1 – 4,294,967,295)

CatOS Switch: (can be enabled per-port)

set dot1x re-authperiod 4000
set port dot1x 0/5 re-authentication enable

You can force a manual re-authentication on a per-port basis, with this command:

IOS Switch: **dot1x re-authenticate interface FastEthernet 0/5**
CatOS Switch: **set port dot1x 0/5 re-authenticate**

This will force an already-connected client to re-submit its EAP credentials for authentication. This is useful for disconnecting suspect users (i.e. auditors).

You can also force an interface into a “force authorized” state, basically unlocking an interface in an 802.1x failed/unauthorized state:

