Vipin Jain
3Com Corporation

Paul Congdon
Hewlett Packard Company

John R. Vollbrecht
MERIT Network, Inc.

Andrew Smith
Extreme Networks

Raj Yavatkar
Intel Corporation

Brian Rosen
Fore Systems

John Roese
Cabletron

Port Based Network Access Control

1. Introduction

The IEEE 802.1 specification currently does not provide for authentication of peers accessing the switch port. There are circumstances in which it may be desirable to restrict access to publicly accessible ports. For example, academic institutions may wish to restrict access to registered students or faculty. This proposal describes mechanisms for addition of authentication support to IEEE 802.1.

2. Terminology

This document frequently uses the following terms:

Authenticator
: The end of the point-to-point link requiring the authentication. The authenticator specifies the type of authentication to be used.

Peer
: The other end of the point-to-point link which is being authenticated by the authenticator.

3. Overview

This proposal defines how a local access switch will perform authentication of a host. The purpose of this is primarily for a local access switch to validate the claim of identity by a host, so that access may be granted to the switch port. However, it is also possible for the authentication process to allow the host to authenticate the local access switch.

3.1. Port access restrictions

This document envisages authentication occurring primarily at host boot time or when a host is connected to the network. The host is allowed complete access only after the authentication succeeds.

In the current IEEE 802.1 specification, a switch port is either disabled, forwarding or controlled by the spanning tree protocol. In an authentication-aware switch, the switch port is in the "disabled" or “non-authenticated” state prior to authentication. Once authentication has succeeded, the port is put into "forwarding" state. While in the "blocked" state, DHCP and other initialization traffic will not be forwarded, and as a result, authentication will typically need to occur early in the host boot sequence (prior to DHCP and IP initialization for example).

Authentication-aware switches must support the ability to associate port forwarding state with host authentication and must support the ability to restrict the use of a port by a single host.

Authentication-aware switches may support a process for aging authenticated ports, and may request that a port re-authenticate at any time. Switch ports are in "forwarding" state during re-authentication, and transition to the "blocked" or "non-authenticated" state if re-authentication fails.

In general, authentication should be configurable on a per-port basis, since it will be desirable in some configurations not to perform authentication on certain ports (inter-switch links, server-attached ports etc.).

3.2. Edge authentication

Host authentication must occur at the first point of attachment (i.e. local access switch). Requiring that authentication occur at the edge rather than in the core has several advantages:

a. Security. All authenticated hosts on the local access switch are protected from non-authenticated hosts. If authentication were performed on a core switch it would be possible for a malicious host to attack authenticated hosts connected to the same local access switch or any number of other local access switches between this switch and the core switch. These attacks are eliminated by limiting service to non-authenticated hosts directly on the local access switch.

b. Complexity. In the current IEEE 802.1 spec, a switch port is either disabled, forwarding or controlled by the spanning tree protocol. If authentication is performed in a core switch, there is a possibility of multiple bridge protocol entities on a shared segment initiating authentication. To avoid this, the bridge protocol entity would have to work with the spanning tree states to make sure that only the bridge in the forwarding path initiates authentication. This complicates the bridge design.

c. Scalability. Implementing authentication in the core switch would require that authentication depend on individual MAC addresses not just on physical point of attachment. This in turn requires that the state be kept in the MAC address table. This increases the implementation cost and would require changes to the IEEE 802.1D standard with respect to address aging and learning. Topology changes and spanning tree reconfiguration complicate the interaction in a large network.

d. Availability. Todays's switched networks are designed with availability as one of the primary goals. The core of the network is redundant and fault-tolerant. If authentication is performed in the core, it would require re-authentication of all the hosts whenever topology changes causing port state changes in the spanning tree.

e. Translational bridging. Performing authentication at the access switch avoids complications arising from translational bridging or VLANS. If only a single link exists between the host and the switch, then frames need not be translated or tagged during the authentication exchange. The path to a core switch, deep in the network, may involve a variety of link types (e.g. FDDI, Token-Ring, ATM) and packet formats (e.g. IEEE 802.1Q tagged, MAC encapsulations). The IEEE 802.1 bridging specification would need to spend a significant amount of time wrestling with these conversions. Thus, were authentication to be allowed on core switches, additional rules would need to be defined for other media types or VLAN trunk links to the core switches.

f. Multicast propagation. Were authentication to occur in the core switches, it would be necessary for local access switches to forward authentication traffic to the core switches so that they could respond. Since core switches are not able to sense host connection to the local access switch port, initiation would occur either on receiving traffic from a new host or via host initiation. Requiring a core switch to keep authentication state for each host does not scale. In order for host initiation to reach the core switch, this would require that these (multicast) frames be flooded by authentication-unaware access switches, impacting other hosts. In contrast, if authentication occurs only on local access switches, these multicast frames are not forwarded.

3.3. Logoff Mechanism

It is useful to provide a logoff mechanism for a switch port. When a user logs off a host, it is possible in some environments to bypass a new login and access the host and the network. Providing the logoff mechanism ensures that the session is terminated for a host.

4. Proposed Protocol

4.1. Overview

The approach taken is to encapsulate the Extensible Authentication Protocol (EAP) in Ethernet framing, also referred to as EAPOE (EAP Over Ethernet) in this document. EAP, described in [1], is a general protocol that supports multiple authentication mechanisms. For example, through the use of EAP, support for a number of authentication schemes may be added, including smart cards, Kerberos, Public Key, One Time Passwords, and others.

In addition to describing the encapsulation of EAP in Ethernet framing, this section describes the switch function that relays EAP messages between a peer and the authentication function. The switch function controls the forwarding state of a physical port but does not interfere with the authentication exchanges between a peer and authentication function.

This separation of authentication and switch function permits the use of a "back-end" authentication server that implements the various mechanisms needed to authenticate a peer. The switch function simply controls the forwarding state of a port based on the results of the authentication. For a full description of an authentication function see <draft-ietf-radius-ext-03.txt>. It will be apparent to the reader that this separation does not preclude implementation of both switch and authentication function in the same item of physical equipment.

Rather than only permitting a pre-determined authentication method, EAP allows the authenticator to request more information before determining the specific authentication mechanism. In EAP, the authenticator sends one or more Requests to authenticate the peer. The Request has a type field to indicate what is being requested. Examples of Request types include Identity, MD5-challenge, One-Time Passwords, Generic Token Card, etc. The MD5-challenge type corresponds closely to the CHAP authentication protocol. Typically, the authenticator will send an initial Identity Request followed by one or more Requests for authentication information. However, an initial Identity Request is not required, and may be bypassed in cases where the identity is presumed.

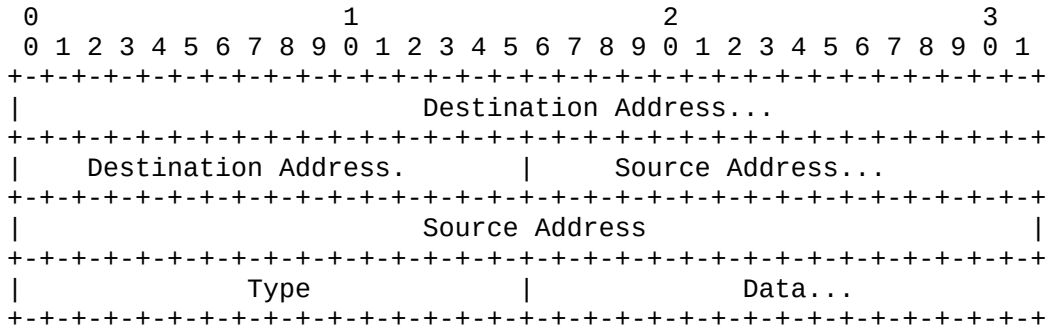
The peer sends a Response packet in reply to each Request. As with the request packet, the Response packet contains a type field which corresponds to the type field of the Request.

The authenticator ends the authentication exchange with a Success or Failure packet.

4.2. Packet formats

4.2.1. Ethernet frame format

A summary of the Ethernet encapsulation of EAPOE is shown below. The fields are transmitted from left to right.



Destination Address

The Destination Address field contains the address of the destination of the Ethernet frame. For EAPOE packets, the destination must be the EAPOE multicast address 01-20-9C-01-23-02. This address is expressed in IEEE illustrative (little-endian) notation.

[Issue: What is the desired behavior of the multicast traffic with respect to authentication aware and unaware switches? It might be advantageous to use the IEEE 802.1 reserved address range instead, as those packets are not supposed to be flooded by 802.1D bridges – even if they are non-authentication aware.]

Source Address

The Source Address contains the MAC address of the source of the Ethernet frame. This must represent a unicast address.

Type

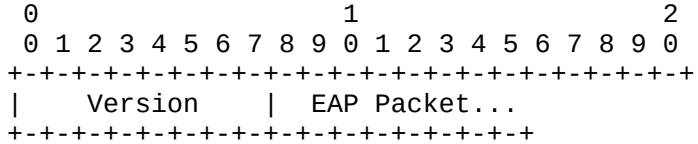
The EtherType value for EAPOE is not yet assigned.

Data

Exactly one EAPOE packet is encapsulated in the Data field of an Ethernet frame.

4.2.2. EAPOE Message Format

A summary of the EAPOE packet format is shown below. The fields are transmitted from left to right.



Version

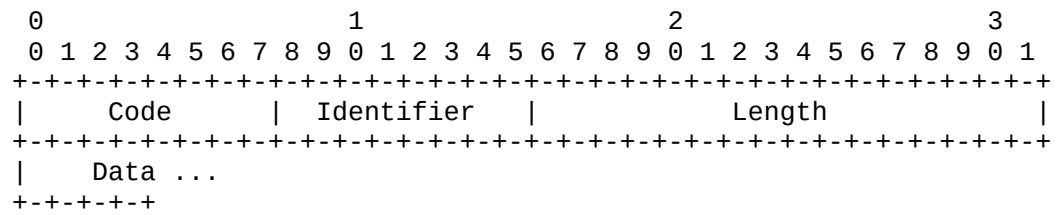
The Version field is one octet and identifies the current version of EAPOE protocol. The implementation conforming to this specification must use the value 1.

EAP Packet

The field contains the EAP packet as described below. Exactly one EAP packet is encapsulated.

4.2.3. EAP Packet Format

A summary of the EAP packet format is shown below. The fields are transmitted from left to right.



Code

The Code field is one octet and identifies the type of EAP packet. EAP Codes are assigned as follows:

1	Request
2	Response
3	Success
4	Failure

An EAP-Start message is an EAP-Request packet with no data in it.

Identifier

The Identifier field is one octet and aids in matching responses with requests. The Identifier field and switch port together uniquely identify an authentication exchange. Thus the use of a single octet identifier field results in a restriction of 256 authentications per switch port. This restriction is not likely to be an issue given that authentication occurs at the local access switch, rather than in the core.

In formulating the Identifier in new EAP-Request/Identity frame, the switch will choose a random value. The peer must insert this Identifier in subsequent EAP-Response frames. In formulating the identifier in a retransmitted frame, the switch will utilize the same Identifier value.

Length

The Length field is two octets and indicates the length of the EAP packet including the Code, Identifier, Length and Data fields.

Octets outside the range of the Length field should be treated as Data Link Layer padding and should be ignored on reception. Note: since in this specification EAP frames are transported directly over the link layer medium, the frame size should not exceed the maximum permissible on that medium, since fragmentation is not supported. For example, when transported over IEEE 802.3 link layer encapsulation, the frame size should not exceed 1500 octets.

Data

The Data field is zero or more octets. The format of the Data field is determined by the Code field.

5. Protocol Operation

5.1. Authentication initiation

Authentication can be initiated either by the peer or by the switch. If authentication is enabled on a given port, authentication must be initiated by switches on sensing a ’port up’ indication. As noted below, if the switch does not receive a response, it will retransmit the authentication frames with exponential backoff.

There are also situations in which a peer should initiate authentication. On booting, the peer may not be listening for an authentication initiation during the early portions of the boot sequence, and as a result, the peer may miss the initiation frame(s). As a result, if the host did not initiate authentication itself, an authentication failure could occur purely due to timing issues.

Note that a previously authenticated host may be rebooted. In this circumstance, an authentication-aware switch will typically not initiate authentication since it may not sense a ’port up’ indication. In this case, an argument could be made that authentication is not necessary since the host has already been authenticated. However, it is possible in some environments to reboot a machine, bypass the normal login and access the network. To prevent an unauthorized user from accessing the network by rebooting an authenticated machine, a host initiation may be necessary upon reboot.

The above arguments dictate that a host that does not receive an initiation frame from the switch on boot must initiate authentication by sending an EAP-Start frame. Any initial frames can be enqueued until the authentication either completes or fails.

5.1.1. Switch initiation

If the Identity is already known (such as via a previous authentication) then a switch may begin the conversation by sending a unicast EAP-Request frame, using the switch port MAC address as the source, and the peer MAC address as the destination. The switch will typically initiate the conversation when it receives a ’port up’ indication. Before authentication completes, the port is put in the "disabled" or "non-authenticated" state.

If the peer Identity is not known, then the switch must send a multicast EAP-Request/Identity frame. This frame will use the switch port MAC address as the source address, and the EAPOE multicast address as the destination. This is typically how an authentication-aware switch will begin the authentication conversation.

Peers receiving an EAP-Request frame from the switch must respond with an EAP-Response frame using the peer MAC address as the source address, and the EAPOE multicast address as the destination.

Authentication-aware switches may support a process for periodic re-authentication and may request that a port re-authenticate at any time. For example, if the switch reboots, authentication state can be recovered by multicasting EAP-Request/Identity frames on all ports. If the port is in "forwarding" state prior to re-authentication, then it will remain in that state during re-authentication. If the authentication fails for a port that was in "forwarding" state during re-authentication, the port is moved to "disabled" or "non-authenticated" state.

5.1.2. Peer initiation

In order to request that the switch initiate authentication, the peer sends an EAP-Start packet, using its own MAC address as the source address, and the EAP multicast MAC address as the destination. An EAP-Start frame consists of an Ethernet frame with Type=EAP, but containing no data.

The switch receiving an EAP-Start frame must respond by sending an EAP-Request/Identity frame, using the MAC address of the peer as the destination, and the switch port MAC address as the source address.

5.2 Logoff Mechanism

When switch port logoff is desired by a peer, the host originates an EAP-Start message but does not respond to the subsequent EAP-Request/ Identity messages from the switch. As a result, the switch will timeout, and the switch port will be placed in the "blocked" or "non-authenticated" state.

5.3. Soft state

Authentication-aware switches may maintain soft state relating to the port status. This implies that authentication state will be removed periodically (IDLE_TIMEOUT seconds).

The default IDLE_TIMEOUT interval is 3600 seconds (one hour). As with switch and peer initiated re-authentication, the implications of setting this to a lower value should be carefully thought out before proceeding.

5.4. Retransmission

As noted in [1], the EAP authenticator (switch) is responsible for retransmission of packets between the authenticating peer and the switch. The exception to this is the EAP-Start, which is retransmitted by the peer. Thus if an EAP packet is lost in transit between the authenticating peer and the switch (or vice versa), the switch will retransmit. The switch MUST adopt a retransmission strategy that incorporates a randomized exponential backoff algorithm to determine the delay between retransmissions. The delay between retransmissions should be chosen to allow sufficient time for replies from the host to be delivered based on the characteristics of the link between the host and the switch. For example, in a 10Mb/sec Ethernet network, the delay before the first retransmission should be 4 seconds randomized by the value of a uniform random number chosen from the range -1 to +1. Switches with clocks that provide resolution granularity of less than one second may choose a non-integer randomization value. The delay before the next retransmission should be 8 seconds randomized by the value of a uniform number chosen from the range -1 to +1. The retransmission delay should be doubled with subsequent retransmissions up to a maximum of 64 seconds.

As noted in [13] DHCP clients incorporate a similar randomized exponential backoff algorithm to determine the delay between retransmissions, allowing a retransmission delay of up to 64 seconds. Assuming that the switch is in blocking mode prior to authentication and initiates authentication on receiving an initial DHCP frame from the host, the initial DHCP frame will be dropped and will trigger authentication initiation by the switch. Assuming that the authentication can complete in time to avoid a DHCP timeout, the DHCP conversation will complete successfully.

Note that it may be necessary to adjust retransmission strategies and authentication timeouts in certain cases. For example, when a token card is used additional time may be required to allow the user to find the card and enter the token. If the user takes a particularly long time to find the card, then a DHCP timeout can occur.

This problem cannot be ameliorated by enqueing the initial DHCP frames since DHCP client timers are started when the packets are enqueued, not when they are sent.

5.5. Transition

It is desirable that the transition between a non-authenticated and an authenticated environment be as smooth as possible. When an authentication-capable peer connects to a non-authentication enabled access switch, the peer will not receive an EAP-Request/Identity multicast packet. As a result, the peer will initiate an EAP-Start frame to the multicast address. While non-authentication aware switches will typically forward this frame to all ports, there should be no ill-effects.

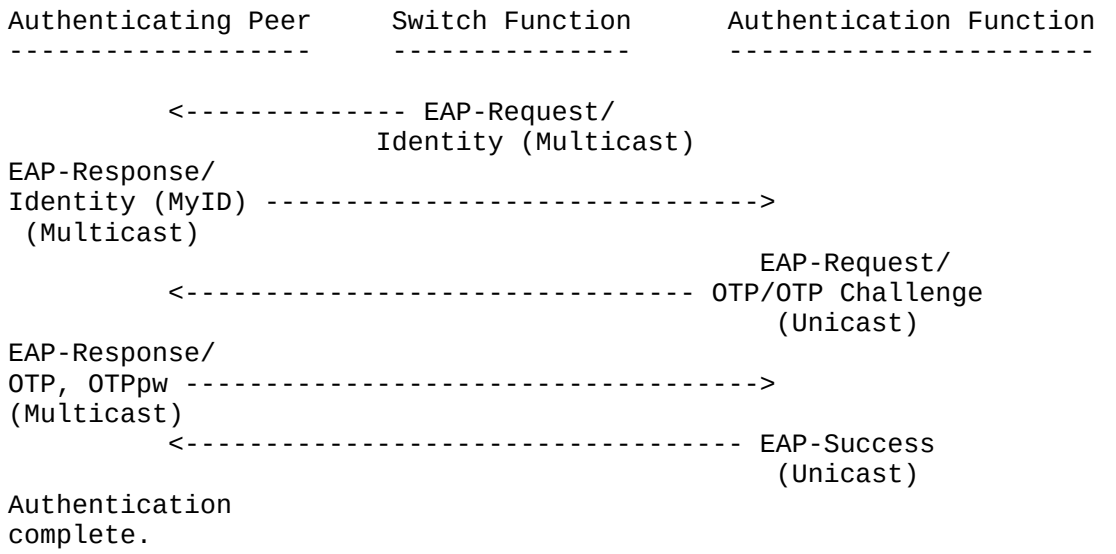
[Issue: It might be advantageous to use the IEEE 802.1 reserved address range instead, as those packets are not supposed to be

flooded by 802.1D bridges – even if they are non-authentication aware].

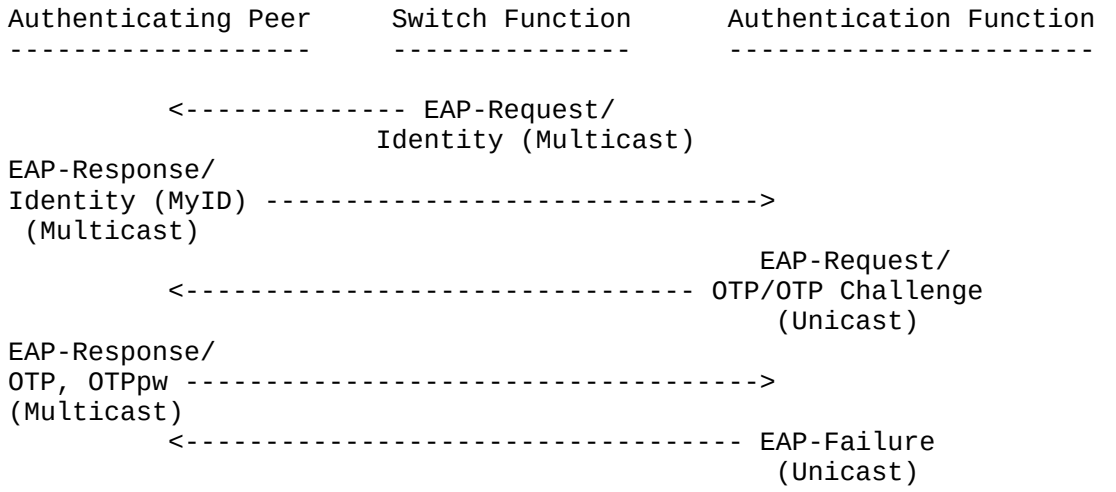
When a non-authentication aware peer connects to an authentication enabled access switch, the peer will ignore EAP-Request/Identity frames.

5.6. Examples

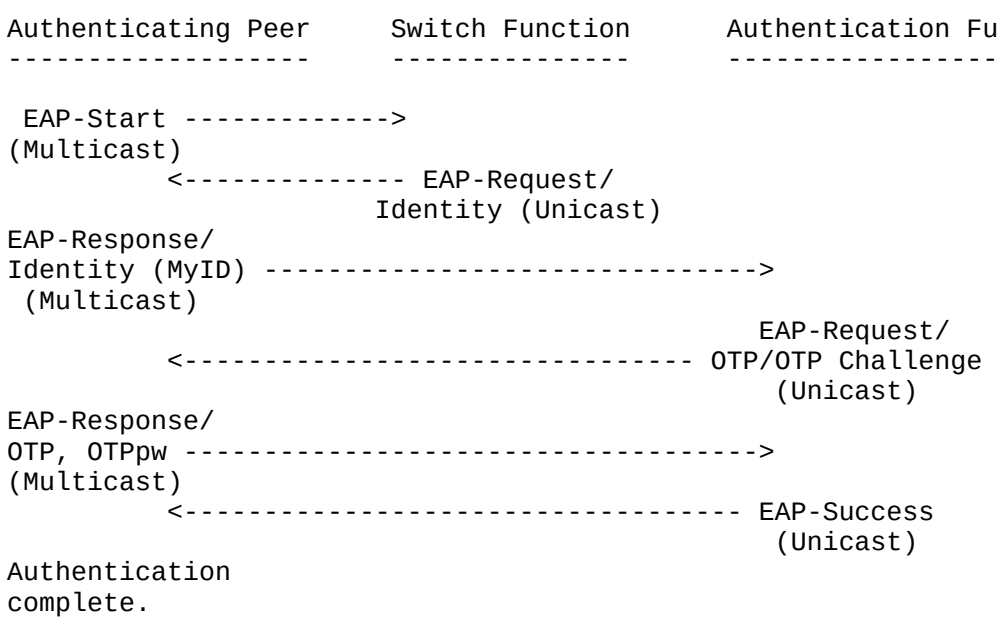
The example below shows a switch-initiated conversation for the case of a One Time Password (OTP) authentication. OTP is used only for illustrative purposes; other authentication protocols could also have been used, although they might show somewhat different behavior.



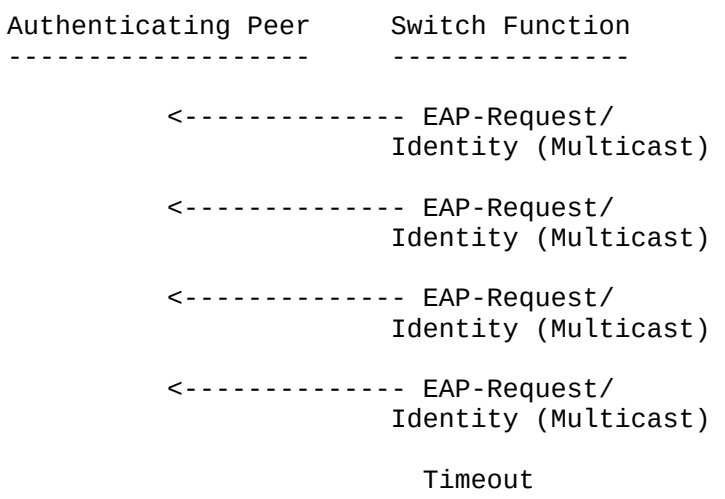
In the case where the peer fails EAP authentication, the conversation would appear as follows:



A peer-initiated authentication conversation will appear as follows:



In the case where the peer does not support authentication, but where authentication is enabled on that port, the conversation would appear as follows:



6. State machines

6.1. Switch state machine

The switch state machine consists of the following states:

Disconnected
Connected

Acquired
Authenticating
Hold
Authenticated

Disconnected
In this state, there is no connection on the port. This is the initial state which all authentication-enabled ports are in when the switch boots. The switch port should be in the "blocked" or "non-authenticated" state. When a 'port up' event is indicated, the switch transitions to the connected state for that port.

If the switch senses a 'port down' event, it transitions to the disconnected state for that port.

Connected In this state, the switch port has sensed a 'port up' event. It transmits an EAP-Request/Identity multicast on the port, starts a "tx" timer, and waits for a response. If the "tx" timer expires, the switch retries twice, sending an EAP-Request/Identity multicast and doubling the "tx" timer with each retry. If the switch times out after the second retransmission, it transitions to the quiet state for that port. The default value of "tx" timer (EAP_TX_TIMEOUT) is 5 seconds.

If the switch receives an EAP-Response/Identity multicast frame from a peer, it transitions to the authenticating state for that port. If the switch receives an EAP-Start frame, it transitions to the acquired state for that port.

If the switch senses a 'port down' event, it transitions to the disconnected state for that port.

Quiet In this state, the switch has sensed a 'port up' event, but has not acquired a peer. The switch starts a "quiet" timer. If the quiet timer expires, the switch transitions to the connected state for that port. The default value of "quiet" timer (EAP_QUIET_TIMEOUT) is 60 seconds.

If the switch receives an EAP-Response/Identity multicast frame from a peer, it transitions to the authenticating state for that port. If the switch receives an EAP-Start frame, it transitions to the acquired state for that port.

If the switch senses a 'port down' event, it transitions to the disconnected state for that port.

Acquired In this state, the switch has acquired a peer. The switch transmits an EAP-Request/Identity unicast frame to the first acquired peer, starts the "tx" timer, and waits for a response. If the "tx" timer expires, the switch retries twice, sending an EAP-Request/Identity multicast and doubling the "tx" timer with each retry. If the switch times out after the second retransmission, it transitions to the connected state for that port.

If the switch receives an EAP-Response/Identity multicast frame, it transitions to the authenticating state for that port.

If the switch senses a ’port down’ event, it transitions to the disconnected state for that port.

Authenticating

In this state, the switch is authenticating a peer. The switch transmits an EAP-Request unicast frame to the peer, starts the “tx” timer, and waits for response. If the “tx” timer expires, the switch retries twice, sending an EAP-Request unicast frame to the peer and doubling the “tx” timer with each retry. If the switch times out after the second retransmission, it transitions to the connected state for that port.

If the switch receives an EAP-Response multicast frame from a peer, it responds with another EAP-Request if required until authentication fails or succeeds. The “tx” timer is restarted (with the default value) at each stage with two retries. If authentication fails, the switch sends an EAP-Failure frame to the peer and transitions to the hold state for that port. If authentication succeeds, the switch sends an EAP-Success frame to the peer and transitions to the authenticated state for that port.

If the switch senses a 'port down' event, it transitions to the disconnected state for that port.

Hold

In this state, the switch is holding for a time period in which it will not accept packets so as to discourage brute force attacks.

The switch blocks the port, starts the “quiet” timer, and waits for its expiry. At the expiration of “quiet” timer, the switch transitions to the connected state for that port.

If the switch senses a 'port down' event, the switch transitions to the disconnected state for that port.

Authenticated

In this state, the switch has successfully authenticated the peer. The switch places the port in the "forwarding" state. If the switch supports port aging, it starts the “reauthenticate” timer for that port and transitions to the connected state on its expiry. The default value of “reauthenticate” timer is 3600 seconds.

If the switch receives an EAP-Start multicast frame, it transitions to the acquired state for that port.

If the switch senses a 'port down' event, it transitions to the disconnected state for that port.

6.2. Peer state machine

The peer state machine consists of the following states:

- Acquiring
- Acquired
- Authenticating
- Authenticated

Acquiring In this state the peer is attempting to acquire a switch to authenticate to. This is the initial state that the peer is in on warm or cold restart. The peer sends an EAP-Start frame to the multicast address, sets a “start” timer and waits for the switch to send a unicast or multicast EAP-Request/Identity frame. If the timer expires, the peer sends another EAP-Start frame, doubles the “start” timer, and starts again. The peer retries twice and then assumes that it is attached to a non-authentication enabled switch and transitions to authenticated state. The default value of “start” timer (EAP_START_TIMEOUT) is 2 seconds.

On receiving a unicast or multicast EAP-Request/Identity frame, the peer transitions to the acquired state.

Acquired In this state the peer sends a multicast EAP-Response/Identity frame to the switch, sets a “auth” timer and waits for a unicast EAP-Request frame. If the timer expires, the peer transitions to the acquiring state.

If the peer receives a unicast EAP-Request frame (not an Identity), it transitions to the authenticating state. If the peer receives a unicast or multicast EAP-Request/Identity frame, it sends a multicast EAP-Response/Identity frame to the switch, sets the “auth” timer and waits for a unicast EAP-Request frame. The default value of “auth” timer (EAP_AUTH_TIMEOUT) is 60 seconds.

Authenticating
In this state the peer is authenticating to the switch. It sends an EAP-Response frame, sets the “auth” timer and waits for a unicast EAP-Request, EAP-Success or EAP-Failure frame. On timer expiration, the peer transitions to the acquired state. On receiving a unicast EAP-Request frame (not an Identity), the peer sends an EAP-Response frame, sets the “auth” timer and waits for a unicast EAP-Request, EAP-Success or EAP-Failure frame. On receiving a unicast EAP-Failure frame, the peer transitions to the acquiring state. On receiving a unicast EAP-Success frame, the peer transitions to the authenticated state. On receiving a unicast or multicast EAP-Request/Identity frame, the peer transitions to the acquired state.

Authenticated
In this state the peer is successfully authenticated by the switch. On receiving a unicast or multicast EAP-Request/Identity frame, the peer transitions to the acquired state.

7. Additional Services

This section describes additional services that can be provided along with port based access control. Some of these services are enabled by IEEE 802.1Q VLANs. This is done to give the reader some insight into different operating environments and not meant to be an addition to 802.1D specification as it is outside the scope of IEEE 802.1D.

7.1. Manageability of Hosts

In many installations, it is essential that network management traffic be allowed between the network management station and hosts, in order to permit activities such as network monitoring and software update. If many hosts were to be made inaccessible as a result of failed authentications, network management capabilities would be compromised.

For example, as a result of a power failure, it is conceivable that many hosts would be unable to successfully authenticate, and as a result might be unable to locate a DHCP server. Were these hosts to be completely cut off from the network, then they would never receive a routable IP address, and network administrators would be unable to diagnose the problem, since the hosts would not be reachable by the network management station.

To address this issue, authentication-aware switches may support VLAN policy. This allows the switch to assign a VLAN to a port based on the outcome of authentication. In authentication-aware switches supporting VLANs, a port is put into the "forwarding" state during authentication, permitting access to the "non-authenticated" VLAN. Once authentication has succeeded, a new VLAN ID is assigned for that port, and the port remains in "forwarding" state.

Switch support for a non-authenticated VLAN enables hosts failing authentication to obtain IP addresses via DHCP so that they can remain manageable. This is useful for enabling hosts to obtain an account and login credentials via a registration server. This also makes it possible to keep track of unauthenticated hosts and manage them if necessary.

7.2. Accounting and Policies

Authentication-enabled switches may support additional services such as accounting or QoS policy. For example, after a connection is sensed on a port, a timer can be reset, or after authentication succeeds, the switch can reset the port counters. This allows the switch to keep track of how long connectivity was maintained on a port or how many octets were sent in and out. It is also possible for the switch to tag packets entering or leaving the port with a given priority, based on the host identity.

7.3 Host Identity for Access

The Identity presented by the host may either correspond to a user, group, or machine identity. Host implementations supporting use of a machine identity will typically authenticate once at startup, and will remain authenticated until a 'port down' event occurs or the host or switch reboots or re-authentication occurs. In such implementations, accounting data will indicate a single long-lived session for host. Thus, it will not be possible to account for usage by user. In contrast, implementations supporting user or group identity may authenticate with each user login. In such implementations, accounting data will provide per-user information.

7.4 VLAN Enhancements

As discussed previously, VLANs can be used to facilitate management of hosts failing authentication.

When logoff mechanism is used with VLAN-enabled switches, the host is placed in an "non-authenticated" VLAN during the period between logins. In order to maintain IP connectivity, the host would need to release its DHCP address, and acquire a new address so that it would be functional in the non-authenticated VLAN during the period between logins.

Note that with a VLAN-enabled switch, peer initiation is required in order to guarantee assignment of an authenticated address. When VLANs are supported, a DHCP server will typically be provided on the unauthenticated VLAN. As a result, without host initiation the initial DHCP packet sent by the host could reach the DHCP server on the unauthenticated VLAN, allowing the DHCP conversation to complete prior to authentication. The result is that an authentication-capable host will be assigned to the non-authenticated VLAN. Timing problems are less likely for switches without VLAN support, since the port will be blocked and thus the DHCP conversation cannot complete prior to authentication.

8. Security considerations

The following security issues have been identified relating to switch port authentication:

- Piggybacking
- Snooping
- Crosstalk
- Rogue switch
- Bit flipping
- Negotiation attacks

8.1. Piggybacking

Since it is possible that more than one host may be connected to a switch port, a switch implementing this specification must support anti-piggybacking functionality. Piggybacking occurs when an

unauthenticated host gains access to the switch port based on the successful authentication of another host. In order to enable piggybacking prevention, authentication-aware switches must be configurable on a per-port basis to set an alarm or block port access when multiple hosts are detected.

8.2. Snooping

In this attack, an attacker on the same switch port listens in on the authentication conversation in an effort to gain further information useful in an attack. Since EAP transmits the Identity in the clear, it is possible for an attacker to learn the identity of users authenticating to the switch. However, password compromise can be avoided by use of EAP methods employing strong cryptography.

8.3. Crosstalk

In this attack, a peer on one port attempts to interfere with authentications occurring on another port. For example, a peer may send an EAP-Failure message to the broadcast address, or to a peer on another port, or may send an EAP-Response to the MAC address of another switch port.

In order to prevent crosstalk between ports, authentication-enabled switches must discard all frames with EtherType=EAPOE and a destination address other than the switch port MAC address or the multicast address. In addition, authentication-enabled switches must not leak EAP frames destined for the multicast address to other ports.

Alternatively, a host on another LAN may attempt to send packets that will interfere with switch port authentication occurring on another segment. However, this is not possible since EAPOE frames are not routable.

8.4. Rogue switch

In this attack, the attacker replaces the switch with a suitably modified device. In such an attack, the attacker could send an EAP-Request with a lesser form of authentication (for example, EAP-MD5 with a static challenge) in order to perpetrate a dictionary attack and recover the user’s password. This attack can be prevented by configuring the client to require an EAP type supporting mutual authentication.

8.5. Bit flipping

The goal of EAP is to provide extensible authentication. Other security services, including integrity protection, encryption, or replay protection are not provided by this proposal. If such services are desired, then it is recommended that other solutions that provide security associations such as IPSEC [6] be employed.

8.6. Negotiation attacks

In this attack, the attacker attempts to subvert the EAP negotiation by inserting or modifying packets on the wire. The goal of this attack is to deny service or to reduce the level of security negotiated between the switch and the peer.

While individual EAP authentication types may provide message integrity protection for the data portion of EAP-Request and EAP-Response packets, the EAP header is not integrity protected. In addition, EAP-Success and EAP-Failure messages are not integrity protected, nor are EAP-Request and EAP-Response packets of types Identity, NAK, OTP, or MD-5.

This means that an attacker can send an EAP-Failure message to the peer from the switch's MAC address without fear of detection. Also, in response to an EAP-Request sent by the switch, the attacker could send an EAP-NAK in an attempt to cause the switch and peer to negotiate down to a less secure form of authentication.

While such attacks can result in a denial of service, the attacker must have physical access to the switch port in order to carry them out. Such attacks are detectable by switches, RMON probes, or sniffers, and can be made more difficult by having switches employ spoofing protection, i.e., dropping incoming frames claiming to originate from switch MAC addresses.

Subversion of the authentication negotiation can be averted using negotiation policy on the peer and switch. For example, the peer or switch can be configured to only accept a single form of authentication for a claimed Identity.

9. References

[1] Blunk, L., Vollbrecht, J., "PPP Extensible Authentication Protocol (EAP)", RFC 2284, March 1998.

[2] Rivest, R., Dusse, S., "The MD5 Message-Digest Algorithm", RFC 1321, April 1992.

[3] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", RFC 2119, March, 1997.

[4] Rigney, C., Rubens, A., Simpson, W., Willens, S., "Remote Authentication Dial In User Service (RADIUS)", RFC 2138, April 1997.

[5] Yergeau, F., "UTF-8, a transformation format of Unicode and ISO 10646", RFC 2044, October 1996.

[6] Atkinson, R., "Security Architecture for the Internet Protocol", RFC 1825, August 1995.

[7] IEEE Standards for Local and Metropolitan Area Networks: Overview and Architecture, ANSI/IEEE Std 802, 1990.

[8] ISO/IEC 10038 Information technology - Telecommunications and information exchange between systems - Local area networks – Media Access Control (MAC) Bridges, (also ANSI/IEEE Std 802.1D-1993), 1993.

[9] ISO/IEC Final CD 15802-3 Information technology - Telecommunications and information exchange between systems – Local and metropolitan area networks - Common specifications - Part 3: Media Access Control (MAC) bridges, (current draft available as IEEE P802.1D/D15).

[10] IEEE Standards for Local and Metropolitan Area Networks: Draft Standard for Virtual Bridged Local Area Networks, P802.1Q/D8, January 1998.

[11] ISO/IEC 8802-3 Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Common specifications - Part 3: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications, (also ANSI/IEEE Std 802.3-1996), 1996.

[12] IEEE Standards for Local and Metropolitan Area Networks: Demand Priority Access Method, Physical Layer and Repeater Specification For 100 Mb/s Operation, IEEE Std 802.12-1995.

[13] Droms, R., "Dynamic Host Configuration Protocol", RFC 2131, March 1997.

10. Acknowledgments

Thanks to Bernard Aboba of Microsoft for useful discussions of this problem space.