Chapter 2, Port-Based Authentication Concepts

Author: Jim Geier

Principal Consultant, Wireless-Nets, Ltd.

Email: jimgeier@wireless-nets.com

This chapter is a sample from the book Implementing 802.1x Security Solutions, made available with permission from Wiley Publishing.

Wireless Nets, Ltd.

Independent Consulting Services

www.wireless-nets.com

CHAPTER

2

Port-Based Authentication Concepts

This chapter focuses on concepts dealing specifically with 802.1X and port-based authentication. It introduces you to the applicable terminology and protocols, such as EAPOL, EAP, EAP-Methods, and RADIUS, that make up a port-based authentication system. The following chapters cover these protocols in detail, but for now you'll learn how they work together and how they enable an 802.1X port-based authentication system to operate.

802.1X Port-Based Authentication Terminology

Authentication is the process of identifying a person or thing (see Figure 2-1). For example, Sally arrives at an airport and attempts to check in for her flight to Dallas. The airline agent asks to see Sally's driver's license to ensure that the person claiming to be Sally is indeed Sally. The agent looks at the driver's license and verifies that Sally is the person in the photo on the license, and the name on the license is Sally. This information is Sally's *credentials*, which is accepted for many transactions, such as checking in for airline flights. The process as just described, which we're all very familiar with, is *authentication*. The airline agent has verified Sally's identity through credentials, thus authentication has taken place. Based on the credentials, the airline clerk can either authorize or not authorize Sally to continue checking into the flight.

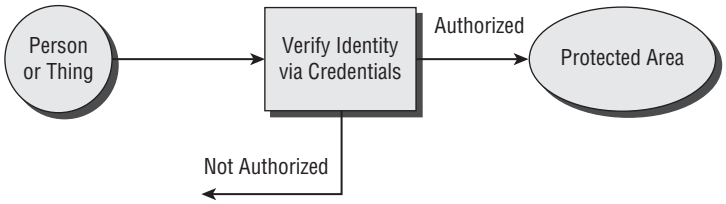

Figure 2-1: Simple example of authentication

Based on the airport analogy, authentication seems pretty simple. You merely verify that someone or something is who or what they claim to be, assuming the person or thing has the right credentials. That's really all there is to the basic definition of authentication.

An authentication system for a computer network can get more complex, though. Machines (such as Ethernet switches and network interface cards) must be told exactly what to do, and the precision and complexity of the necessary instruction set leaves no room for error. Incompatible protocols and even very slight misunderstandings in communications with machines generally results in non-interoperability—the system doesn't function correctly and ceases to be of any value. Humans, though, can apply reason to make adjustments when communication gets rough. For example, assume that Sally is checking into an international flight and she speaks only French, whereas the airline agent speaks only Spanish. The airline agent may ask in Spanish for Sally to produce her passport. Sally wouldn't understand this, and may instead show the airline agent her driver's license. The airline agent doesn't accept the driver's license, but despite the language difficulties, the airline agent may use some other means, such as body language, to impress on Susan that she must produce a passport. This may seem like a trivial example, but the point is that humans can adapt very easily based on the situation. Machines have some capability to adapt, but machines must be programmed and configured to adapt to specific situations. The designers of the network components, for instance, may not have thought of every conceivable situation, so the system has a tendency to break when unforeseen circumstances occur.

In addition to the standard, run-of-the mill issues just described, the standards and specifications that form a complete 802.1X port-based authentication system are written by different organizations: the Institute of Electrical and Electronic Engineers (IEEE) and the Internet Engineering Task Force (IETF). The IEEE standard that applies to port-based authentication is 802.1X, which addresses EAPOL, and the IETF provides RFCs for EAP, EAP-Methods, and RADIUS. All of these standards and specifications are needed. Thus, no single integrated standard specifies all of the components needed to implement a complete port-based authentication system. This results in a complexity that in turn often results in interoperability issues. In fact, the multiple

standards that make up a port-based authentication system are what makes learning 802.1X and related specifications relatively difficult. You could download a copy of 802.1X and study it for weeks, but without references to several other documents, you probably wouldn't have a clue how port-based authentication really works. Standards and specifications also often change, and you must be careful to ensure that what you implement is backwardly compatible with versions that you choose for parts of the system. So, again, the point here is that 802.1X is much more difficult than what the definition of simple authentication implies.

Another term that must be understood in the realm of port-based authentication is "port," which is a Layer 2 (Data Link) connection in a computer network. For wired networks, the word "port" in port-based authentication refers to a port on an Ethernet switch, as shown in Figure 2-2. Of course, many different hardware devices, such as desktop PCs, laptops, servers, cameras, access points, and hubs, can connect to an Ethernet port. The complete link connection, in relation to networks, is made at Layer 1 (Physical Layer) and Layer 2 (Data Link Layer). The Ethernet cable that provides the interconnection establishes the physical part of the link. Port-based authentication attempts to verify the identity of these devices connected to the Ethernet port via a physical cable, and the authentication takes place at Layer 2 (Data Link Layer).

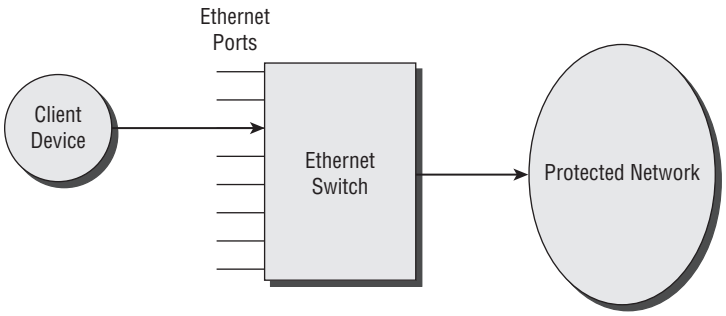

Figure 2-2: A wired Ethernet port provides a physical link.

Ports also apply to wireless LANs, but in the wireless world, the port is an association with an access point. Instead of producing a physical connection, a wireless client device, such as Wi-Fi-enabled laptop, goes through a process of associating with an *access point*. All access points in a wireless LAN periodically broadcast an 802.11 beacon frame. When a wireless client equipped with an 802.11 client radio first boots up, the client radio scans all channels and identifies the presence of access points in the surrounding area. The client radio then attempts to associate with the access point having the strongest signal. The association process involves a series of 802.11 frame transmissions between the client device and the access point, which results in the associated

state shown in Figure 2-3. A successful association with an access point allows the wireless client, based on its MAC address, to communicate through the access point to other devices on the network infrastructure. As with an Ethernet port connection, the association provides a link whereby a device can be authenticated before being allowed access to the network.

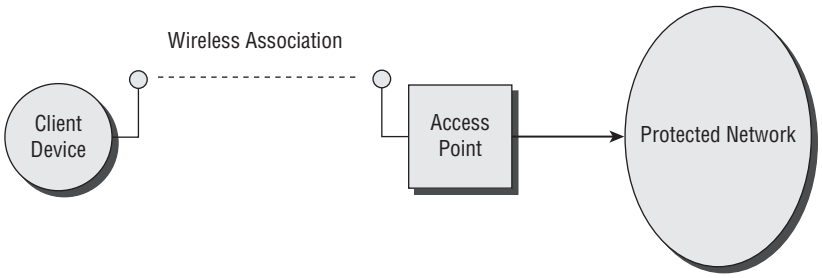

Figure 2-3: A wireless LAN association provides a virtual link.

Keep in mind that authentication is different from authorization. They're often treated closely together. Using the preceding analogy, when Susan showed valid credentials to the airline agent when checking into her flight, she was authenticated and then authorized to continue checking in. *Authorization* is a process of granting privileges to a person or device based on the outcome of the authentication. Imagine that John from the accounting department is attempting to log in to a network, and he is prompted to enter his username and password. After entering this information, the system verifies that John's username and password match what's contained in a database. So far, this has only involved authentication. Based on John's username, however, the system allows John access only to the servers belonging to the accounting department, and not any of the warehouse and human resources applications. This latter step of admitting John onto the network deals with authorization. Authorization is certainly important, but avoiding it for now will make learning authentication processes much easier.

Authentication Benefits

Port-based authentication keeps unauthorized users and client devices from accessing protected resources on the network, such as servers, corporate applications, and databases. Without authentication, a hacker could easily access the LAN by connecting a laptop to an Ethernet port within the facility, or associate with a wireless LAN access point from the parking lot of the company. If a hacker is allowed to connect to the network, they'll look for any and all ways to exploit security weaknesses.

Once connected to the network, a hacker has a surprisingly wide variety of tools and methods available to crack into corporate resources. A hacker, for example, could run a TCP (Layer 4) port scanner that causes all port 80 (http) devices connected to the network to echo back their IP address and other information, such as SNMP port status. Many of these port 80 IP addresses are http administration interfaces for access points, servers, and printers. Scores of companies fail to configure login security for administration interfaces on printers, which allows the hacker to aim their browser at a printer's administration port and reconfigure the printer. This may not sound like a big problem, but some printers allow you to configure the printer to print to a file, such as one located on the hacker's laptop, instead of (and in some cases in addition to) printing the associated document. Thus, a hacker connecting to the network may be able to passively redirect printed documents to their laptop. This is a significant compromise in security, especially when the printed documents contain employee social security numbers, competitive proposals, and sensitive intellectual property. There are dozens and possibly hundreds of ways that a seasoned hacker can breach security if they're able to connect to the network. Port-based authentication, however, will significantly reduce these risks and even prevent them from happening.

Implementing port-based access control constitutes a big step toward securing a wired or wireless network. It's not a silver bullet for providing network security, however. In addition, you must employ other methods, such as data packet encryption, intrusion detection, denial of service prevention, security awareness programs, and facility access controls in order to cover all possible security vulnerabilities.

In addition to keeping unauthorized people off the corporate network, a port-based authentication system also supports the following:

- **User location information:** An application can easily track the location of users, for instance, based on the switch or access point where the applicable client device was authenticated. The location information can map to a wide variety of applications. For example, a hospital can use this information to track the location of doctors and nurses using wireless client computing devices.
- **Billing and accounting mechanisms:** Port-based authentication, when combined with billing and accounting mechanisms, enables Internet service providers (ISPs) to implement fee-based Internet access. If the user is not authorized, they can be directed to pay for service via a credit card and then be given credentials (username and password) that the subscriber can use when logging into the system. Port-based authentication prompts users to enter their username and password, which the system uses to authenticate them. If the credentials match what the system has stored in a database, then a user will be authorized

to access the protected side of the network, which is the Internet. Figure 2-4 illustrates this concept.

- **Personalized network access:** Based on credentials offered during authentication, the system can authorize the user to access certain applications.

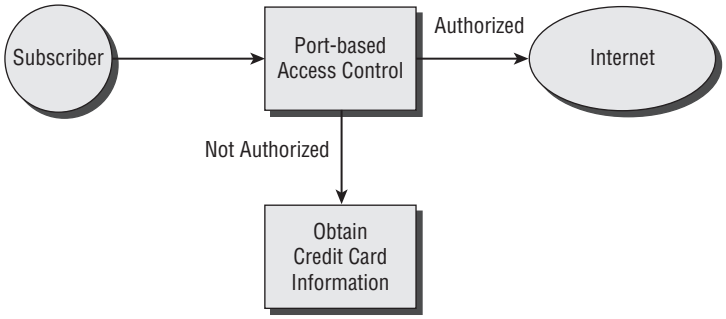

Figure 2-4: Fee-based Internet access controlled by port-based authentication

Primary Components

Until now, we've been looking at port-based authentication from a generic point of view, but now we should start using the proper names and actual protocols that you'll find in the 802.1X standards and specifications. As shown in Figure 2-5, the primary components of a port-based authentication system include supplicants, authenticators, and authentication servers.

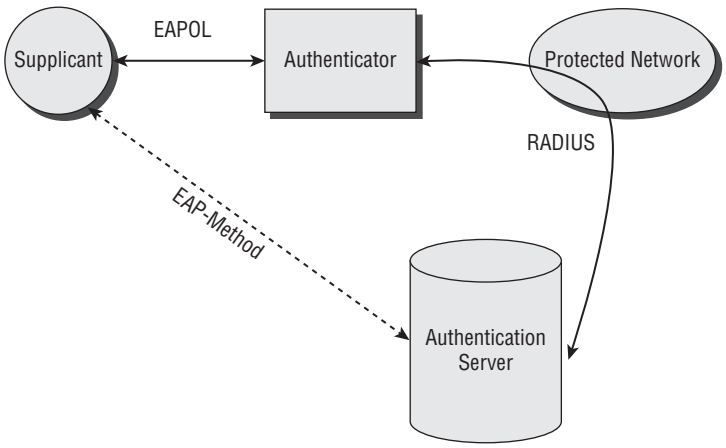

Figure 2-5: A port-based authentication system consists of a supplicant, authenticator, and authentication server.

Supplicant

A *supplicant* is a client device that needs to be authenticated before being allowed access to the network. Think of the supplicants as unknown users. Their identity is in question until they can produce valid credentials to the authentication server.

In order to be considered a valid supplicant, a typical client device, such as a laptop or IP phone, would need to implement 802.1X and a specific EAP-Method. For example, Windows XP comes with 802.1X built in with a variety of EAP-Methods, such as EAP-TLS. (Sometimes EAP-Methods are referred to as *EAP types*.) The supplicant communicates with the authentication server using EAP as the transport and a specific EAP-Method that provides the actual authentication mechanism. As explained later in this chapter, the actual communications between the supplicant and the authenticator is accomplished via EAPOL, which is defined by 802.1X. EAPOL delivers (encapsulates) the EAP and EAP-Method frames as data.

Authenticator

An *authenticator* is a Layer 2 network device, such as an Ethernet switch or a wireless LAN access point. In an enterprise network, all switch ports may implement 802.1X in order to support company-wide 802.1X port-based authentication. The authenticator acts as a security gate between the supplicants and the protected network. The gate (actually, port) stays closed until the authentication system verifies the credentials of the supplicant and deems that the supplicant is authorized to access the protected network. Once the system authenticates the supplicant, the authenticator will open a port so that the supplicant can access the protected network.

In addition, the authenticator is a translator between the supplicant and the authentication server. As the supplicant and authentication server converse, all communications flow through the authenticator. For example, the supplicant will send its credentials to the authentication server by encapsulating the credentials (based on the specific EAP-Method) in an EAP frame, which is all encapsulated in an EAPOL frame. The EAPOL frame is sent to the authenticator, which then removes the EAP-Method data from the EAPOL frame. The authenticator sends the EAP-Method data encapsulated in a RADIUS frame directly to the authentication server. Thus, the conversation between the supplicant and the authentication server is based on a common language.

Authentication Server

As mentioned above, the authenticator and the supplicant have a conversation regarding the authentication. The authentication server, for instance, will at some point request the credentials from the supplicant. The supplicant will

then offer the credentials to the authentication server. The port-based authentication standards and specifications don't make any particular type of authentication server mandatory, but nearly all implementations utilize RADIUS. As a result, RADIUS is the de facto standard recognized by the networking industry.

In an enterprise system, the authentication server is likely a separate component attached to the network. There will probably be multiple authentication servers to improve availability and performance. Each authenticator points to a primary authentication server, with possibly several others listed as secondary servers that can be called upon if the primary authentication server is unresponsive.

In some cases, the authentication server may be embedded in the authenticators. This distributed authentication server model significantly reduces authentication traffic over the network, which is desirable for wireless networks where roaming frequently occurs. This can improve performance for all clients. In addition, smaller networks may strongly benefit from using a switch or access point that also provides authentication server functions. This is cost effective for smaller networks because it reduces hardware costs.

A Simple Analogy: Getting the Protocols Straight

As you can see, 802.1X port-based authentication involves several different protocols—namely, EAPOL, EAP, EAP-Methods, and RADIUS. In addition, a lot of layering takes place with these protocols. With no single standard to refer to, it's easy to get lost and not make any sense of the details that the rest of this book will cover. Therefore, let's explore an analogy that should help you fully understand where the protocols apply and how the transfer of data takes place.

Imagine that Rob (supplicant), located in Bangor, Maine, writes and mails a letter (EAP-Method data) to his friend Tony (authentication server), who lives in Houston, Texas. Rob mails the letter via a special courier who will deliver the letter by truck (EAPOL/EAP) to Dayton, Ohio (authenticator), which is approximately halfway between Bangor and Houston. In Dayton, the courier continues delivery of the letter to Houston by airplane (RADIUS). Tony receives and reads the letter successfully. Figure 2-6 illustrates this process.

The delivery process for the letter is similar to the layering process that takes place in an 802.1X port-based authentication system. The overall goal of the system is to allow the supplicant (Rob) to communicate with the authentication server (Tony) via a particular EAP-Method, which includes the sending of EAP-Method data back and forth between the supplicant (Rob) and the authentication server (Tony). In order to funnel the EAP-Method data through the system, EAPOL (delivery truck) carries the letter to the authenticator (Dayton), and

RADIUS (airplane) delivers the EAP-Method data (letter) to the authentication server (Tony). Figure 2-7 depicts the actual 802.1X layering process. In addition, another layer, not depicted in the figure, would include the specific LAN protocols, such as 802.3 or 802.11.

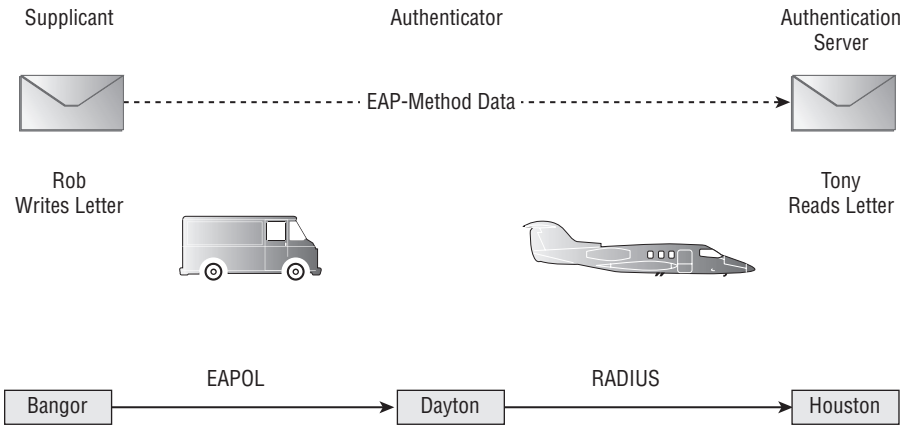

Figure 2-6: Analogous letter delivery depicting a port-based authentication layering process

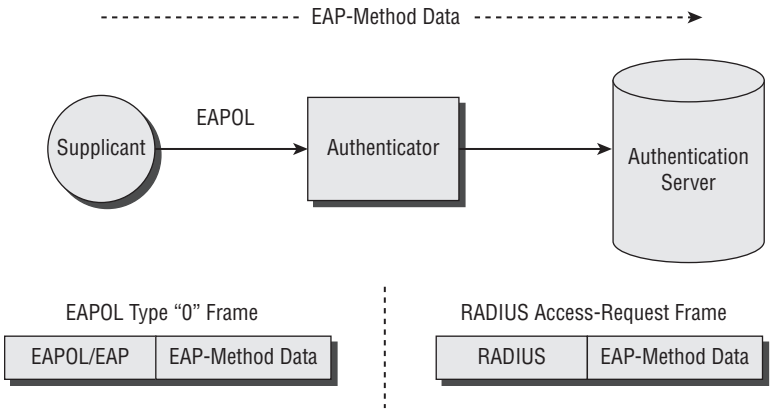

Figure 2-7: 802.1X port-based authentication system layering

The layering is done to allow different protocols between the supplicant and authenticator (EAPOL), and the authenticator and the authentication server (RADIUS). This enables the protocols to address the varying needs of each link in the system and at the same time allow a conversation to occur between the supplicant and authentication server (EAP-Method).

Port-Based Authentication Operation

The operation of an 802.1X-based port-based authentication system makes use of a variety of standards and specifications, which the previous sections identify. You now know what 802.1X port-based authentication does and what components and protocols are involved. Let's take a closer look at how the overall system operates.

A Simple Analogy—Understanding the Overall System

At the beginning of this chapter, an analogy describing Sally checking in for a flight at an airport defined authentication. This particular analogy was very simple and doesn't include all of the components of a port-based authentication system. As the following analogy unfolds, parenthetical terms and phrases map the analogy to an actual 802.1X port-based authentication system.

Assume that Terry (supplicant) arrives at the White House (protected network) in order to meet with the president (see Figure 2-8). As Terry enters the driveway (switch port), a gate guard (authenticator) orders Terry to stop the car. The gate guard blurts out, "Why are you here?" Terry says, "I'm here to see the president." The guard then calls Eva (authentication server), the primary point of contact for the President's security unit and lets Terry talk directly with Eva, who's inside the White House. Terry can be seen by Eva through a security video camera, and Eva asks Terry to put his passport in front of the camera so that Eva can clearly see his name, picture, and passport number (EAP-Method data). After verifying that Terry is who he claims to be (authenticated), Eva finds Terry's name on the list of authorized meeting attendees and tells the gate guard to issue him a pass to the meeting room (authorized services). The guard then lets Terry drive through and access the meeting room.

Several variations on this analogy often pop up in actual port-based authentication systems. If Eva doesn't find Terry's name on the authorized list, then she would tell the guard to not let Terry through (unauthorized access). The gate guard would use force if needed to keep Terry from entering the White House grounds. If Terry were a tourist (visitor), the guard would tell Terry that he could park down the street and arrange for a tour of the White House (guest access).

When Eva asks Terry to show his passport, Terry might not have one. Terry could negotiate with Eva and possibly use his driver's license instead. This comes up with port-based authentication methods when the supplicant doesn't support the primary EAP-Method. When this happens, the supplicant and the authentication server can negotiate use of a different EAP-Method. There's a possibility, however, that Eva may not accept a driver's license as a valid form of identification. If this were the case, then she would, obviously, inform the

gate guard to not let Terry through, and the guard would probably offer Terry guest access by signing up for a White House tour.

If Eva doesn't answer the phone when the guard calls, then the guard could try calling other people in the security unit. Eventually, the guard may reach a different security person, who can talk directly with Terry and handle the identification (authentication) process. The guard may not, however, reach anyone in the security unit. The phones may be down or the security unit may be so busy that they can't service any new requests. Terry would then need to wait until the guard puts him in touch with someone from the security unit.

Alternatively, Terry may arrive at the security gate, but the guard doesn't notice Terry is there. The guard may be yakking with another guard about last night's game. In this case, Terry may say, "Excuse me!" This would probably get the attention of the guard, who will then ask, "Why are you here?" The process then continues from that point on as described above.

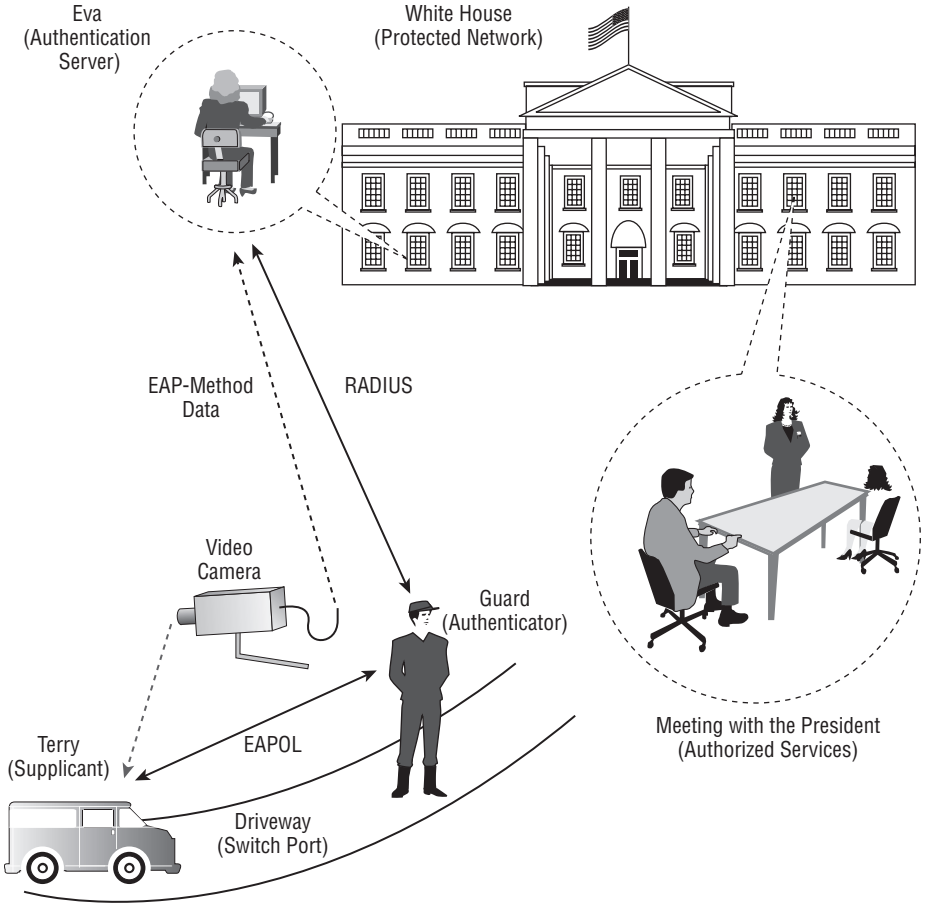

Figure 2-8: Analogous security gate authentication system

Supplicant to Authentication Server: EAP-Methods

The actual conversation regarding authentication occurs between the supplicant and the authentication server. This is similar to the conversation occurring between Terry and Eva in the preceding analogy, where Terry is the supplicant and Eva is the authentication server. In a port-based authentication system, a specific EAP-Method defines how the authentication takes place between the supplicant and the authentication server. The conversation between the supplicant and authentication server includes EAP-Method data, which represents various elements, such as the supplicant's credentials. Figure 2-9 illustrates communications between the supplicant and the authentication server. The conversation between the supplicant and the authentication server includes multiple exchanges of EAP data, depending on the type of EAP-Method.

The implementation and result of an EAP-Method is the goal of the port-based authentication system. The process that Terry and Eva completed when verifying that Terry, based on the information supplied in his passport, was indeed the person he said he was, is what an EAP-Method provides. In actual 802.1X port-based authentication systems, EAP-Methods make use of different types of credentials, such as username/passwords, encryption keys, and digital certificates.

The standards require implementation of the following EAP-Methods:

- MD5 challenge
- One-Time Passwords (OTP)
- Generic token card

In addition, there are many proprietary and RFC-based EAP-Methods, such as EAP-TLS, EAP-TTLS, EAP-FAST, and EAP-LEAP. Chapter 5, "EAP-Methods," discusses details of the various EAP-Methods.

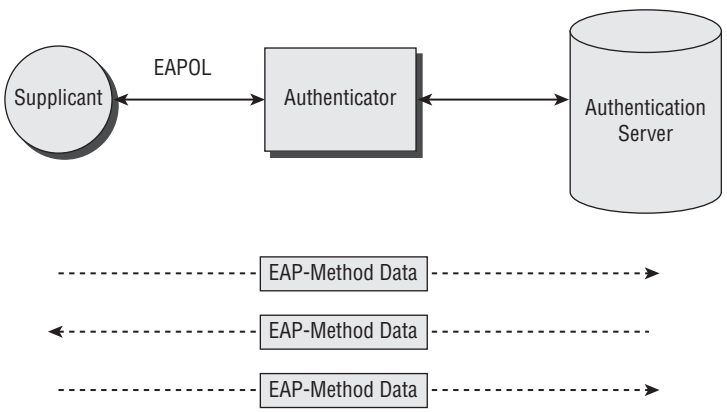

Figure 2-9: Communications between the supplicant and authentication server

Supplicant to Authenticator: 802.1X / EAPOL

802.1X only applies between the supplicant and the authenticator. This is analogous to communications between Terry and the gate guard in the White House example. A complete 802.1X port-based authentication system makes use of other protocols, such as RADIUS. 802.1X is only part of the overall system.

Figure 2-10 illustrates the communications between a supplicant and an authenticator.

EAP was designed as a point-to-point protocol (PPP) for communications over a serial link. EAPOL is defined in the 802.1X standard to adapt EAP for operation over LANs.

To do this, EAPOL adds three additional fields to EAP:

- Version
- Type
- Length

As a result, EAPOL encapsulates EAP frames as data. Chapter 3 explains the details of these fields. For now, in this chapter, it's important to learn the different types of EAPOL frames to understand the basics of 802.1X operation.

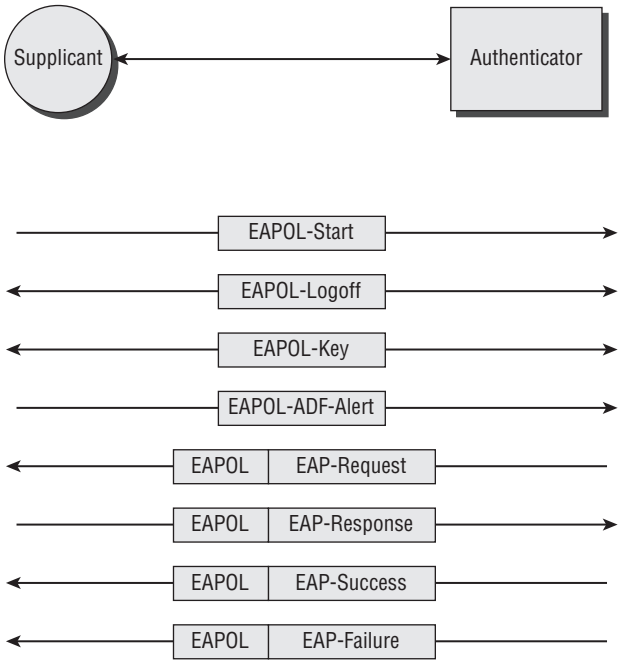

Figure 2-10: Communications between supplicant and authenticator

A Type "0" EAPOL frame means that the frame is carrying an EAP frame. This requires the destination, whether it's the supplicant or the authenticator, to merely strip off the EAPOL header and process the EAP frame. Thus, Type "0" EAPOL frames merely pass through EAP frames, which are generally carrying EAP-Method data.

In addition to carrying EAP-Method data, other EAP frames manage the authentication information. For example, EAP provides a mechanism for the supplicant and the authentication server to negotiate which EAP-Method to use. This is similar to the previous analogy when determining whether a driver's license could be used instead of a passport for verifying Terry's identity to Eva. In addition, other EAP frames provide the means for exchanging credentials and declaring the success or failure of the authentication. The exchange of these frames can be found in the analogy in which the gate guard tells Terry that either he can or he can't enter the White House grounds.

EAP doesn't have any security features, such as encryption of data carried in the EAP frame bodies. This requires designers to implement security in other layers. For example, if the link between the supplicant and the authenticator is wireless, then it would be best to implement some sort of link encryption, such as 802.11i. In this case, 802.11i would encrypt the data portion of the 802.11 frame, which contains the 802.1X protocols.

There are four EAP frame types:

- Request
- Response
- Success
- Failure

As mentioned earlier, EAPOL always carries these EAP frames in EAPOL Type "0" frames.

The supplicant can only issue EAP Response frames, and the authenticator can perform EAP Request, Success, and Failure frames. The authenticator issues EAP Request frames to deliver EAP-Method data traveling from the authentication server to the supplicant, and the supplicant issues EAP Response frames to deliver EAP-Method data going from the supplicant to the authentication server. An authenticator will send an EAP Success frame to the supplicant if the authentication server informs the authenticator that the supplicant is authorized to access the protected network. The authenticator will send an EAP Failure frame to the supplicant if the result of the authentication process indicates that the supplicant is not authorized to access the protected network. The EAP Success frames and Failure frames are sent in response to the EAP-Method outcome. In some cases, an authenticator may issue EAP Failure frames to the

supplicant to initiate the authentication process because the EAP Failure frame causes the supplicant to reset its link.

EAP provides correct ordering of the EAP frames through a "lock-step" mechanism. This is a simple process whereby the authenticator, for example, sets a value in the Identifier field of the EAP frame when sending an EAP Request frame to the supplicant. The supplicant sets the same value in the Identifier field of the EAP Response frame. This informs the authenticator that the supplicant has received the EAP Request frame and to move on to the next frame.

After the link between the supplicant and the authenticator becomes active, the authenticator sends an EAP Request frame (again, encapsulated in an EAPOL Type "0" frame) to demand the identity of the supplicant. The authenticator will not let any non-EAP-Method traffic through to the protected side of the network at this point. Based on the process defined in the EAP-Method, the supplicant and authentication server will converse using the EAP-Method. The communications between the supplicant and the authenticator include transfers of EAPOL Type "0" frames carrying EAP and EAP-Method data. The authenticator simply acts as a translator and keeps the EAP-Method data flowing between the supplicant and the authentication server until the authentication server decides whether to authorize or not authorize the supplicant.

Ultimately, the authenticator may assign the supplicant to an authorized VLAN. If the supplicant ends up not being authorized, then the authenticator can assign the supplicant to an unauthorized port (e.g., guest VLAN), such as one providing access to the Internet only, if this feature is available (see Figure 2-11). In some cases, the switch may support dynamic VLAN assignment so that the supplicant can be connected to one of several authorized VLANs based on authorization that applies to the credentials configured in the authentication server.

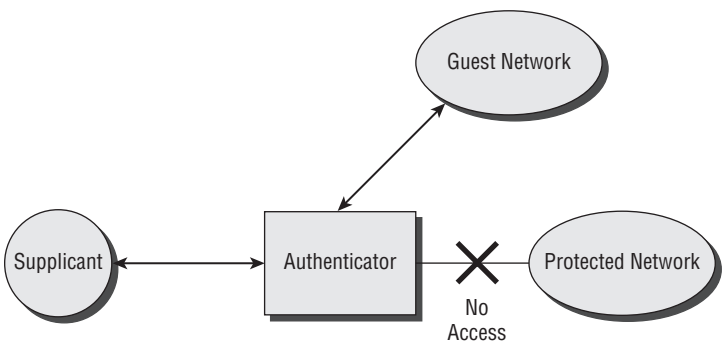

Figure 2-11: Unauthorized client allowed connection to a guest network

After the authenticator sends the initial EAP Request frame to the supplicant, there may be no response from the supplicant. Returning to the analogy presented earlier, Terry may have his car stereo turned up too loud and can't hear the guard. In a real network, the network interface card, such as the Ethernet card or 802.11 adapter, may be faulty or may not support 802.1X. After waiting for a specific period of time (which is configurable), the authenticator will attempt to resend the EAP Request frame. If the authenticator doesn't get any response from the supplicant after sending multiple EAP Request frames (the number is configurable), then the authenticator may shut down the link or connect the client to a guest VLAN, depending on configuration. In a wireless LAN, the authenticator disassociates the wireless client (supplicant) when shutting down the link.

So far, we've only discussed the EAPOL Type '0" frames for carrying EAP and EAP-Method data. Other types of EAPOL frames include the following:

- EAPOL-Start
- EAPOL-Logoff
- EAPOL-Key
- EAPOL-Encapsulated-ADF-Alert

These frames are outside the scope of EAP and don't carry EAP or EAP-Method data. Why, then, do we need them? They provide additional functionality needed to make EAP work on a LAN. EAP wasn't specifically designed for LANs; therefore, 802.1X devised EAPOL to wrap around EAP (EAPOL type "0" frames) and provide additional LAN functionality.

For example, a supplicant can send an EAPOL Start frame. This gets the attention of the authenticator, which responds immediately with an EAP Request frame that requests the identity of the supplicant. In the preceding analogy, this is the case where Terry drives up to the gate guard and the guard is busy talking to someone. Terry must get the guard's attention in order to start the process of checking into the White House grounds. A similar event can happen in real networks because the supplicant may be downstream from other devices, such as hubs, that have been authenticated, and the link is already active. The authenticator wouldn't know if the supplicant comes online; therefore, the supplicant must alert the authenticator with an EAPOL Start frame. Chapter 3 describes the other EAPOL frame types.

802.1X (i.e., EAPOL) applies to Layer 2 in order to keep a supplicant from connecting to the network before authenticating. If authentication is done at Layer 4, for instance, then a network connection would have to be made before starting the authentication process, making the network vulnerable to a hacker. As explained earlier in this chapter, a connection to the network at Layer 2 offers many opportunities for a hacker to exploit the security of the network.

In order to accomplish integration at Layer 2, 802.1X takes advantage of access controls offered by 802.1D, which defines MAC bridges. 802.1D is required by all 802 LANs, including 802.3 (Ethernet) and 802.11 (Wi-Fi). As a result, 802.1X will work with any of the LAN types. The integration is done is a way that keeps 802.1X traffic from disrupting other LAN protocols and allows 802.1X frames to be the first ones sent on the link.

802.1X makes use of the addressing reserved for the 802.1D Spanning-Tree Protocol. 802.1D owns several reserved group addresses. With group addresses, every member of the group processes the frame. 802.1X has been assigned one of the unused 802.1D Spanning-Tree group addresses, which is 01:80:C2:00:00:03. This address is often referred to as the *802.1X Port Access Entry (PAE)* address. All 802-based devices (client cards, switches, access points, etc.) are designed to receive and process frames having this group address.

NOTE When using a packet sniffer, you can easily trace 802.1X communications by filtering the trace on the group MAC address: 01:80:C2:00:00:03, which is uniquely assigned to all 802.1X frames.

Authenticator to Authentication Server: RADIUS

Figure 2-12 illustrates communications between the authenticator and the authentication server using RADIUS. Similar to EAP, RADIUS frames are sent using a lock-step process. RADIUS frame types include the following:

- Access-Request
- Access-Accept
- Access-Reject
- Access-Challenge
- Accounting-Request
- Accounting-Response

Most communications between the authenticator and the authentication server consist of RADIUS Access-Request and Access-Challenge frames. The authenticator sends EAP-Method data to the authentication server via RADIUS Access-Request frames. The authenticator will have removed the EAP-Method data from an EAPOL/EAP frame that it received from the supplicant. If the authentication server receives a RADIUS Access-Request and the IP address of the authenticator, and a shared secret provided by the authenticator matches what the authentication server is expecting, then the authentication server will process the request. If these items don't match, then the authentication server

remains silent and doesn't respond at all. The authenticator will likely keep repeating the RADIUS Access-Request frame multiple times, however. If configured properly, the authenticator will eventually give up and try communicating with a connection with a different RADIUS server. The authentication server sends EAP-Method data to the authenticator (and bound for the supplicant) via RADIUS Access-Challenge frames. Of course, the supplicant will extract the EAP-Method data from the RADIUS frame and send the EAP-Method data to the supplicant via EAPOL/EAP.

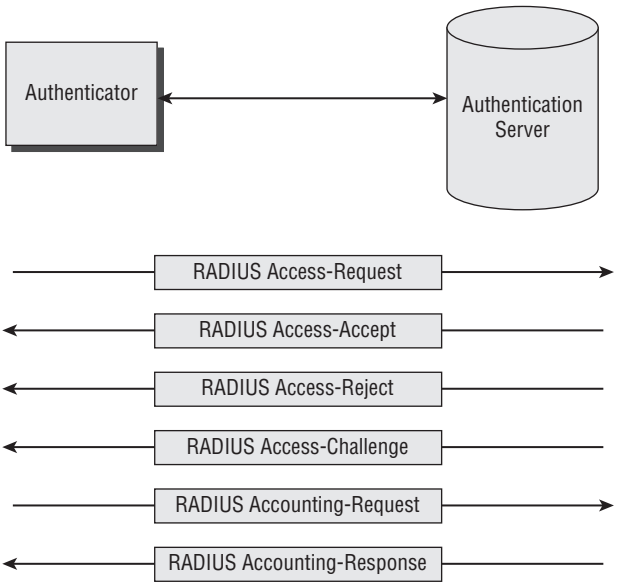

Figure 2-12: Communications between the authenticator and the authentication server

NOTE IETF RFCs 2865 and 3579 extend RADIUS into 802.1X, and RFC 3780 identifies RADIUS attributes.

After the EAP-Method results in deeming the supplicant either authorized or not authorized, the authentication server sends an applicable RADIUS Access-Accept or Access-Reject frame to the authenticator. The authenticator then issues the corresponding EAPOL Success or EAPOL Failure frame to the supplicant. At that point, if the request was successful, then the authenticator opens the port for the supplicant to have access to the protected network.

NOTE Refer to Chapter 4 for details on communications between the authenticator and the authentication server.

A Historical Perspective

The first IEEE 802 standard LANs came about in the early 1980s. As these networks began to proliferate and replace terminal-host mainframe systems, network component vendors and standards bodies were motivated to create a security gate to the networks. At first, this wasn't too important for wired corporate networks because the walls of the facility offered physical access control. If a hacker couldn't get inside the building, then it was nearly impossible to connect to the network. The thrust toward port-based authentication started when employees started accessing the corporate network from remote locations, such as hotel rooms and homes. The opening of the corporate network to the Internet, to enable higher-speed connections than the dial-up telephone provides, required tighter access control. Thus, port-based authentication became a critical component, which drove the writing of related IEEE standards and IETF specifications.

Nowadays, nearly all corporate networks interface with the Internet; and despite the use of firewalls, fears remain that hackers can still get into the network. In addition, many companies and organizations have wireless LANs, either as their entire network or as an extension to existing wired networks. These wireless networks make it even easier for hackers to gain access to the corporate network. In most cases, a hacker can be sitting inside a car located in the parking lot of the company or even down the street in a hidden location.

The first big move toward port-based authentication specifications was the creation of the Extensible Authentication Protocol (EAP), approved in 1998 as IETF RFC 2284, titled "PPP Extensible Authentication Protocol." EAP provides communications between the client device being authenticated and an authentication server. As you learn more details later in this chapter, EAP is really just a point-to-point protocol that carries actual authentication elements. Specific EAP-Methods actually provide the authentication mechanism, such as definition of credentials. There are a few mandatory EAP-Methods that EAP must support, but there are many proprietary EAP-Methods.

Another big step came in 2001 when the IEEE ratified the 802.1X standard (often referred to as 802.1X-2001). This initial 802.1X standard is based largely on EAP. In fact, 802.1X merely extends EAP to operate over LANs. 802.1X defines the EAP over LANs (EAPOL) protocol to accomplish this. 802.1X (and EAPOL) only applies to the interface between the client device being authenticated and the Ethernet switch or wireless LAN access points to which the client device is connecting. In 2004, EAP and 802.1X documents underwent significant revision, which resulted in RFC 3748 for EAP and 802.1X-2004. These are the most current versions and the basis for this book. RADIUS is another major component of a port-based authentication system. RADIUS was more formally introduced into port-based authentications documents around 2003.

Today 802.1X, RADIUS, EAP, and EAP-Methods are fairly well coupled through formal standards and specifications. This leads to the deployment of secure port-based authentication systems that provide much better interoperability than was possible in the earlier days.