A[1] Security Architecture for 802 Networks?

Mick Seaman

This note is a sketch of a potential security architecture for 802 Networks. Its principal purpose is to document the 'obvious'[2]. This draft is far from complete. It is written for those who are not security experts by someone else who is not, taking the view that this is a reasonable thing to attempt, and that a security project will not have succeeded unless its results can be explained and implemented[3] by those with little detailed understanding of security.

The proposed architecture is based on 802.1X taken together with the developments proposed to make it more useful and flexible in networks comprising both 802.11 and 802.3 LANs.

[1] "A" but not "the", there being some many different threats and perspectives on those threats.

[2] That is it contains little in the way of new ideas, not that it represents the only approach to the problem, a common view of what problem we are trying to solve, or does not need correcting.

[3] Though not designed.

Why link layer security

The first question that has to be answered is why have link layer security at all? Surely communication is best protected end to end[4], at the session or application level using IPsec[5,6] or related technology?

Our answers are that:

1) The session level communication traverses an intervening network, consuming resources on that network that a network owner might only wish consumed by authorized[7] parties[8].
2) Allowing communicating parties to transmit frames across an intervening network can expose the security weaknesses of the other systems[9] attached to that network to attack. Most common systems have countless points of weakness, some completely accidental, some intentionally introduced to support ‘efficiency’ or ‘plug and play’ operation.
3) Many LAN protocols that configure the network or the services delivered by the network are not intrinsically secure, and not all these protocols can be redesigned easily to use secure communications. Even if they were so redesigned it is doubtful whether satisfactory implementations would be available soon, or networks already operating could be migrated.

The nature of a solution

The identified needs (above) suggest the following framework, assumed problem, and security mechanisms. The rest of this note discusses them using 802.1X terminology as convenient.

A network owner creates and manages a secured network, that is a network that has a secure perimeter. Within the secured network no constraints are placed on the communications between stations[10], but any station external to the secured network (a *Supplicant*) has to petition the owner to be allowed access. Access to the secured network is provided through *Network access ports*, each equipped with *Authenticator* functionality and capable of denying or restricting access. Each Authenticator has little local configuration information but uses the services of an *Authentication Server*, provided and managed by the network owner.

Of course this model, illustrated in Figure 1, was deliberately chosen to be very similar to that used to control dial up internet access[11], and has the general flavor of a firewall solution[12].

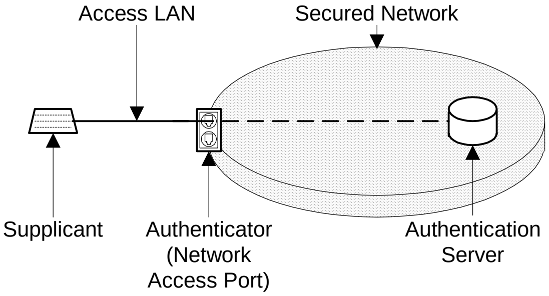

Figure 1 - Framework

[4] Either bi-directionally, or within a closed group of systems.

[5] When 802.1 began work on .1X there was quite a body of opinion in support of the notion that the only true security was application level security, and that the proposed work was irrelevant given the imminent deployment of comprehensive application level security frameworks.

[6] While this objection is not as common as it was, it does have some merit and is not completely dealt with by the refutation given. In certain scenarios it is possible that protecting only that traffic which can be covered by end to end IPsec provides an appropriate level of security.

[7] Or at least parties that can be charged

[8] These resources may be consumed even if one of the parties supposedly involved in a session wants no part of the communication

[9] These may or may not be readily identifiable as the intended destination of frames transmitted by malicious parties.

[10] At least not by the security mechanisms discussed in this note.

[11] It differs in one very important respect. The supplicant does not have to transmit ordinary data frames in a special encapsulation (such as ppoE) to mimic dial up. Once it is granted unqualified access the network perimeter access is granted

[12] Though good firewalls are more elaborate system constructions.

A simple beginning

802.1X as first proposed[13] attempted to provide a practical solution to a simple scenario[14]. Ports allowing access to a LAN were to be placed in a semi-public space, e.g. a conference room used by visitors to a company, and access to the LAN would be controlled by a dedicated bridge port taking instructions from an authentication server for the company[15,16].

From the point of view of getting a result, i.e. producing a working specification, in a finite time this scenario is effective as it admits very few threats:

1) The human (let's call him or her 'H') who is attempting to connect to the LAN with a laptop (the supplicant) is presumed (if authenticated and authorized) to be well intentioned, moderately competent, and careful of the LAN and attached resources.
2) H can inspect the laptop, provide the physical cable from the laptop to the wall jack, and verify by inspection that no intruder is attached (there is no shared hub half way down the cable).
3) The wall jack is firmly attached to the wall, and intruders are not believed to have sufficient uninterrupted access to complete building works to insert equipment behind the jack.
4) The physical wiring behind the wall jack is believed to be wired into the bridge port controlling access with semi-permanent point to point wiring using the cable routing commonly used in office walls and cube layouts, so is not readily susceptible to accidental network additions by the company's own non-IT staff.
5) The company's own staff, who have direct physical access to the secured LAN, are presumed not to be directly interested in helping intruders access resources. To put it another way – if they are of mischievous intent the company has much more to worry about than controlling open access wall jacks in meeting rooms, since they can physically access all sorts of equipment in the network directly.

[13] It became more complicated or sophisticated (depending on your point of view) fairly rapidly as Radius and EAP were leveraged, but I claim authoritative knowledge of the first proposal to .1, initiated by Vipin Jain. See US patents 6,021,495; 6,311,218; and 6,367,018, for some background and focus on concerns at the time.

[14] It is worth discussing the scenario because one of the obstacles to practical security, at a cost a customer is willing to bear, is excessive generality in the solution. Curiously at the time this generality was advocated as supporting ports on shared media hubs, to lower physical equipment costs. The other major obstacle to getting something done was the view that application level security was the final and only true answer.

[15] Use of LDAP and individual or person object data (I forget the jargon) was high on our list of potential "authentication" methods at the time.

[16] From the outset (and for some time previously) allowing different users with different authorizations to access different VLANs was considered. Since a large number of distinctions could be made and significant apparatus invented for controlling VLAN access invented it was deemed practical to leave this question alone. Given the existence of 802.1X is not now (I hope) at issue, it might be practical to revisit this issue.

Some additional threats

Amongst the many threats that do not occur in our first simple scenario are:

1) Intrusion between the Supplicant and the Authenticator/Network Access Port for:
   - Traffic analysis.
   - Eavesdropping.
   - Replay of data (frames).
   - Modification of (data frames).
   - Insertion of spurious data (frames) on established connections/sessions.
   - Removal of selected data (frames) on established connections/session.
   - Masquerade to establish additional communications/connections/sessions.
   - Theft of credentials for future masquerade.
2) Masquerade as a legitimate or expected Authenticator/Network Access Port to deceive the Supplicant into:
   - Revealing confidential data.
   - Revealing credentials that can be used for future masquerade as the supplicant.
   - Accepting data that may be used to attack or compromise the supplicant's system.
   - Communicating to the expected systems through the masquerading Authenticator and thus encountering threats listed under (1) above.
3) Masquerade as a legitimate or expected Authenticator/Network Access Port to deceive the Authentication Server[17] into:
   - Facilitating an attack on the supplicant, as per (2) above.
4) Intrusion between the Authenticator/Network Access Port and the Authentication Server to:
   - Modify the communication to the Authenticator to support access to the secured network by an unauthorized supplicant.

A further set of threats is omitted from our scenario analysis because the supplicant is not concerned about them, or is taking separate defensive measures:

5) Protection of the supplicant against data from the expected[18] secured network that attempts to attack or compromise the supplicant's system.

Simple scenario benefits

The simple scenario described has a simple solution with some nice properties that we would do well to retain in more complex scenarios when and if possible:

1) Communications on the secured network do not have to be encrypted or otherwise protected. Most systems function exactly as they would without security. Existing frame formats are not modified.
2) Communication between the Supplicant and the Authenticator does not have to be protected, and frame formats are not modified.
3) Communication between the Authenticator and the Authenticating System does not have to protected, and frame formats are not modified[19].
4) The behavior of the Authenticator can be made very simple as it can act on simple "Authorized"/"Unauthorized" commands sent in clear from the Authenticating System and otherwise simply has to distinguish the authentication/authorization dialogue that it relays in Unauthorized state from the other frames that it must discard[20].

[17] A direct threat to data within the secured network is only possible if the intruding/masquerading system is attached to the network, so many threats that might be expected to occur under this heading do not appear. Once surreptitiously attached to the secured network an intruder can do many evil things.

[18] "expected" in the sense that it is the network that the supplicant intends to connect to.

[19] The Authenticating System is within the perimeter of the secured network, as shown in Figure-1. A modest change to this scenario uses a secure tunnel beginning within the secured network to securely convey communication to and from an Authenticating System outside the perimeter of the secured network.

[20] Sadly the initial 802.1X Authenticator became rather more complex than this ideal, which has it transparently passing frames, just snooping on authorization results.

Mutual authentication

It is possible, even likely, that the system with supplicant functionality will wish to authenticate and authorize the network access port it is trying to use[21].

Since the security architecture is designed to protect against unauthorized access to the resources that a network access port is safeguarding, each Authenticator may choose its Authentication Server independently, as in Figure 2. To put this another way : just because your boss says I can use his network, doesn't mean that my boss will let you have access to hers.

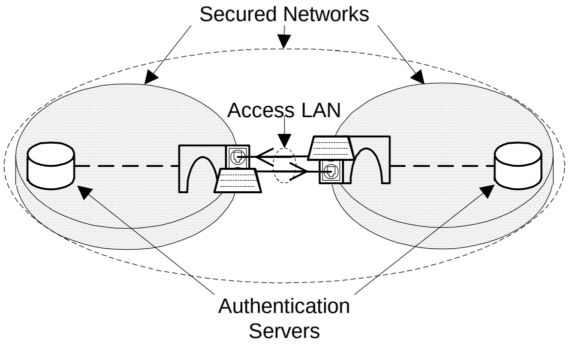

Figure 2

A growing network

If an authorized network access port is allowed to relay all data traffic[22], it effectively includes the supplicant within the perimeter of the secured network. If the supplicant is associated with a Bridge Port, with other ports on that bridge providing network access ports then the perimeter of the secured network can grow, as Figure 3 illustrates.

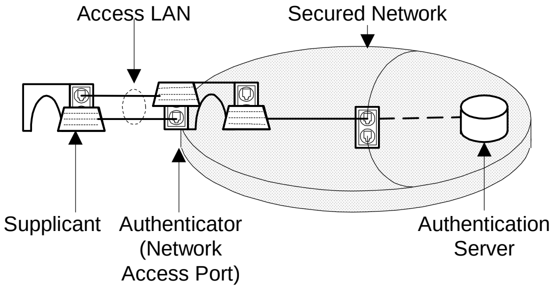

Figure 3

In this scenario it is likely that both bridges, the one initially at the edge of the secured network and the one seeking to participate in that network, will wish to authenticate each other so that neither is accepting frames from or sending frames to an unauthorized party. The bridge that is closest to the authentication server controls the path to that server, and has to authenticate and authorize the other before the latter can proceed. Fortunately no special sequencing is required in the protocol, just suitably long retry timers and counters.

[21] In the simple scenario this was unnecessary since H knows, or at any rate thinks he knows, which network he is trying to access, and trusts that is not malicious.

[22] Or almost all, given that there is a role for some filtering in a network that has no aspirations to security.

Segregating traffic

When the authenticator in a bridge allows the traffic from a supplicant into a network it may place restrictions on that traffic. Frames for certain configuration and routing protocols, e.g. RSTP, can be discarded. In general all the controls that a firewall might impose can be applied at a network access port.

Traffic can also be allowed into a network, but kept separate from traffic originating within a more tightly controlled perimeter, or from traffic permitted from other network access ports. VLANs can be used to segregate traffic as long as all the systems within a network perimeter can be trusted not to bridge traffic between VLANs. Since VLANs are already widely deployed and it is not trivial to invent and deploy a new segregation scheme they have to be a method of choice. It is useful to think of VLANs as providing a number of separate 'planes' within a network perimeter, with not all planes accessible through network access ports at the perimeter.

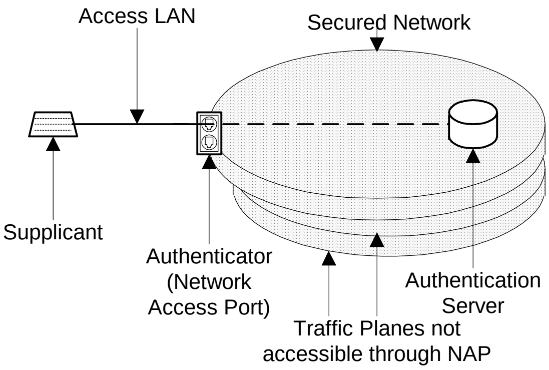
Access LAN
Secured Network
Supplicant
Authenticator (Network Access Port)
Authentication Server
Traffic Planes not accessible through NAP

Figure 4

For such an approach to be deployable in a way that meets the requirements of the approach so far, the controls applied by each Authenticator and its use of network planes have to be dictated by the Authentication Server on the basis of the Supplicants identity. Radius and Diameter servers, with their ability to supply the Authenticator with additional parameters provide a convenient, if not necessarily secure[23], way to do this.

Insecure access link scenarios

The three party framework proposed (Supplicant / Authenticator / Authenticating System) and the simple scenario described above can be used as the starting point for a whole range of scenarios depending on the additional threats each attempts to address.

Scenarios that just include threats from the first two categories, i.e. intrusion between Supplicant and Authenticator, or masquerade as an Authenticator (to the Supplicant), form an interesting subset, illustrated by Figure 4[24].

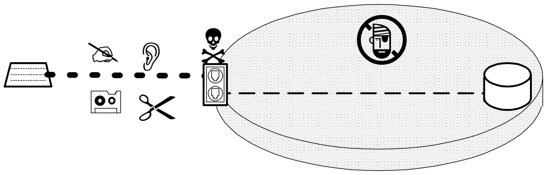

Figure 5

In pure form these scenarios imply that data and control frames need not be encrypted or protected within the secured network. The protective security measures introduced to secure the access link can be custom designed to fit the threats admitted for a particular access technology. Policy protocols within the secured network can be used to control the authenticator without being fully secured themselves.

[23] Big red flag.

[24] Note that a man in the middle between the two networks shown in Figure 2 does not have to interfere with the authentication and authorization dialogue to eavesdrop, modify, or inject data traffic. Simple access to the link is enough if both conjoined networks and their connecting link transfer data in clear.

Insecure connectivity scenarios

Unfortunately in wireless networks, where the infrastructure itself is wireless, the scenario is more likely to resemble Figure 6.

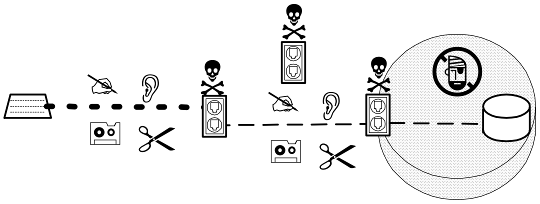

Figure 6

Three general approaches can be used to secure such networks:

In the first the link nearest the already secured portion of the network is secured first, then links attaching to that are secured. This is basically the approach discussed in "A Growing Network" above.

In the second each authenticator sets up a secure tunnel, probably using IPsec, to its authentication server. The authenticator doesn't care whether the tunnel passes through a network that has already been secured, in whole or in part.

In both these cases the problem has been reduced to that of an insecure access link. In the third approach the authentication and authorization dialogue assumes nothing about the security characteristics of the connectivity between all three parties – Supplicant, Authenticator, and Authenticating System. The Needham—Schroder private key method and its derivatives such as Kerberos[25] or Otway—Rees are examples of such an approach. There is a lot to be said for such an approach since it is less vulnerable to 'weakest link' failure[26].

Universal or targeted solutions?

A very interesting question is the degree to which it is desirable or possible to strive for a universal 802 wide solution to the problem of securing layer 2 networks. The threats perceived by each MAC technology vary as the opportunities for physical intrusion vary and the need to understand the threats[27] to be countered when designing a solution is fundamental to security technology. Moreover the devices that have to implement a security solution vary, from a laptop PC to an OLT (Optical Line Terminator). A universal solution would impose an additional burden on all devices, as well as inappropriately (from someone's point of view) distributing the burden between Supplicant, Authenticator, and Authentication Server.

However the effort in producing one design that is even passably secure is large, especially if the limits of its capabilities are to be documented properly. While the use of technology and frameworks developed outside the narrow confines of a single MAC group makes those who want complete control over the system they are designing nervous, it seems inevitable.

Once the "end to end conversation protection" is admitted not to be part of the layer 2 problem space, the business requirements of all the MACs show strong similarities. They all need to form useful parts of a bridged network without permitting unauthorized parties to eavesdrop, inject traffic, or steal service. The network needs to be able to contain physically secured regions where frames are transmitted in clear, otherwise the security solution won't be universally deployed. The security policy for the network needs to be capable of being administered from existing policy systems[28], not from a completely new system that is to be built from scratch[29].

Avoiding weak links in the authentication and authorization processes themselves argues for the use of some Needham—Schroder derivative, extended to allow the rapid reuse of acquired credentials for a period to facilitate roaming.

[25] Kerberos solves the problem of liveness of keys in basic Needham—Schroder, but introduces a dependency on NTP which seems fatal in our application unless we are really assuming a secured network backend – but even this assumption does not address a Supplicant's exposure to a rogue authenticator that has hijacked an old key. Otway—Rees is a more elegant solution to the key liveness problem, but would seem to lack suitable backend and infrastructure support today.

[26] All approaches involving concatenation of anonymously relaying networks with firewall type security are open to weakest link criticisms. The only way to get real conversational security is between the ends of the conversation, but as described above that is not the practical problem we face at layer 2.

[27] And the cost of those threats, to calculate the allowable expenditures and inconvenience of the counter measures.

[28] Hand carrying keys to each end of a secured link is not a serious solution for any MAC technology that aims for mass deployment. Similarly securing a link without making the identity of the ends known for the purposes of policy administration means that deployment in public networks will continue to be dogged by very high operational expenditures that dwarf the advertised cost of the equipment.

[29] There are too many failed policy protocols and frameworks already, the degree of dictatorial force required to avoid attracting the authors of these to any new effort is not available to us.

This is it so far, sections to be written:

- *Describe a secured network comprising a set of links each individually secured with mechanisms sufficient for threats to that technology*
- *Contrast that with an approach where the union of threats (including intruding bridges/network access points) is considered and protection is applied over a number/region of the links in one operation*
- *Look at the types of system playing Supplicant and Authenticator roles to formulate a sense of the balance between (a) specifying/implementing a single protocol solution and (b) picking two or more threat scenarios to make some cases easier (one of the cases being .1X as of today)*
- *Consider the parameters that a network access port needs to be able to place all or part of the incoming traffic on (an) appropriate plane(s), including constraints on how much traffic ought to be admitted (at what priority). Discusses the use of Radius (or Diameter) for distributing these parameters in a way that is tied to authorization, as well as some alternatives.*
- *Further augment the model to include supplicants roaming between network access points.*
- *Provide a brief (and not necessarily accurate) introduction to Needham-Schroder methods just to summarize the idea, and support a model of roaming.*

Enough for now.