Available online at www.sciencedirect.com
ScienceDirect
Journal of Network and Computer Applications 30 (2007) 900–919
Journal of NETWORK and COMPUTER APPLICATIONS
www.elsevier.com/locate/jnca

A network access control approach based on the AAA architecture and authorization attributes ☆

Gabriel López[a,*], Oscar Cánovas[b], Antonio F. Gómez[a], Jesús D. Jiménez[a], Rafael Marín[a]

[a]*Department of Information and Communications Engineering, University of Murcia, 30071 Murcia, Spain*
[b]*Department of Computer Engineering, University of Murcia, 30071 Murcia, Spain*

Received 28 July 2005; accepted 28 July 2005

Abstract

Network access control mechanisms constitute an increasingly needed service, when communications are becoming more and more ubiquitous thanks to some technologies such as wireless networks or Mobile IP. This paper presents a particular scenario where access rules are based not only on the identity of the different users but also on authorization data related to those users. In order to accomplish this general goal, it will be necessary to add to the traditional system-specific services for authentication and authorization, and also some entities able to manage the information related to identity, roles and permissions. Network access will be based on the 802.1X framework and the Authentication, Authorization, and Accounting (AAA) architecture, as they constitute the basis for most of the existing proposals for limiting the access to a restricted network. These proposals will be extended making use of an authorization infrastructure based on SAML statements, the RBAC model, and XACML as the main language for expressing authorization policies. The solution that we present in this paper is a consequence of an exhaustive and non-trivial analysis of the different mechanisms that could be used to provide this kind of service. As we will see, the correct integration

☆Partially supported by Euro6IX Project (IST-2001-32161) and SEINIT Project (IST-2002-001929). A preliminary version of this paper appeared in the Proceedings of the 19th IEEE IPDPS, The First IEEE International Workshop on Security in Systems and Networks (SSN'2005).

*Corresponding author.
E-mail addresses: gabilm@dif.um.es (G. López), ocanovas@um.es (O. Cánovas), skarmeta@dif.um.es (A.F. Gómez), jdjimenez@dif.um.es (J.D. Jiménez), rafa@dif.um.es (R. Marín).

1084-8045/$ - see front matter © 2005 Elsevier Ltd. All rights reserved.
doi:10.1016/j.jnca.2005.07.010

of these different mechanisms leads to the definition of a scalable and versatile network access control system which conforms to the guidelines outlined by the AAA initiative.
© 2005 Elsevier Ltd. All rights reserved.

Keywords: Authorization; Access control; Attributes; SAML; XACML

1. Introduction

During the last few years, it has become an increasing concern to the network administrators about how to control the users that are making use of computers networks. This situation has been especially compounded by the proliferation of wireless networks and the discovery of serious security vulnerabilities related to these types of networks. As a direct consequence of these facts, several security technologies have recently emerged in order to provide access control mechanisms based on the authentication of users.

Traditionally, network access systems have been based on login/password mechanisms to authenticate the different users requesting a network connection, which provides a very limited degree of security. Other systems follow a more advanced approach for mutual authentication based on X.509 identity certificates (Housley et al., 2002), therefore offering a stronger security solution which makes use of public key cryptography. These systems are especially useful for those Internet Service Providers (ISPs) which are concerned about the real identity of the requestor as a key element in order to make a decision.

There are other organizations where the different members or users are classified according to their administrative tasks, the type of service obtained, or some other internal requirements. For example, in a university environment, users can be part of the *administrative staff*, *professors*, or *researchers*. In industry, we can easily find hierarchical relationships comprising *employees*, *managers*, *CEOs*, etc. Even ISPs classify their customers according to the different type of contracts (*domestic service*, *business service*, *premium service*, *etc*.), which involves different types of connection properties and services.

In these previous scenarios, the identity could not be sufficient to grant the access to the resource being controlled, since we should know the role being played by the user in order to offer the right service. Therefore, we need a system able to grant to the different users the set of attributes specifying those privileges or roles. This kind of systems is usually designed following the principles of the Role-Based Access Control (RBAC) model (Ferraiolo et al., 2001). In this way, when end users request a resource or service, the decision is taken according to the set of attributes assigned to them. For example, an ISP might state that only users showing a *premium* attribute will obtain a particular quality of service or network bandwidth.

It is worth noting that authentication is based on X.509 identity certificates and authorization is based on attributes, and therefore they constitute two completely independent operations. However, they are usually related since authentication represents the first stage of most access control systems, i.e., obtaining the user's identity. Once the identity has been securely established, it is necessary to infer whether the user is authorized to make use of the network.

Different authorization proposals can be used in the above-mentioned application scenarios, as for instance SPKI (Ellison et al., 1999), X.509 AC (Farrel and Housley, 2002)

or SAML (Maler et al., 2003). Indeed, there are several projects that make use of these proposals, such as Liberty (Beatty et al., 2003) or Shibboleth (Cantor, 2005), which have extended some well-known authentication solutions in order to provide authentication mechanisms in Web environments.

This paper presents a network access control approach based on X.509 identity certificates and authorization attributes, and addresses some of the challenges derived from the integration of existing authentication systems and a flexible, scalable and manageable authorization system. Our proposal is based on the SAML and the XACML (Godik and Moses, 2005) standards, which will be used for expressing access control policies, authorization statements, and authorization protocols. Our authorization proposal is mainly based on the definition of access control policies including the sets of users pertaining to different subject domains which will be able to be assigned to different roles in order to gain access to the network of a service provider, under specific circumstances. These policies are central elements in our system, and require the definition of some entities responsible for managing their lifecycle. Moreover, our starting point is a network scenario based on the Authentication, Authorization and Accounting (AAA) architecture (de Laat et al., 2000), where we centralize all the operations related to authentication, authorization, and accounting.

The rest of this paper is structured as follows: Section 2 describes the main scenarios where attribute-based authorization is recommended and provides an overview of the different requirements imposed by our application scenario and the decisions that we adopted. Section 3 describes the application scenario and the proposed architecture for authorization. Section 4 presents the two different design alternatives based on push and pull models. Then, Section 5 presents how the proposed architecture can be adapted to the described scenarios. Section 6 includes the related work that informed our research. Finally, we conclude the paper with our remarks and some future directions derived from this work.

2. Requirements imposed by the different scenarios

Adding an attribute-based authorization mechanism to a network access service involves a set of requirements that must be addressed by every component of our system. On the one hand, the network infrastructure should offer standard access mechanisms, which should also be extensible in order to incorporate authorization mechanisms. Furthermore, these access mechanisms should impose minimum requirements for the end users, i.e. they have to be as transparent as possible.

On the other hand, the authorization proposal should be adaptable to a network infrastructure clearly based on an entity managing all the access requests. It has to consider the different requirements imposed by end users to exchange authorization credentials, and also, it has to be flexible to be integrated in the existing protocols which are being used to provide network access. In addition, it has to provide support for inter-domain scenarios, where the authorization information might be exchanged between different administrative domains.

Finally, in these inter-domain scenarios, it will be necessary to define some kind of agreement among the participating domains concerning the management of the different end users, network services provided by each organization, and other issues like authority recognition.

2.1. Network access scenarios

There are several scenarios where an attribute-based authorization model, together with an authentication model, if needed, can improve the access control management. The authorization model can be used in an intra-domain environment, where users and resources belong to the same organization, or in an inter-domain environment, where users pertaining to a home domain request resources located in a foreign domain. In fact, depending on the number of organizations involved and the requirements imposed by such organizations, like anonymous access, privacy protection, etc., we can identify some of the different scenarios described below.

2.1.1. Intra-domain scenarios

In this kind of scenarios there is only one organization involved, which must protect the resources against both unauthorized well-known users and unknown users.

In a campus environment there are a wide number of people who play different roles depending on their function inside the organization. There are students, professors, administrative staff, researchers, etc., and all of them are registered as valid campus members.

On the other hand, there are also a wide number of available services or resources that could be requested by the campus users, like *library*, *research and administrative databases*, *X500 public directory*, *Internet services* (*web*, *ftp*, *etc.*), *wireless network connection*, *Internet public connection* (*kind of*), etc.

Obviously not every user can access all the services, and every service cannot be accessed by all the users either. For example, the *X500 public directory* or *Internet public connection* are public services that could be used by every campus member, but the *research-database* can only be queried by *researchers* and *professors*, or the *administrative-database* can be only accessed by *administrative staff*. Finally, a good *quality-of-service* (*QoS*) *network connection* can be offered to *researchers* members, whereas a medium *QoS connection* can be offered to the rest of the members.

To control the users requesting access to a service, an Authentication Authority is necessary, which asserts that the user is a campus member. However, in order to ensure that the user can gain access to the requested resource, the authentication process is not enough, and an Authorization Authority becomes necessary. Based on this assumption, when a member tries to gain access to the resource, he is first authenticated using, for example, a public key certificate, a shared secret or a login/password mechanism. Then he is authorized based on the user's role(s) he plays, that is, the user's attribute(s), and a resource access policy defining the access control requirements.

There are other scenarios where previously unknown users try to gain access to resources protected by an organization. This is the case of an airport where people pay for a network connection while they are waiting in the boarding area. In this scenario, the network access providers might force users to obtain a ticket in a vending machine or in a similar way. This ticket can be considered as an authorization ticket rather than an authentication ticket.

Now let us suppose that users can use the vending machine to request different kinds of network connections, depending on *network bandwidth*, *security level*, *QoS options*, *etc.* In this situation the network ticket issued by the vending machine must not only specify that

the user is authorized to use the network but should also specify some features that the network provider must apply.

2.1.2. Inter-domain scenarios

More complex scenarios appear when more than one administrative domain is involved. That is, two or more different organizations need to share resources depending on the user's attributes.

In an inter-campus scenario, two or more campus domains need to share resources. For example, students of campus A can use the Internet connection in campus B but, for example, with different QoS in relation to students of campus B.

Domains maintain local Authentication and Authorization Authorities. Authentication could be delegated, based on login/password, shared secrets or in a public key infrastructure. When an end user, pertaining to a particular campus, requests a resource in a foreign campus (after an authentication procedure, if needed) the target domain must be able to obtain the user attributes from his home domain. Furthermore, the resource access policy defined in the foreign domain should know how to interpret these attributes since the authorization will be based on that information. This inter-domain scenario requires a Service Level Agreement (SLA) (Blake et al., 1998) between the participating domains.

In inter-enterprise scenarios, like the one described above, are there are two or more organizations that need to share resources based on mutual business agreements. For example, an employee might visit the foreign organization for a business meeting or for a temporary collaboration. In this situation the resource access control mechanism is required to prevent visitors from accessing restricted information or using unauthorized services. Once the employee is authenticated, the foreign domain needs to ensure he is authorized to gain access to the requested resources, that is, a visitor *manager* can have access to the *corporative network*, but a *subordinate* visitor can have access only to a *restricted network*. These types of relationships are usually less stable than the campus-based ones, and in some cases, it is also desirable to conceal visitor identity.

2.2. Network requirements

The solution presented assumes that each administrative domain should address the following requirements.

2.2.1. AAA server

The AAA architecture defines a central element that must be present in every domain, the AAA server. It is responsible for receiving and processing authentication, authorization, and accounting requests related to end users. In our solution, authentication is a mandatory stage, although is beyond the scope of this paper as to how to integrate the authentication process with some identity management systems like PKIs (Public Key Infrastructures). On the other hand, accounting represents a future direction that will be addressed in future works. Indeed, the AAA standard does not impose that an AAA server must provide support for these three operations, since any of them can be optional.

In relation to authorization, additional requirements are needed: the AAA server must have a module (an application-specific module or ASM) that will be responsible for managing the authorization attributes, and another one for obtaining authorization decisions.

Finally, it is worth noting that the AAA working group has also defined transport mechanisms that will be used to exchange authorization information in inter-domain scenarios.

2.2.2. Network access technologies

In a network access scenario, we have to provide to the end users some mechanisms to query a Network Access Point (NAP) regarding whether they can access the network. This mechanism should provide a high degree of security and be extensible in order to incorporate an authorization system. 802.1X (LAN MAN Standards Committee of the IEEE Computer Society, 2001) and PANA (Jayarama et al., 2005) are two different existing frameworks that can be used as network access technologies. Whilst 802.1X is a de facto standard that can be found in most of the current networks, especially wireless networks, PANA is a promising work in progress. Although any of these solutions might be used in our system, this work is based on 802.1X as the main technology used to communicate end users and NAPs. 802.1X was designed to enable easy integration with a protocol able to exchange generic authentication information; this protocol is named (Blunk and Vollbrecht, 1998) and supports different authentication methods called EAP methods. As we will see later, there are some tunneled EAP methods (i.e. PEAP (Anderson et al., 2004)) that can be used to transport generic data inside. Extensions to these tunneled methods are required to exchange authorization data.

2.2.3. Transport of authorization data

Our solution requires a protocol able to transport the authentication, authorization, and accounting requests from any service point, for example an NAP, to the AAA server. Just like for network access technologies, we can find several proposals fulfilling this requirement, such as RADIUS (Rigney et al., 1997) or DIAMETER (Calhoun et al., 2003). Both of them provide mechanisms to exchange EAP packets between NAPs and AAA servers. However, whilst RADIUS is the most widely deployed standard, DIAMETER provides a high degree of flexibility that can be used more efficiently to address the requirements of our application scenario. There are several reasons why we based our work on DIAMETER. RADIUS has many problems and lacks features that support roaming and mobility requirements; they are mainly scalability, security problems in untrusted proxy environments. Since this protocol only supports weak hop-by-hop security, it does not define data-object security mechanisms as well as well-specified agent support. Additionally RADIUS was originally designed for small networks supporting just a few end users and a specific set of access-control technologies. In comparison, DIAMETER was designed from the beginning to support roaming and mobility, and it is based on scalability and security premises: it has an explicit support for agents by providing scalability and strong hop-by-hop security based on IPsec (and optionally TLS), reliable transport, etc. Additional advantages of using DIAMETER can be found in Blunk and Vollbrecht (1998).

2.3. Authorization requirements

An attribute-based authorization system for a network access environment must satisfy the following requirements.

2.3.1. Authorization specification

Nowadays, there are several proposals to represent and to manage authorization statements. X.509 Attribute Certificates define an extension to the X.509 standard that can be used to link any type of attribute to an entity, and proposals such as SPKI and SAML also provide some statements that can be used to express not only attributes but also authentication proofs and authorization decisions.

Our application scenario requires an authorization specification that must be standard, widely accepted, and suitable for current systems. Therefore, SPKI does not constitute the right approach, despite being a flexible and suitable solution; it is not widely deployed or accepted. On the other hand, our application scenario is based both on authorization attributes, for example user roles, and authorization decisions, and, additionally, we also need a mechanism for expressing authorization queries and responses. This requirement is not fulfilled by X.509 Attribute Certificates, since they can only be used to express attributes.

However, SAML (an XML-based standard) does provide a flexible solution which is being widely used more and more in Web environments (Beatty et al., 2003; Cantor, 2005). Moreover, SAML provides transport mechanisms to exchange authorization data between the different entities composing our system.

As we will see, SAML statements are exchanged between the different elements, even in multi-domain scenarios, using DIAMETER or EAP, which will require their extension in order to encapsulate the statements.

2.3.2. Authorization policies

When an end-user requests a network connection, our system has to obtain a decision based on the attributes related to that user. Therefore, we have to express in some way the set of privileges related to the users having particular attributes. The document containing this type of rules will be referred to here as a *resource access policy*.

On the other hand, the role assignment rules, i.e. which users can obtain which attributes, must also be expressed in a policy document that will be used afterwards to create the SAML attribute statements. This document will also contain additional information, such as the validity period related to the attributes, the set of resources controlled by these attributes, etc. The document containing these rules will be referred to as a *role assignment policy*.

In order to represent these policies, we can find several alternatives in the literature. First, some existing systems develop their own specification (Community Authorization Service, 2004). Second, other systems are based on XML to define a new XML schema for expressing their own authorization policies, such as Akenti (Akenti Distributed Access Control, 2004) or PERMIS (Chadwick et al., 2003). Finally, we can also find some works making use of XACML, an XML-based standard, to represent access control policies, since this de facto standard was specifically designed for this purpose.

In our system, we use XACML not only to express the role assignment policy or the resource access policy but also to encode the authorization queries and responses, which can be easily integrated with SAML statements (Anderson and Lockhart, 2004). Additionally, the use of policies involves the necessity of an authorization engine able to process role assignments, role hierarchies, authorization privileges, resources, and obligations. It also has to verify time constraints, complex conditions, and inter-domain relationships (also known as recognition policies).

2.4. Inter-domain requirements

When an end user pertaining to a particular domain is requesting a network connection in a foreign domain, the target service must be able to obtain the user attributes from his home domain. Furthermore, the resource access policy defined in the foreign domain should know how to interpret these attributes since the authorization will be based on that information. This inter-domain scenario requires a *Service Level Agreement* (SLA) between the two participating domains. This SLA should express the way to which both domains will deal with the authentication and the authorization of their users. It is worth noting that, in our system, authentication in inter-domain scenarios will be delegated to target domains, i.e. there is no need for further exchange of information among the different AAA servers in order to authenticate the user since it is reasonable to assume that two cooperating domains will be cross-certificated, will make use of a bridge CA, or will use a similar solution.

3. Architectural elements

Once we have analyzed all the requirements imposed on our solution, it is worth noting that the integration of these elements constitutes an interesting, open, and challenging research field. According to these requirements, in this section we introduce the system architecture and the participating elements.

In our application scenario, every end user belongs to a home domain, where he is given a set of attributes stating the roles he plays. When the end user requests a network connection in a particular domain, the request is obtained by the AAA server, and it makes a query to obtain the attributes linked to the user from an authority responsible for managing them. Finally, the AAA server sends an authorization query to a policy decision point, and that element provides an answer indicating whether the attributes satisfy the resource access policy. Furthermore, this policy can also establish the set of obligations derived from the decision, for example some QoS properties, bandwidth, security options, etc. As we will see, this general scheme works both in single and inter-domain scenarios.

As Fig. 1 shows, the main components of our architecture are as follows:

- *End user*: Is the entity requesting the access to the network. The end user, who pertains to a source domain where he can play one or more roles, will try to gain access to a network making use of these roles. This process starts once the user has connected to a switch port, or has been associated to a wireless access point. In our scenario, the user must have a valid identity, i.e. an X.509 authentication certificate issued by the CA related to the source domain, and some attributes that can be used to determine the access rights. It is worth noting that the user should be able to exchange some messages with some of the authorities before accessing the network, which involves the user being initially placed in a restricted network, e.g. a Virtual LAN, containing the set of entities responsible for access control management.
- *AAA server*: According to the specifications of the AAA architecture, an AAA server is responsible for receiving and processing the authentication, authorization and accounting information related to the users trying to gain access to the network. It is also responsible for querying other entities in order to retrieve any additional information that could be needed in order to perform its task. In our application

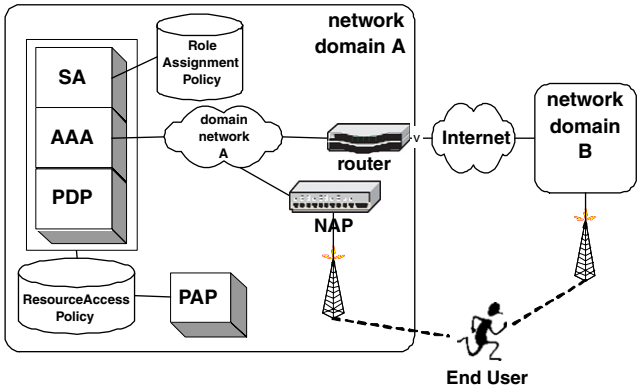

Fig. 1. Architectural elements.

scenario, an AAA server is used to manage the network itself, which makes use of DIAMETER as the transport mechanism. Each AAA server contains two different application-specific modules (ASM) which are responsible for dealing with generation of SAML statements (Source Authority), and authorization decisions (Policy Decision Points).

- *Source authority* (*SA*): This ASM manages the assignment of roles to users. The SA will receive requests, always through the AAA server, from two different entities. On the one hand, when the Push model is used, the user has to contact the SA to obtain his roles before requesting the access to the network. On the other hand, using the pull model, the local or foreign AAA server will request the SA for the user's attributes. Every SA has its own role assignment policy, expressed in XACML.
- *Role assignment policy*: This policy contains the rules governing the assignment of roles to users pertaining to a particular domain. It includes statements such as "in the source domain Source, the set of roles R1, R2. Rn can be assigned to the users contained in the $o = org, c = ES$ X.500 subtree for the validity period V". Therefore, this policy must contain a valid set of users, the set of roles that can be assigned to those users, and the conditions for the assignment.
- *Policy decision point* (*PDP*): This ASM is the entity responsible for generating the statements related to authorization decisions. Moreover, this element interacts with another central element of an AAA server, the policy repository where the resource access policy is stored. In an RBAC environment, a PDP has to obtain the role(s) assigned to the user since the access control policy is expressed in terms of roles. In the pull model, this can be done by querying the user's SA, using the AAA infrastructure. Following the push model, the attributes will be presented by the users. Finally, the PDP will generate an authorization decision statement regarding all the collected evidences. This statement specifies whether the user is allowed to access the network, and the set of obligations for the NAP in order to enforce the decision.
- *Policy administration point* (*PAP*): This entity defines, signs, and publishes the resource access policy.
- *Resource access policy*: Is the policy defined by the PAP. It has to contain which subject domains can obtain access to which resources according to the role(s) previously

assigned, and also the obligations derived from that decision. The policy should contain statements such as "the users pertaining to the source domain Source, and playing the role R1, will get access to the network N1 with a QoS1". Therefore, the resource access policy should define the following elements: recognized source authorities, roles that can be assigned by each source authority, resources to be protected, properties or parameters of these resources, and roles that must be played in order to use these resources. This policy can also specify the relationships among the different roles (hierarchical or specialization). This policy is also expressed using XACML (details regarding policies are beyond the scope of this paper and will be presented in a future work).

- *Network access point* (*NAP*): This element has two main functions: first, it has to forward the client requests to the appropriate AAA server of the target domain; second, once the AAA has obtained the authorization decision, the NAP obtains and enforces the properties of the network connection. NAPs can be wired or wireless, and must support the EAPOL protocol to exchange authentication information with the end user, and the DIAMETER protocol to communicate with the AAA server.

According to the proposed scenario, the network administrator can make use of two different design alternatives based on the well-known push and pull models.

4. Design alternatives

Interactions among the different components described in the previous section depend on the requirements imposed by the user to gain access to the network. On the one hand, the end user can follow a pull approach, which requires minimum overload and is more suitable for limited terminals, such as PDAs or mobile phones. In this way, all the authorization tasks are performed by the network infrastructure. On the other hand, following a push model, the user can present a particular set of attributes to the system, following his preferred disclosure policy. The push model involves a certain support for selecting and transporting attributes from the end-user terminal, representing a more intrusive approach. As a consequence, we have to provide solutions to these two different environments. Furthermore, this support must be extended to single and inter-domain scenarios.

However, the two alternatives make use of the same authentication process. First, the user connects his computer to an available port or is associated to a wireless access point. In both cases, the NAP will be configured with 802.1X in order to perform user authentication. Therefore, the next step is the exchange of EAP packets between the user and the AAA Server. This server will enforce the use of EAP-TLS (Aboba and Simon, 1999) in order to authenticate the user making use of his identity certificate. The authentication process will be performed in a delegated manner, that is, there is no need for further exchange of information among the different AAA servers in order to authenticate the user since it is reasonable to assume that two cooperating domains will be trusted. Once the TLS handshake is finished, and the user has been authenticated, the sequence of messages depends on the selected design alternative.

4.1. Design alternative 1: Pull model

The first alternative provides to the user an authenticated and authorized connection, but in a transparent way since the management of the authorization data will be performed using a pull approach (that is, the PDP will recover all the information needed to make the decision, using the AAA infrastructure). A possible disadvantage is that the user will not be able to select the set of properties that must be satisfied by the connection (it will be determined by the policy). The main advantage is that the 802.1X software being used by the user does not have to be modified in order to provide authorization services. For the sake of clarifying, in this section we differentiate how the pull model can be used in single-domain and inter-domain scenarios.

In this proposal, the first step is the authentication of the end user by means of a handshake EAP-TLS. Once the AAA server has validated the identity certificate of the user (which requires an explicit recognition of the PKI related to the user's domain) it has to verify whether he has been authorized to make use of the network.

The AAA server has to make use of one of its application-specific modules, the Source Authority, in order to obtain the set of roles related to the subject before initiating the decision process. In a single-domain scenario, this interaction is performed by means of a Programming Interface (PI). The AAA server provides an *SAMLRequest* containing an *AttributeQuery*. This query indicates that the expected response must be encoded using *AttributeStatements*. It also includes information about the subject requesting the access, usually some identifying piece of data obtained from the user's certificate, and, optionally, the type of attributes expected.

Once the SA receives the query, it obtains the subject information and establishes, using the role assignment policy, the set of roles played by the subject in the source domain. These roles can also be based on information obtained from the *AttributeQuery*, such as the description of the resource being requested. In this way, the SA could select the roles that are more appropriate (for example, when the resource description is a network address, only the roles that have been recognized by the target network will be selected). Thus, the SA will generate a signed *SAMLResponse* message containing status information and an *AttributeStatement* with the user role(s) (Fig. 2).

However, in an inter-domain scenario, Fig. 3, when the user is trying to gain access to a foreign network, the foreign AAA server first has to discover how to contact the user's home AAA server, and then they have to use some transport mechanism to exchange attribute queries and responses. The contact information could be included in the policy issued by the PAP, in the section related to the recognized source authorities, as in (Akenti Distributed Access Control, 2004), or a discovery service could also be defined for that task (Cantor, 2005). Regarding the transport mechanism, we have defined and implemented an extension to DIAMETER that we call DIAMETER-SAML. The extension consists of new DIAMETER attributes which are used to encapsulate SAML payloads, following the same guidelines as those of other existing extensions, such as DIAMETER-NAS (Calhoun et al., 2004) and DIAMETER-EAP (Eronen et al., 2005). Once the foreign AAA server knows about the local AAA, it sends the attribute query request and waits for a response containing the user's attributes.

At this point, the target AAA server has all the needed information about the subject. The next step is to decide whether those evidences satisfy the *Resource Access Policy*. In order to do this, the AAA generates an *XACMLAuthorizationDecisionQuery* which is

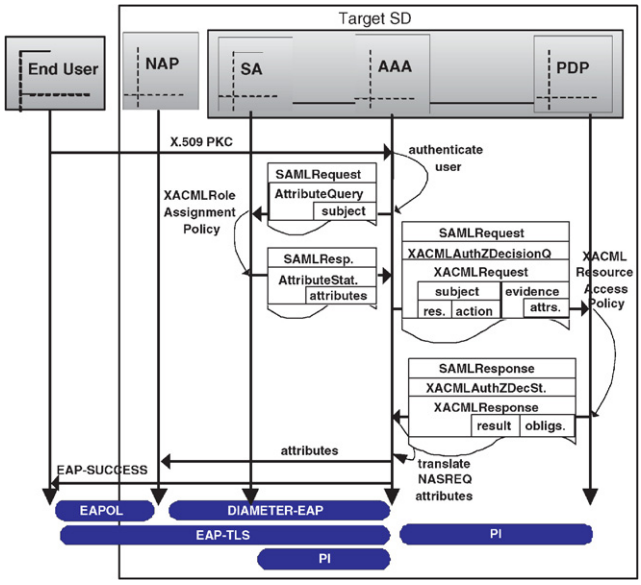

Fig. 2. Single-domain pull model.

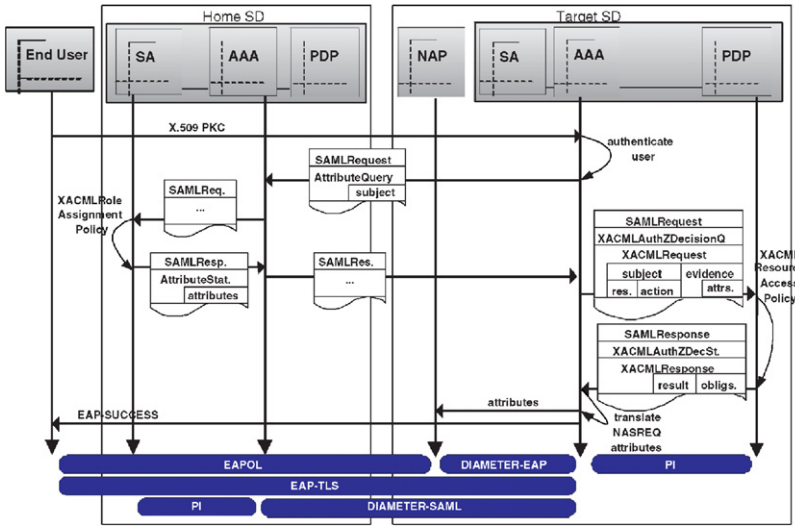

Fig. 3. Inter-domain pull model.

passed to the PDP in order to obtain an authorization decision. This sentence contains references to

- *The resource being requested*: Here the identifier must be expressed according to the encoding rules established by the policy.

- *The subject making the request*: Who can be identified using an NAI (Adoba and Beadles, 1999), X.500 names or emails.
- *Actions to be performed on the resource*: In this model, the default action requested by the user is determined after examining the policy since he is unable to specify the type of connection.

The target PDP has to verify the set of permissions given to the roles related to the user. If the requested access is part of these permissions the request will be granted. Once the decision has been obtained, the PDP has to generate an *XACMLAuthorizationDecisionStatement* containing the target resource, some information about the authorized subject, the permissions that have been granted, and the set of obligations derived from the decision.

In our application scenario for network access control, the set of controlled actions are determined by some of the attributes defined by DIAMETER NAS Application. This application defines generic attributes related to some network properties, such as the *NAS-Filter-Rule* to define filter rules that must be enforced in the NAP, *QoS-Filter-Rule* to specify filter rules related to QoS parameters, *Frame-Protocol* to encapsulate data, *Framed-MTU* to defined the maximum MTU, *Frame-IP-Address* and *Frame-IP-Netmask* to specify the IPv4 address and mask that will be assigned to the user, *Framed-Routed* to define routing parameters, *Tunnel-Type* to identity the tunneling protocols that can be used by the client (VLAN, PPTP, IP-IP), or *Tunnel-Medium-Type* to define the transport mechanisms that will be used to create the tunnels (IPv4, IPv6, 802, etc.). These attributes will be used in the resource access policy in order to express the set of obligations that must be enforced by the NAP.

Finally, the AAA server obtains the response and enables a network connection conforming to the obligations included by the PDP in the response. For this purpose, an exchange of DIAMETER NAS attributes must be performed between the AAA and the NAP.

This alternative provides a method for strong authentication of users, and a simple and transparent authorization service based on SAML. Since the client software does not have to be modified and the current implementations of the 802.1X framework do not have support for specifying the type of connection being required or the set of attributes that can be used as evidences in order to gain the access, this alternative has the disadvantage of providing no control to the user about the type of service required. In our opinion, this need not be seen as a disadvantage in most of the existing environments where a default access is being provided or where the users do not wish to become involved in authorization issues. However, for those environments requiring more fine-grained control we also provide a second alternative based on a push model.

4.2. Design alternative 2: Push model

Following this approach, end users are able to present their authorization credentials. Once again, these authorization credentials will be expressed using SAML attribute statements containing the roles played by the user. Similar approaches are also explained in Akenti Distributed Access Control (2004) and Beatty et al. (2003).

According to the standard, the SAML statements should have a short lifetime since they were designed to be used in SSO (Cantor, 2003) environments or authorization scenarios

based on Web, where users establish short-term sessions with the resources being accessed. The absence of revocation mechanisms for SAML statements, and its recommended usage for short-term sessions, suggests that the SAML documents should not be cached in intermediate entities, like a certificate repository. We addressed this issue providing several alternatives to obtain fresh SAML attributes statements that can be considered as trustworthy. As we will see, end users are able to obtain their attributes not only from their home domain but also when they are connected to a restricted foreign network.

As Fig. 4 shows, once the user has collected all the information he needs (this acquisition process will be explained later) he initiates the 802.1X authentication process with the foreign AAA server. In this case, we are going to use the PEAP (Protected EAP) protocol, which defines how to establish a TLS channel that can be used to authenticate the communicating parties and to protect further messages related to the authorization process. For example, the SAML statement generated by the user will be protected using PEAP.

During the authorization step, the end user has to select which attribute(s) is going to present as evidence, and, optionally, which kind of network service that he wishes to obtain (for example, based on NAS application attributes). Thus, the client software has to generate an *SAMLRequest* message including an *SAMLAuthorizationDecisionQuery* and the following data elements:

- *Subject* requesting the access.
- *Resource* identifier (network to be accessed).
- *Action* (NAS application attributes).
- *Attributes* to be used as evidence.

This message will be sent using PEAP to the AAA, and then forwarded to the PDP. Once the PDP obtains the request, the rest of the process follows the same rules as those that we explained for the pull alternative.

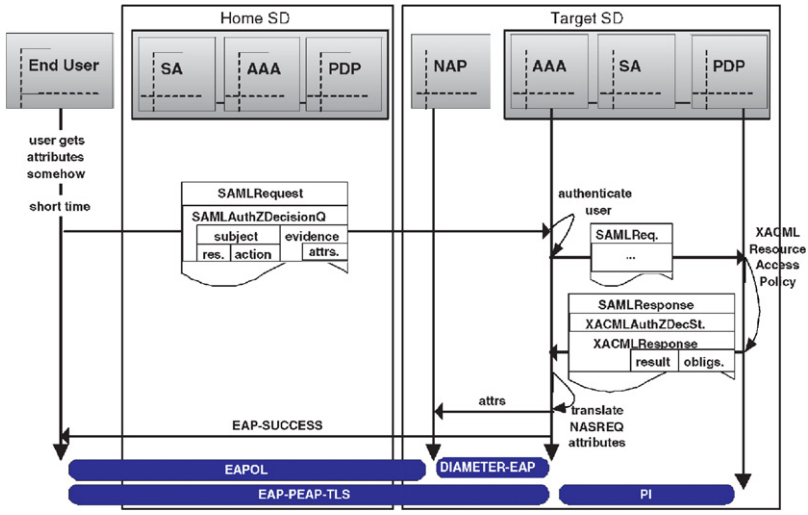

Fig. 4. Inter-domain push model.

On the other hand, in relation to the recovery of SAML attributes, we imposed two main requirements: first, end users should be able to retrieve their attributes using the same mechanism both if they are in the source domain or in the foreign one, that is, it should be opaque for the user; second, an intermediate web server should be used in order to provide an intuitive interface to access to the AAA server (i.e. to its SA application-specific module) since users are not intended to support DIAMETER.

As Fig. 5 shows, when the user is connected to his home domain, after the authentication step, the web server creates an *SAMLAttributeQuery* containing the identity of the end user whose attributes will be obtained. This query is sent using DIAMETER-SAML to the AAA server, and according to the role assignment policy, a list of *SAMLAttributeStatements* is returned to the web server. In this way, the user obtains the list of roles he can play. It is worth noting that these attribute statements are short-term credentials, and must be used next, for example, in a scenario similar to the one shown in Fig. 4.

Once the user is in the foreign domain, attributes are obtained using the foreign web server, which interacts with the foreign AAA server. As we can see in Fig. 6, the foreign AAA server makes use of DIAMETER-SAML, as shown in Section 4.1, to obtain the *SAMLAttributes* from the home server.

It is worth noting that, in case the information contained in attributes is considered as sensitive, the *SAMLResponses* provided by the home AAA server can be protected using the XML Encryption Standard (Eastlake, 2002).

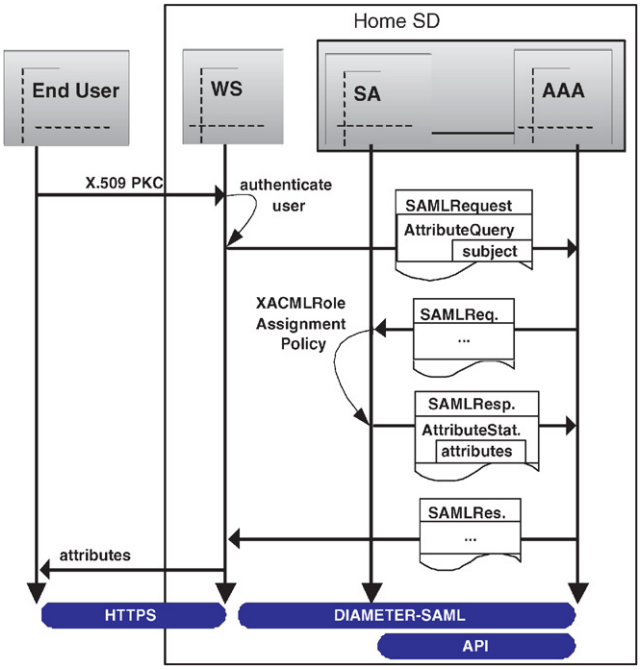

Fig. 5. Home domain recovery.

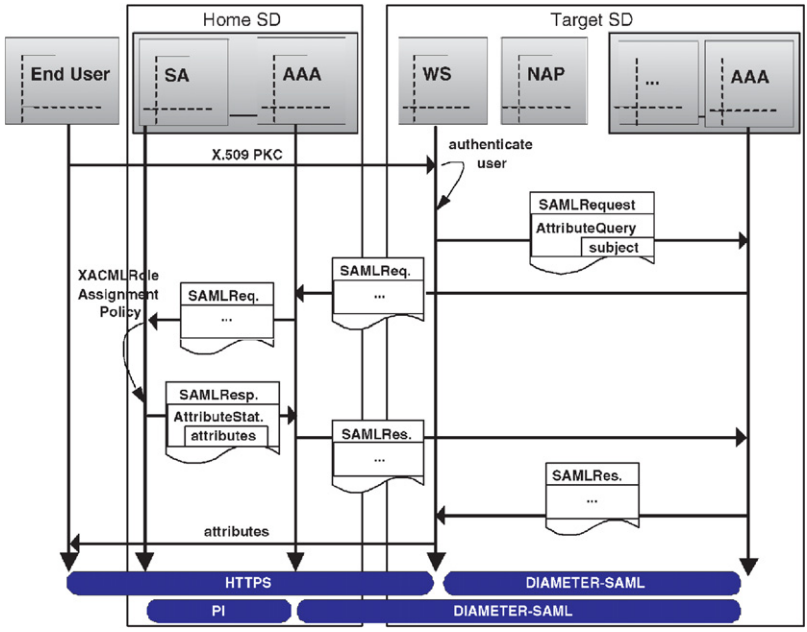

Fig. 6. Foreign domain recovery.

The main advantage of this alternative is that it provides a complete visibility and control of the authorization process to the end user, since he can select the type of connection, security properties, quality of service, etc. Moreover, he can provide personal information by means of references to some of his attributes. On the other hand, the software used by the client (usually referenced as a supplicant) must be modified in order to deal with SAML statements, as we can find in other existing proposals (Nikander, 2002).

5. Adaptation to the described scenarios

Following the requirements imposed by Section 2, and using the architecture and design alternatives presented in Sections 3 and 4, this section describes how our SAML-based network access service can be integrated with the different network access scenarios previously described.

5.1. Intra-domain campus scenario

As we previously mentioned, relationships between users and resources in a campus environment are, in most cases, stable and lasting, that is, a student or administrative member does not change his status frequently. Since all the entities involved (users and resources) pertain to the same domain, it is not a requirement that the user must hold his authorization credentials to request the network access. In this case, a pull model is preferred, enabling the user to avoid becoming involved in the authorization process.

In a campus scenario, authentication constitutes a usual requirement. In this case, the use of PKIs is recommended to manage the identity certificates related to the end users. Authorization mechanisms like SAML can be easily integrated in this environment, since the authorization management is carried out in a transparent way toward the users. If the first option is selected, a specific policy language, such as XACML, can be used to govern the decision.

5.2. Intra-domain airport scenario

When users and resources have a sporadic relationship, like in this kind of a scenario, there is no previous knowledge about the user identity and his attributes. The use of a vending machine, or a similar method, to obtain user attributes and the need to provide a mechanism to the end user to select the kind of connection suggests that a push model must be defined in this type of scenarios.

Since user authentication is not required, the use of an X.509 PKI is not needed, and the use of X.509 ACs as an authorization mechanism is such that ACs cannot be linked with public key certificates. In fact, SAML as an authorization mechanism is especially justified in this scenario.

5.3. Inter-domain campus scenario

When two or more campuses establish a trust relationship, based on the definition of an SLA, this relationship is usually very stable. In fact, the agreement has to specify the Authorization Authorities involved, the roles defined in each domain, and how these must be managed in the external domain. If all this information has been previously exchanged, it is not usually required that the users hold their attribute credentials, and a pull model is preferred, in a manner similar to other environments like Cantor (2005).

In this scenario, authentication in the foreign domain might not be required, depending on the user privacy requirements established in the SLA. If required, the use of a PKI infrastructure, using delegated authentication, is recommended.

The authorization mechanism used can be based on the SAML or X.509 AC standard, but in the absence of a PKI, the use of SAML is recommended, since ACs cannot be linked with public key certificates. The exchange of authorization data between domains, using the AAA servers, requires the definition of a new DIAMETER application, to transport SAML sentences as we showed in Section 2.

It must also be taken into account as to whether the different domains have previously established authorization mechanisms. If both domains make use of the same solution, the integration can be straightforward. On the other hand, if one domain is based on SAML and the other is based on X.509 AC, conversion mechanisms, like the one presented in Cánovas et al. (2004), should be used.

5.4. Inter-domain enterprise scenario

As mentioned in Section 2, inter-enterprise scenarios are not usually based on long and stable relationships, as it occurs in inter-campus scenarios. In this scenario, both pull and push models can be used, depending on the agreement between the organizations, but the push model is preferred, since the attributes are issued to be used in a specific moment, and

they have a lifetime shorter than attributes used in other scenarios. These can be held by the users and presented to the NAP in the foreign organization.

SAML is very suitable in this scenario, since the SAML sentences are supposed to have a very short lifetime. We also need an authorization language able to combine different authorization policies in order to reflect the existing collaborations among different organizations. XACML was indeed designed to support this kind of combining algorithms.

6. Related work

This section describes some works related to the resource access control and authorization mechanisms. Each of them covers a part of the proposed scenario: network access infrastructures, attribute-based authorization mechanisms and authorization architecture in multi-domain environments. However, any one of them constitutes a complete solution for the whole architecture proposed in this paper.

Some proposals, like Nikander (2002) and GreenPass (Goffee et al., 2004) give a solution for the deployment of network access control protocols like 802.1X, EAP or EAPOL, and how these can be integrated in an authorization scenario. Nikander (2002) defines a wireless network access architecture based on 802.1X and EAP, which can be used in an intra or inter-domain scenario through the use of a RADIUS-based architecture. However, it does not define how the high-level authentication and authentication processes are performed. GreenPass describes another scenario based on 802.1X and EAP, using the EAP-TLS protocol for authentication purposes. Moreover, this paper describes how a high-level authorization framework, based on SPKI/SDSI, can be integrated. However, the proposed scenario does not provide support for inter-domain scenarios.

Other interesting works are mainly related to the design of authorization decision makers, like PERMIS or Akenti. PERMIS provides an authorization system for distributed environments based on X.509 attribute certificates. It specifies the format of the authorization policy, expressed in XML, which includes information about recognized SOAs, subject domains, user roles (and their hierarchy), targets, actions and permissions. Akenti offers a similar proposal for distributed environments, but it does not make use of the X.509 attribute certificates since its certificates are instead encoded using XML. Both, and PERMIS mainly, as described before, can be easily integrated into a solution based on X.509 AC, but they only cover a piece of the whole infrastructure whilst the rest of the components are not defined.

Web services constitute the key scenario where authorization and authentication services are gaining more and more acceptance. One of the main reasons for this is the emergence of SAML as a specification language for authorization credentials, and its use in solutions like the Shibboleth and Liberty Alliance projects. However, these projects are not the most appropriate solutions to the proposed scenario, since they are Web-oriented and a deep transformation would be necessary to be adapted to a low-level network access control environment.

7. Conclusions

This paper addresses one of the authorization scenarios which is receiving more and more attention from several international initiatives. Network access control is paramount

in inter-domain environments. One of the main challenges derived from this type of heterogeneous and distributed scenarios is the complexity of the authentication and authorization mechanisms. As we showed, it is necessary to integrate several different technologies in an effective way. For this reason, it is especially important to provide versatile solutions to these mechanisms. We believe that the proposal presented here provides a wide range of real possibilities in order to accomplish the task of designing an access control system. Moreover, our proposal makes use of widely accepted standards that make it feasible.

Several conclusions can be obtained with respect to the technologies involved. First, although the proposed scenario is based on 802.1X it can be easily adapted for use with other technologies, like PANA. Second, protocols like EAP and DIAMETER can be extended to transport SAML-based statements. Finally, the use of XML-based solutions, SAML and XACML, for network access control purposes opens up an interesting research field that could be extended to other application scenarios.

With respect to the design alternatives, this paper presents two different RBAC solutions, which can be individually selected in order to implement the access control service that is best suited for a particular environment. Authorization can be performed in a transparent way, from the user's point of view, using the pull model. The push model slightly overloads the system in relation to the previous model, but it provides more options to the end users. One additional advantage of the selected design for the push model is that we did not introduce new protocols in order to send the information from the user to the AAA server. The proposal is based on previously defined protocols, such as PEAP and SAML.

In the related work section we can see that solutions like the ones described above, oriented to web environments, are not suitable for the proposed scenario, and new solutions, like the one presented, must therefore be defined. As a statement of direction, we are considering how to integrate other mechanisms which will provide accounting and anonymity services.

References

Aboba B, Simon D. PPP EAP-TLS authentication protocol. Internet Engineering Task Force, October 1999. Request for Comments (RFC) 2716.

Adoba B, Beadles M. The network access identifier, January 1999. Request for Comments (RFC) 2486.

Akenti Distributed Access Control, July 2004. http://www.itg.lbl.gov/Akenti.

Anderson A, Lockhart H. SAML 2.0 profile of XACML, 2004. OASIS Draft.

Anderson H, Josefson S, Zorn G, Simon D, Palekar A. Protected EAP protocol (PEAP), 2004. IETF Draft.

Beatty J, et al. Liberty protocols and schema specification version 1.1. Liberty Alliance Project, January 2003. http://www.projectliberty.org/.

Blake S, Black D, Carlson M, Davies E, Wang Z, Weiss W. An architecture for differentiated services, December 1998. Request for Comments (RFC) 2474.

Blunk L, Vollbrecht J. Extensible authentication protocol (EAP). Internet Engineering Task Force, March 1998. Request for Comments (RFC) 2284.

Calhoun P, Loughney J, Guttman E, Zorn G, Arkko J. Diameter base protocol. Internet Engineering Task Force, September 2003. Request for Comments (RFC) 3588.

Calhoun PR, Zorn G, Spence D, Mitton D. Diameter network access server application, 2004. IETF Draft.

Cánovas O, López G, Gómez AF. A credential conversion service for SAML-based scenarios. In: Proceedings of first European PKI workshop, June 2004. p. 297–305.

Cantor S. OASIS security assertion markup language (SAML): SSO use cases and scenarios, January 2003. OASIS Draft.

Cantor S. Shibboleth architecture: protocols and profiles, February 2005. http://shibboleth.internet2.edu/docs/draft-mace-shibboleth-arch-protocols-06.pdf.

Chadwick DW, Otenko A, Ball E. Role-based access control with x.509 attribute certificates. IEEE Internet Comput 2003;7(2):62–9.

Community Authorization Service, July 2004. http://www.globus.org/security/CAS/GT3/.

de Laat C, Gross G, Gommans L, Vollbrecht J, Spence D. Generic AAA architecture. Internet Engineering Task Force, August 2000. Request for Comments (RFC) 2903.

Eastlake D. XML-signature syntax and processing. World Wide Web Consortium, February 2002. http://www.w3.org/TR/xmldsig-core/.

Ellison C, Frantz B, Lampson B, Rivest R, Thomas B, Ylonen T. SPKI certificate theory, September 1999. Request for Comments (RFC) 2693.

Eronen P, Hiller T, Zorn G. Diameter extensible authentication protocol (EAP) application, 2005. IETF Draft.

Farrel S, Housley R. An Internet attribute certificate profile for authorization. Internet Engineering Task Force, April 2002. Request for Comments (RFC) 3281.

Ferraiolo D, Sandhu R, Gavrila S, Kuhn DR, Chandramouli R. Proposed nist standard for role-based access control. ACM Trans Inf System Security 2001;4(3).

Godik S, Moses T. OASIS eXtensible access control markup language (XACML) version 2.0, February 2005. OASIS Standard.

Goffee N, Kim S, Smith SW, Taylor P, Zhao M, Marchesini J. Greenpass: decentralized, pki-based authorization for wireless lans. In: Third annual PKI research and development workshop. NIST, April 2004.

Housley R, Polk W, Ford W, Solo D. Internet public key infrastructure, Part I: X.509 certificate and CRL profile, April 2002. Request for Comments (RFC) 3280.

Jayarama P, López R, Ohba Y, Parthasarathy M, Yegin A. PANA framework, 2005. IETF Draft.

LAN MAN Standards Committee of the IEEE Computer Society. IEEE Draft P802.1X/D11: Standard for Port based Network Access Control, March 2001.

Maler E, Mishra P, Philpott R. Assertions and protocol for the OASIS Security Assertion Markup Language (SAML) V1.1, September 2003. OASIS Standard.

Nikander P. Authorization and charging in public wlans using freebsd and 802.1x. In: Proceedings of the Freenix track: 2002 USENIX annual technical conference, June 2002.

Rigney C, Rubens A, Simpson W, Willens S. Remote authentication dial in user service (RADIUS). Internet Research Task Force, April 1997. Request for Comments (RFC) 2138.