Making the Case for EAP Channel Bindings

T. Charles Clancy
University of Maryland
tcc@umd.edu

Katrin Hoeper[*]
Motorola
khoeper@motorola.com

***Abstract*— In current networks that use EAP and AAA for authenticated admission control, such as WiFi, WiMAX, and various 3G internetworking protocols, a malicious base station can advertise false information to prospective users in an effort to manipulate network access in some way. This paper identifies and discusses the resulting threats (e.g. the lying NAS problem in enterprise networks and the newly identified lying provider problem in roaming environments) and shows how these threats can be exploited for a number of attacks, including traffic herding, denial of service, cryptographic downgrade attacks, and forced roaming. Finally, the paper presents how an EAP channel binding protocol can thwart the identified attacks by allowing a client to inform the EAP server about the unauthenticated information it received during the network selection process. The back-end server can then ensure the consistency of the advertised information with its configured policy. As a result, EAP channel bindings enable an end-to-end validation of network properties, which is otherwise infeasible in existing AAA infrastructures. Standardization activities currently exist within the IETF to implement this technique.**

I. Introduction

The Extensible Authentication Protocol (EAP) was originally standardized in the 1990s as a way to perform password-based authentication over a dialup connection. Over the past 15 years, however, its popularity has grown as a generic protocol for authenticating network access, and it is currently a favorite of many existing and emerging wireless networking protocols, such as IEEE 802.11 WiFi, IEEE 802.16 WiMAX, and a number of 3G/4G heterogeneous internetworking systems.

In order to support authentication in large, distributed networks, EAP typically relies on the Authentication, Authorization, and Accounting (AAA) suite of protocols to interconnect base stations and authentication servers. The two major AAA protocols are the Remote Authenticated Dial-In User Service (RADIUS) [1] and Diameter [2].

Security associations and trust relationships in AAA infrastructures are typically nebulous. In order to enable scalability, AAA proxies are often used, which breaks ones ability to provide end-to-end security. As a result, an authentication server may not be able to trust information it receives in otherwise encrypted channels.

In this paper, we explore the interplay between EAP and AAA in these complex infrastructures, and present a class of attacks against EAP possible in existing deployments. We then describe EAP Channel Bindings, where protected channels within the EAP conversation can be used to provide end-to-end validation of network properties, which is otherwise impossible in existing AAA infrastructures.

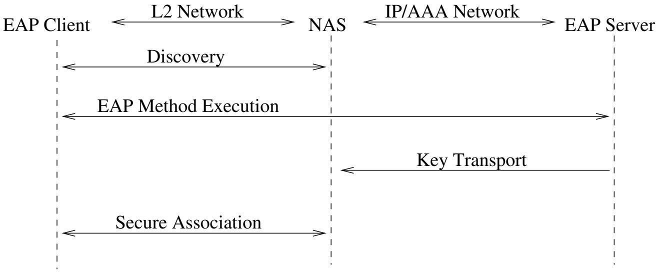


Fig. 1. EAP and AAA architecture, showing the steps executed during an EAP authentication

The remainder of this paper is organized as follows. Section 2 describes the existing EAP and AAA architectures in further detail. Section 3 outlines attacks to these architectures. Section 4 describes EAP Channel Bindings, the proposed solution to the problem. Section 5 concludes.

II. EAP and AAA Architectures

This section describes how EAP and AAA function from a basic standards perspective, and then outlines how they are deployed in both enterprise and service-provider networks.

A. Passthrough EAP Authentication

EAP, as its name implies, is an extensible framework supporting many different cryptographic authentication protocols called *methods*. These methods support a vast array of credential types, including passwords, secret keys, certificates, and SIM cards. In this paper we won't focus on the methods specifically, except insomuch as they provide mutual authentication and derive keying material known to the client and server.

EAP was designed to be a protocol executed directly between the EAP client and EAP server. The EAP server was originally collocated with the Network Access Server (NAS), which is the network edge device responsible for enforcing admission control. However, in an effort to provide centralized authentication and accounting services, the architecture was split such that EAP server could be physically separated from the NAS, and one EAP server could service many NASes within a single domain. AAA was used to provide the interconnect between the NAS and EAP server. This is

[*] The research for this work was conducted while Katrin Hoeper was with the Computer Security Division at the National Institute of Research and Technology (NIST), Gaithersburg, Maryland.

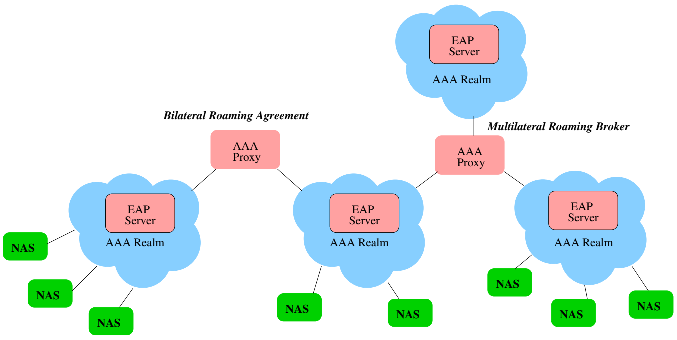


Fig. 2. Example AAA proxy architecture, showing multiple autonomous realms interconnected with proxy servers to support roaming agreements

known as using EAP in *passthrough* mode, and is depicted in Figure 1.

EAP itself makes no security requirements of the AAA infrastructure. Modern EAP methods are resistant to cryptographic attacks. EAP simply requires that the AAA infrastructure properly route its messages. For key transport, however, it is assumed that the transported key will be encrypted to provide confidentiality. In RADIUS this is typically accomplished using a password-based RADIUS shared secret, and in Diameter, either shared-key or public-key approaches can be used to secure the links. Section II-C describes intricacies of large AAA infrastructures in further detail, especially when AAA proxies are involved.

B. *Enterprise Usage Model*

The enterprise usage model focuses on deployments of EAP and AAA within a single administrative domain, such as a college campus or office building. The basic assumption is that the same entity manages all the NASes in the network, and controls the AAA service. In this case, the AAA server is in a position to know about all the NASes in the network, and has unique security associations established with each one. Thus, when it receives a AAA message, it can definitively know the source of that message.

C. *Service Provider Usage Model*

The service provider usage model is significantly more complex than the enterprise case, and is exemplified by WiFi hotspot vendors or cellular service provider networks. Here we often find bilateral roaming agreements and/or brokers that serve as a clearinghouse for smaller operators.

The ability to scalably provide roaming is provided by AAA proxies. A proxy is a device that can implement both the AAA client and AAA server roles. It sits as an intermediary between devices and aggregates security associations. Figure 2 shows a number of interconnected AAA realms.

Proxies allow service providers to significantly decrease the number of security associations in a network. If N operators each have M NASes, a mesh of $\mathcal{O}(N^2M)$ total security associations would be required to provide end-to-end security between all servers and NASes. With proxies, only $\mathcal{O}(M)$ NAS security associations are required within each realm, and roaming can be handled at the operator level, rather than individual NAS level, involving up to $\mathcal{O}(N^2)$ security associations if all operators have bilateral roaming agreements, and only $\mathcal{O}(N)$ if a multilateral roaming broker is used.

To accomplish this, proxies act as a server and decrypt inbound AAA traffic, and then act as a client, re-encrypting it and sending it back out. This gives the proxy tremendous power: it can alter, observe, or fabricate traffic from any NAS to any server.

There are some provisions in Diameter that use public-key certificates to provide end-to-end security through proxies, but these are rarely deployed, and are not used in any major AAA roaming network today. Consequently, protecting proxy servers from compromise is extremely important in ensuring the integrity of a AAA roaming infrastructure.

III. Existing Threats to the EAP Architecture

In the previous section, we described two different use cases for EAP: the enterprise authentication case, and the service provider authentication case. Each of these two cases suffers from NASes providing misinformation to possible clients, which can seriously impact the overall network security. In this section we detail these attacks and their impact.

A. *Lying NAS Threat*

The lying NAS threat is well documented problem [3]. It stems from the fact that a NAS may advertise any information it wants to the client before a security association is formed, and there is no way to verify it's veracity.

For example, imagine Figure 1 was an IEEE 802.11 authentication [4] using WPA2. During the *Discovery* phase, the NAS sends beacon messages which are received by the EAP client. These messages include information such as the name of the network (SSID), supported communications rates, and the allowable modes of authentication and encryption (RSN-IE). The wireless subscriber makes a decision about whether or not to attempt authentication with that access point based on the advertised information. However, this broadcast information is unauthenticated and can be easily spoofed.

Additionally, the NAS could provide false AAA attributes to the EAP server during the client authentication in an effort to affect the authorization decision. For example, it could say the client was connecting to a different underlying technology (i.e. WiMAX instead of WiFi). If the EAP server has a unique security association with every NAS, it is possible to validate the source of the information, and therefore determine its consistency with system policy, but current server software does not support this.

The ability of the NAS to lie to both the EAP client and EAP server makes a number of attacks possible. By manipulating beacon information, a NAS can influence clients' decision about access point selection. By luring clients of interest through manipulation of advertised capabilities, data rates, etc, it can more easily eavesdrop on their sessions (given it knows their session keys).

A more serious attack exists in networks that support multiple subnetworks, e.g. a corporate network allowing wireless access to both the Internet and confidential intranet via

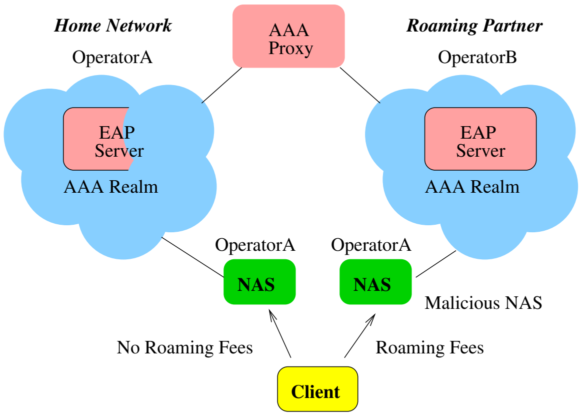


Fig. 3. AAA Roaming relationship where a NAS on OperatorB is masquerading as OperatorA in an effort to lure clients for the purpose of charging them roaming fees

authentication to the same EAP server. Here a malicious NAS connected to the less-secure network can masquerade as a NAS for the more-secure network, and lure unsuspecting clients to connect to a network with a false sense of security regarding their traffic.

Finally, another example attack is a cryptographic downgrade attack. In a WiFi network, after deriving session keys, WEP, TKIP, and AES are all allowed ciphers for encrypting traffic, with AES being the current best practice, and WEP being completely exploitable. A NAS may be configured to only advertise AES to clients, but a malicious NAS could change the advertisements to only include WEP, and force clients to use a weak cipher. This could enable other adversarial forces in the network to break the client's session keys and compromise its session.

B. Lying Provider Threat

The lying provider is a relatively new concept in EAP, and is primarily relevant to the service provider use case. In the service provider model, the authenticating EAP server does not have a unique security association with all the NASes in its partner networks. In addition, traffic from those NASes comes via AAA proxies that decrypt, reframe, and retransmit the underlying AAA messages. Hence, NAS-specific information is either not relevant or lost en route.

Since the EAP server cannot distinguish individual NASes, an attack from a single NAS becomes an attack from any NAS, and consequently an attack from the roaming partner itself, hence the designation as the "lying provider threat". The EAP server can only verify the information received from the last hop and, thus, it does not even know that it was a malicious NAS providing false information – it could be a AAA proxy somewhere within the network.

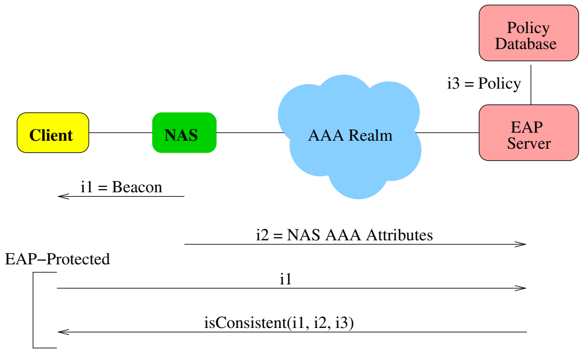


Fig. 4. Channel Bindings architecture where information provided by the NAS is validated by the server during an EAP Method execution

In the service provider model, most malicious attacks revolve around money. Where multiple providers who share roaming agreements are operating in the same geographic vicinity, their goal is to attract customers and maximize profits from roaming fees. One of the best ways to do this is by masquerading as a user's home network. Figure 3 diagrams such an attack where an operator can provide roaming service to clients who have no idea they are being charged roaming fees.

IV. EAP Channel Bindings

In an effort to address the discussed shortcomings in the current EAP architecture, use of EAP Channel Bindings [5] has been proposed to provide a mechanism for verifying the consistency of information asserted by the NAS to the EAP client and server with network policy.

A. Architecture

The goal of channel bindings is to *bind* session-specific metadata about the *channel* to the authentication session. This ensures that both the client and server have the same understanding about the network, and that this understanding is consistent with network policy. There are two basic ways that have been proposed to achieve this: by mixing this information into the session key derivation [6] or by exchanging it in an integrity-protected channel during the authentication protocol execution [5].

There are numerous reasons why the latter approach is more robust in current EAP deployments. If information is mixed directly into keying material, acceptable minor variations in this data between the client and server would result in keying failure and prevent session initiation. Since most modern EAP methods support the exchange of arbitrary data within an integrity-protected channel, the latter approach allows for policy-based comparison of information, rather than bitwise equality. It also allows for a gradual roll-out of channel bindings in large networks through parameter enrollment, and increases the ability to debug authentication failures.

Implementing channel bindings simply requires the client be able to securely send information advertised to it during the network selection process to the server for validation. The server must then be able to securely respond with the results of its consistency check. Figure 4 shows the entire process.

During the network selection phase, information i_1 is advertised in broadcast beacons. The client uses this information to make a decision about which NAS to associate with. During the authentication phase, information i_2 is transmitted to the EAP server as a part of the AAA protocol, and contains some information about the client and the circumstances of the connection. Then during the EAP method execution, the client can securely transmit i_1 to the EAP server. With access to i_1 and i_2, and after querying a policy database for network policy i_3, the server can make an authorization decision about whether or not this client should be receiving service from this NAS. The result of this consistency check and authorization decision can then be securely returned to the client.

The policy database allows for a large number of different validation and authorization options. The server can now control which networks the client may connect to at which times, verify the consistency of information advertised by roaming partners with the roaming agreement, detect attempts to use weaker security mechanisms not authorized by network policy, and otherwise detect lying NASes and lying providers.

It should be noted that AAA proxies provide the major constraint in the ability to validate received information i_2. Proxied entities should be treated as indistinguishable from the EAP server's perspective, since it has no unique security associations with them to securely receive parameters. Thus, channel bindings are only useful for validating "last-hop" information. For example, if a roaming partner provides both WiFi and WiMAX access, and an EAP client says it's connecting to a WiMAX network, the EAP server would have no way to verify whether the NAS the client's connecting to is supposed to provide WiMAX service, or only WiFi. It can only verify that per its roaming agreement with the partner network, both WiFi and WiMAX are permissible access technologies and that the connection is therefore permissible.

B. Protocol

To implement this architecture, a protocol is needed for communicating between the EAP client and server during an authentication session. Most modern EAP methods support communicating information within the EAP method after session keys have been established, so implementing the channel bindings protocol simply requires definition of a payload format and allocation of type codes for these methods [7].

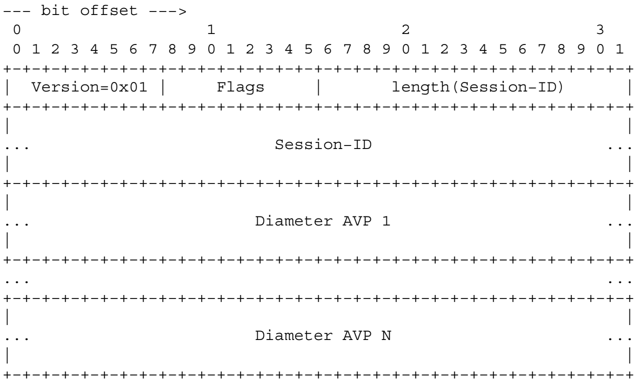
Another possible approach would be to implement the transport within a single wrapper method [8], which tunnels another EAP method execution within it. This would minimize code development requirements, but decrease the ability for EAP implementations to negotiate which method to use for authentication.

The proposed approach, in order to maximize consistency of AAA attributes advertised by the NAS to the server with information received from the client, is to have the client format attributes of interest as Diameter AVPs [7]. Once EAP methods have the ability to carry arbitrary AVPs, they can then easily carry the channel binding information.

C. Lower-Layer Bindings

Each lower layer will need to decide which fields advertised during the initial network discovery phase are most important for ensuring system security and, thus, should be included in i_1. In addition, information included in i_1 needs also to be useful and verifiable by the EAP server. The following AVPs seem to meet these requirements for making authorization decisions:

- **NAS-Port-Type:** Indicates the underlying link-layer technology (e.g. WiFi, WiMAX, PPP, etc)
- **Called-Station-Id:** Contains network name and NAS MAC address information
- **Calling-Station-Id:** MAC address of the EAP client
- **Mobility-Domain-Id:** Contains domain information in mobile networks, used with IEEE 802.11r
- **Mesh-Key-Distributor-Domain-Id:** Contains key domain information required for key derivation, used with IEEE 802.11s
- **User-Name:** Username of the authenticating EAP user
- **Cost-Information:** Part of the Diameter Credit-Control application [9] and used to indicate billing information



```
--- bit offset --->
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Version=0x01 |     Flags     |       length(Session-ID)      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
...                          Session-ID                       ...
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
...                      Diameter AVP 1                       ...
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
...                                                           ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
...                      Diameter AVP N                       ...
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Fig. 5. Protocol payload formatting for exchanging Diameter AVPs within EAP methods as defined in [7]

D. AAA-Layer Bindings

When deploying channel bindings it needs to be decided which AAA attributes in RADIUS Accept-Request messages can and should be validated by a AAA server, i.e. which data should be included in information i_2. As noted before, this data can be manipulated by AAA proxies. However as thorough of a validation as possible should be conducted in an effort to detect possible attacks. For example, if the IP address of a particular NAS is unknown, the server could still check whether the subnet of the IP address was correct. Or, if an attribute cannot be directly verified by a server, this attribute could be linked to another verifiable one and be stored as a value pair in the policy database (i_3).

The following AAA attributes that are typically transmitted from the NAS to the EAP server during an EAP authentication seem suitable for such a check as part of channel bindings:

- **User-Name:** Username of the authenticating EAP user
- **NAS-IP-Address:** IP address of the NAS
- **Calling-Station-Id:** MAC address of the EAP client
- **NAS-Identifier:** Identifier populated by the NAS
- **NAS-Port-Type:** Indicates the underlying link-layer technology (e.g. WiFi, WiMAX, PPP, etc)

E. Deployment Considerations

Deployment of channel bindings is non-trivial. First, software must be modified to support carrying channel binding information within EAP methods. This involves implementation of the protocol within existing methods, and modifying hardware drivers on EAP clients to support passing lower-layer information into the EAP stack. For example, an EAP implementation may need access to the original 802.11 beacons for the access point to which the client is connecting, and currently that information is not normally passed up the network stack and into the operating system.

After the necessary software tools are available, operators will need to deploy them. The key to ensuring security in this approach is use of the policy database. Without that, the EAP server has no known-good standard against which to compare information it receives from clients and NASes. Manually entering this information can be a tedious and time-consuming task, especially in an enterprise system where per-NAS entries would be required.

To facilitate roll-out, an automatic policy creation based on existing network access behavior is recommended. The EAP server can simply save the received i_1 and i_2 it receives into the policy database, with the assumption the network is currently not under attack, and NASes are not behaving in a malicious manner. Furthermore, the presented channel binding validation can be implemented incrementally.

V. Conclusion

This paper describes current threats to the EAP-based authentication systems that utilize a AAA infrastructure. These attacks involve the lying NAS threat in enterprise scenarios and the lying provider threat in service provider scenarios. To address these attacks, use of a channel bindings protocol is described that allows a client to inform the EAP server about unauthenticated information it received during the network selection process, and the EAP server can then verify that this information is consistent with the network policy. This prevents numerous attacks currently possible against EAP networks, including traffic herding, denial of service, cryptographic downgrade attacks, and forced roaming.

References

[1] C. Rigney, S. Willens, A. Rubens, and W. Simpson, "Remote authentication dial in user service (RADIUS)," June 2000.

[2] P. Calhoun, J. Loughney, E. Guttman, G. Zorn, and J. Arkko, "Diameter base protocol," IETF Proposed Standard, RFC 3588, September 2003.

[3] B. Aboba, L. Blunk, J. Vollbrecht, J. Carlson, and H. Levkowetz, "Extensible authentication protocol," IETF Proposed Standard, RFC 3748, June 2004.

[4] IEEE 802.11-2007, "Wireless LAN medium access control and physical layer specification," March 2007.

[5] T. Clancy and K. Hoeper, "Channel binding support for eap methods," IETF Internet Draft, draft-clancy-emu-chbind, November 2008.

[6] Y. Ohba, M. Parthasarathy, and M. Yanagiya, "Channel binding mechanism based on parameter binding in key derivation," IETF Expired Internet Draft, draft-ohba-eap-channel-binding-02, December 2006.

[7] T. Clancy, "Eap method support for transporting aaa payloads," IETF Internet Draft, draft-ietf-emu-aaapay, July 2008.

[8] Y. Ohba, S. Das, and R. Lopez, "An eap method for eap extension (eap-ext)," IETF Expired Internet Draft, draft-ohba-hokey-emu-eap-ext-02, July 2007.

[9] H. Hakala, L. Mattila, J. Koskinen, M. Stura, and J. Loughney, "Diameter credit-control application," IETF Proposed Standard, RFC 4006, August 2005.