EMU Working Group   S. Hartman, Ed.
Internet-Draft   Painless Security
Intended status: Standards Track   T. Clancy
Expires: August 13, 2011   LTS
                                                           K. Hoeper
                                                       Motorola, Inc.
                                                    February 9, 2011

Channel Binding Support for EAP Methods
draft-ietf-emu-chbind-07.txt

Abstract

   This document defines how to implement channel bindings for
   Extensible Authentication Protocol (EAP) methods to address the lying
   NAS as well as the lying provider problem.

Status of this Memo

   This Internet-Draft is submitted in full conformance with the
   provisions of BCP 78 and BCP 79.

   Internet-Drafts are working documents of the Internet Engineering
   Task Force (IETF). Note that other groups may also distribute
   working documents as Internet-Drafts. The list of current Internet-
   Drafts is at http://datatracker.ietf.org/drafts/current/.

   Internet-Drafts are draft documents valid for a maximum of six months
   and may be updated, replaced, or obsoleted by other documents at any
   time. It is inappropriate to use Internet-Drafts as reference
   material or to cite them other than as "work in progress."

   This Internet-Draft will expire on August 13, 2011.

Table of Contents

1. Introduction

   The so-called "lying NAS" problem is a well-documented problem with
   the current Extensible Authentication Protocol (EAP) architecture
   [RFC3748] when used in pass-through authenticator mode. Here, a
   Network Access Server (NAS), or pass-through authenticator, may
   represent one set of information (e.g. network identity,
   capabilities, configuration, etc) to the backend Authentication,
   Authorization, and Accounting (AAA) infrastructure, while
   representing contrary information to EAP peers. Another possibility
   is that the same false information could be provided to both the EAP
   peer and EAP server by the NAS. A "lying" entity can also be located
   anywhere on the AAA path between the NAS and the EAP server.

   This problem results when the same credentials are used to access
   multiple services that differ in some interesting property. The EAP
   server learns which client credentials are in use. The client knows
   which EAP credentials are used, but cannot distinguish between
   servers that use those credentials.

   As a concrete example, consider an organization with two different
   IEEE 802.11 wireless networks. One is a relatively low-security
   network for reading e-mail while the other has access to valuable
   confidential information. An access point on the e-mail network
   could act as a lying NAS, sending the SSID of the confidential
   network in its beacons. This access point could gain an advantage by
   doing so if it tricks clients intending to connect to the
   confidential network to connect to it and disclose confidential
   information.

   A similar problem can be observed in the context of roaming. Here,
   the lying entity is located in a visited service provider network,
   e.g. attempting to lure peers to connect to the network based on
   false advertized roaming rates. This is referred to as "lying
   provider" problem in the remainder of this document. The lying
   entity's motivation often is financial; the entity may be paid
   whenever peers roam to its service. However a lying entity in a
   provider network can gain access to traffic that it might not
   otherwise see.

   This document defines and implements EAP channel bindings to solve
   the lying NAS and the lying provider problems, using a process in
   which the EAP peer provides information about the characteristics of
   the service provided by the authenticator to the AAA server protected
   within the EAP method. This allows the server to verify the
   authenticator is providing information to the peer that is consistent
   with the information received from this authenticator as well as the
   information stored about this authenticator. "AAA Payloads" defined

in [I-D.clancy-emu-aaapay] served as the starting point for the
mechanism proposed in this specification to carry this information..

2. Terminology

In this document, several words are used to signify the requirements
of the specification. These words are often capitalized. The key
words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD",
"SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document
are to be interpreted as described in [RFC2119].

3. Problem Statement

In a [RFC4017] compliant EAP authentication, the EAP peer and EAP
server mutually authenticate each other, and derive keying material.
However, when operating in pass-through mode, the EAP server can be
far removed from the authenticator. A malicious or compromised
authenticator may represent incorrect information about the network
to the peer in an effort to affect its operation in some way.
Additionally, while an authenticator may not be compromised, other
compromised elements in the network (such as proxies) could provide
false information to the authenticator that it could simply be
relaying to EAP peers. Hence, the goal must be to ensure that the
authenticator is providing correct information to the EAP peer during
the initial network discovery, selection, and authentication.

There are two different types of networks to consider: enterprise
networks and service provider networks. In enterprise networks,
assuming a single administrative domain, it is feasible for an EAP
server to have information about all the authenticators in the
network. In service provider networks, global knowledge is
infeasible due to indirection via roaming. When a peer is outside
its home administrative domain, the goal is to ensure that the level
of service received by the peer is consistent with the contractual
agreement between the two service providers. The same EAP server may
need to support both types of networks. For example an enterprise
may have a roaming agreement permitting its users to use the networks
of third-party service providers. In these situations, the EAP
server may authenticate for an enterprise and provider network.

The following are example attacks possible by presenting false
network information to peers.

o Enterprise Network: A corporate network may have multiple virtual
   Lads (VLANs) running throughout their campus network, and have
   IEEE 802.11 access points connected to each VLAN. Assume one VLAN

connects users to the firewalled corporate network, while the
other connects users to a public guest network. The corporate
network is assumed to be free of adversarial elements, while the
guest network is assumed to possibly have malicious elements.
Access Points on both VLANs are serviced by the same EAP server,
but broadcast different SSIDs to differentiate. A compromised
access point connected to the guest network but not the corporate
network could advertise the SSID of the corporate network in an
effort to lure peers to connect to a network with a false sense of
security regarding their traffic. Conditions and further details
of this attack can be found in the Appendix.

o Enterprise network: The EAP GSS-API mechanism
   [I-D.ietf-abfab-gss-eap] mechanism provides a way to use EAP to
   authenticate to mail servers, instant messaging servers and other
   non-network services. Without EAP channel binding, an attacker
   could trick the user into connecting to a relatively untrusted
   service instead of a relatively trusted service.

o Service Provider Network: An EAP-enabled mobile phone provider
   could advertize very competitive flat rates but send per minute
   rates to the home server, thus, luring peers to connect to their
   network and overcharging them. In more elaborate attacks, peers
   can be tricked into roaming without their knowledge. For example,
   a mobile phone provider operating along a geo-political boundary
   could boost their cell towers' transmission power and advertise
   the network identity of the neighboring country's indigenous
   provider. This would cause unknowing handsets to associate with
   an unintended operator, and consequently be subject to high
   roaming fees without realizing they had roamed off their home
   provider's network. These types of scenarios can be considered as
   "lying provider" problem, because here the provider configures its
   NAS to broadcast false information. For the purpose of channel
   bindings as defined in this draft, it does not matter which local
   entity (or entities) is "lying" in a service provider network
   (local NAS, local authentication server and/or local proxies),
   because the only information received from the visited network
   that is verified by channel bindings is the information the home
   authentication server received from the last hop in the
   communication chain. In other words, channel bindings enable the
   detection of inconsistencies in the information from a visited
   network, but cannot determine which entity is lying. Naturally,
   channel bindings for EAP methods can only verify the endpoints
   and, if desirable, intermediate hops need to be protected by the
   employed AAA protocol.

o Enterprise and provider networks: In a situation where an
   enterprise has roaming agreements with providers, a compromised
   access point in a provider network could masquerade as the

enterprise network in an attempt to gain confidential information.
Today this could potentially be solved by using different
credentials for internal and external access. Depending on the
type of credential this may introduce usability or man-in-the-
middle security issues.

To address these problems, a mechanism is required to validate
unauthenticated information advertised by EAP authenticators.

4. Channel Bindings

EAP channel bindings seek to authenticate previously unauthenticated
information provided by the authenticator to the EAP peer, by
allowing the peer and server to compare their perception of network
properties in a secure channel.

It should be noted that the definition of EAP channel bindings
differs somewhat from channel bindings documented in [RFC5056], which
seek to securely bind together the end points of a multi-layer
protocol, allowing lower layers to protect data from higher layers.
Unlike [RFC5056], EAP channel bindings do not ensure the binding of
different layers of a session but rather the information advertised
to EAP peer by an authenticator acting as pass-through device during
an EAP execution. The term channel bindings was independently
adopted by these two related concepts; by the time the conflict was
discovered, a wide body of literature existed for each usage. EAP
channel bindings could be used to provide RFC 5056 channel bindings.
In particular, an inner EAP method could be bound to an outer method
by including the RFC 5056 channel binding data for the outer channel
in the inner EAP method's channel bindings. Doing so would provide a
facility similar to EAP cryptographic binding, except that a man-in-
the-middle could not extract the inner method from the tunnel. This
specification does not weigh the advantages of doing so nor specify
how to do so; the example is provided only to illustrate how EAP
channel binding and RFC 5056 channel binding overlap.

4.1. Types of EAP Channel Bindings

There are two categories of approach to EAP channel bindings:

o After keys have been derived during an EAP execution, the peer and
   server can, in an integrity-protected channel, exchange plaintext
   information about the network with each other, and verify
   consistency and correctness.

o The peer and server can both uniquely encode their respective view
   of the network information without exchanging it, resulting into
   an opaque blob that can be included directly into the derivation
   of EAP session keys.

Both approaches are only applicable to key deriving EAP methods and
both have advantages and disadvantages. Various hybrid approaches
are also possible. Advantages of exchanging plaintext information
include:

o It allows for policy-based comparisons of network properties,
   rather than requiring precise matches for every field, which
   achieves a policy-defined consistency, rather than bitwise
   equality. This allows network operators to define which
   properties are important and even verifiable in their network.

o EAP methods that support extensible, integrity-protected channels
   can easily include support for exchanging this network
   information. In contrast, direct inclusion into the key
   derivation would require more extensive revisions to existing EAP
   methods or a wrapper EAP method.

o Given it doesn't affect the key derivation, this approach
   facilitates debugging, incremental deployment, backward
   compatibility and a logging mode in which verification results are
   recorded but do not have an effect on the remainder of the EAP
   execution. The exact use of the verification results can be
   subject to the network policy. Additionally, consistent
   information canonicalization and formatting for the key derivation
   approach would likely cause significant deployment problems.

The following are advantages of directly including channel binding
information in the key derivation:

o EAP methods not supporting extensible, integrity-protected
   channels could still be supported, either by revising their key
   derivation, revising EAP, or wrapping them in a universal method
   that supports channel binding.

o It can guarantee proper channel information, since subsequent
   communication would be impossible if differences in channel
   information yielded different session keys on the EAP peer and
   server.

4.2. Channel Bindings in the Secure Association Protocol

This document describes channel bindings performed by transporting
channel binding information as part of an integrity-protected

exchange within an EAP method. Alternatively, some future document
could specify a mechanism for transporting channel bindings within
the lower layer's secure association protocol. Such a specification
would need to describe how channel bindings are exchanged over the
lower layer protocol between the peer and authenticator. In
addition, since the EAP exchange concludes before the secure
association protocol begins, a mechanism for transporting the channel
bindings from the authenticator to the EAP server needs to be
specified. A mechanism for transporting a protected result from the
EAP server, through the authenticator, back to the peer needs to be
specified.

The channel bindings MUST be transported with integrity protection
based on a key known only to the peer and EAP server. The channel
bindings SHOULD be confidentiality protected using a key known only
to the peer and EAP server. For the system to function, the EAP
server or AAA server needs access to the channel binding information
from the peer as well as the AAA attributes and a local database
described later in this document.

The primary advantage of sending channel bindings as part of the
secure association protocol is that EAP methods need not be changed.
The disadvantage is that a new AAA exchange is required, and secure
association protocols need to be changed. As the result of the
secure association protocol change, every NAS needs to be upgraded to
support channel bindings within the secure association protocol.

For many deployments, changing all the NASes is expensive and adding
channel binding support to enough EAP methods to meet the goals of
the deployment will be cheaper. However for deployment of new
equipment, or especially deployment of a new lower layer technology,
changing the NASes may be cheaper than changing EAP methods.
Especially if such a deployment needed to support a large number of
EAP methods, sending channel bindings in the secure association
protocol might make sense.

If channel bindings using a secure association protocol is specified,
semantics as well as the set of information that peers exchange can
be shared with the mechanism described in this document.

4.3. Channel Bindings Scope

The scope of EAP channel bindings differs somewhat depending on the
type of deployment in which they are being used. In enterprise
networks, they can be used to authenticate very specific properties
of the authenticator (e.g. MAC address, supported link types and
data rates, etc), while in service provider networks they can
generally only authenticate broader information about a roaming

partner's network (e.g. network name, roaming information, link
security requirements, etc). The reason for the difference has to do
with the amount of information about the authenticator and/or network
to which the peer is connected the home EAP server is expected to
have access to. In roaming cases, the home server is likely to only
have access to information contained in their roaming agreements.

With any multi-hop AAA infrastructure, many of the NAS-specific AAA
attributes are obscured by the AAA proxy that's decrypting,
reframing, and retransmitting the underlying AAA messages.
Especially service provider networks are affected by this and the AAA
information received from the last hop may not contain much
verifiable information any longer. For example, information carried
in AAA attributes such as the NAS IP address may have been lost in
transition and are thus not known to the EAP server. This affects
the ability of the EAP server to verify specific NAS properties.
However, often verification of the MAC or IP address of the NAS is
not useful for improving the overall security posture of a network.
More often it is useful to make policy decisions about services being
offered to peers. For example, in an IEEE 802.11 network, the EAP
server may wish to ensure that peers connecting to the corporate
intranet are using secure link-layer encryption, while link-layer
security requirements for peers connecting to the guest network could
be less stringent. These types of policy decisions can be made
without knowing or being able to verify the IP address of the NAS
through which the peer is connecting.

The properties of the network that the peer wishes to validate depend
on the specific deployment. In a mobile phone network, peers
generally don't care what the name of the network is, as long as they
can make their phone call and are charged the expected amount for the
call. However, in an enterprise network the administrators of a peer
may be more concerned with specifics of where their network traffic
is being routed and what VLAN is in use. To establish policies
surrounding these requirements administrators would capture some
attribute such as SSID to describe the properties of the network they
care about. Channel bindings could validate the SSID. The
administrator would need to make sure that the network guarantees
that when an authenticator trusted by the AAA infrastructure to offer
a particular SSID to clients does offer this SSID, that network has
the intended properties. Generally it is not possible for channel
bindings to detect lying NAS behavior when the NAS is authorized to
claim a particular service. That is, if the same physical
authenticator is permitted to advertize two networks, the AAA
infrastructure is unlikely to be able to determine when this
authenticator lyes.

As discussed in the next section, some of the most important

information to verify cannot come from AAA attributes but instead
comes from local configuration. For example in the mobile phone
case, the expected roaming rate cannot come from the roaming provider
without being verified against the contract between the two
providers. Similarly, in an enterprise, the SSID a particular access
point is expected to advertize is a matter of configuration rather
than something that can be trusted because it is included in an AAA
exchange.

Channel bindings can be important for forming pockets of trust,
especially when provider networks are involved, and exact information
is not available to the EAP server. Without channel bindings, all
entities in the system need to be held to the standards of the most
trusted entity that could be accessed using the EAP credential.
Otherwise, a less trusted entity can impersonate a more trusted
entity. However when channel bindings are used, the EAP server can
use information supplied by the peer, AAA protocols and local
database to distinguish less trusted entities from more trusted
entities. One possible deployment involves being able to verify a
number of characteristics about relatively trusted entities while for
other entities simply verifying that they are less trusted.

Any deployment of channel bindings should take into consideration
both what information the EAP server is likely to know or have access
to, and also what type of network information the peer would want and
need authenticated.

5. Channel Binding Process

This section defines the process for verifying channel binding
information during an EAP authentication. The protocol uses the
approach where plaintext data is exchanged, since it allows channel
bindings to be used more flexibly in varied deployment models (see
Section 4.1). In the first subsection, the general communication
infrastructure is outlined, the messages used for channel binding
verifications are specified, and the protocol flows are defined. The
second subsection explores the difficulties of checking the different
pieces of information that are exchanged during the channel binding
protocol for consistency. The third subsection describes the
information carried in the EAP exchange.

5.1. Protocol Operation

Channel bindings are always provided between two communication
endpoints, here the EAP peer and the EAP server, who communicate
through an authenticator typically in pass-through mode. For the
channel binding protocol presented in this draft to work, the EAP

server needs to be able to access information from the AAA server
that is utilized during the EAP session and a local database. For
example, the EAP server and the local database can be co-located with
the AAA server, as illustrated in Figure 1. An alternate
architecture would be to provide a mechanism for the EAP server to
inform the AAA server what channel binding attributes were supplied
and the AAA server to inform the EAP server about what channel
binding attributes it considered when making its decision.

```
                                         + ------------------------+
   --------          ------------        |  ----------    _____   |
  |EAP peer|<---->|Authenticator|<--> |  |EAP Server|___(_____)  |
   --------          ------------        |  ----------    | DB   | |
        .                 .              |AAA            (_____)  |
        .      i1          .             +------------------------+
       .<---------------.       i2       .         .
        .               .------------>             .
        .               i1                         .
       .------------------------------------------>.
        .      CB_success/failure(i1, i2,info)     .
       .<------------------------------------------.
```
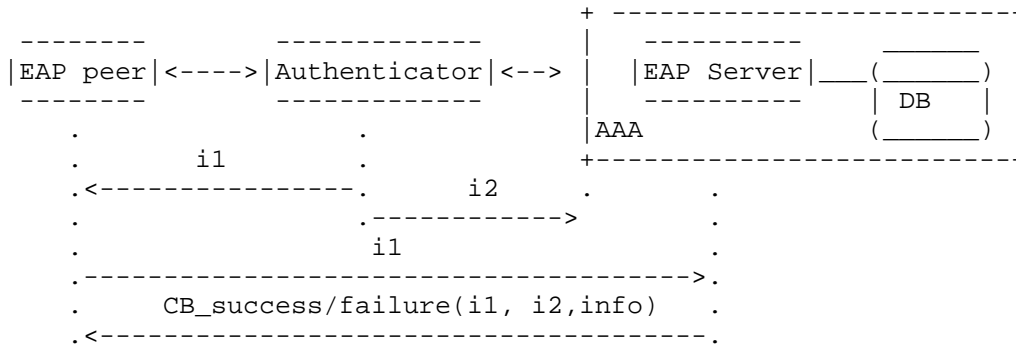
Figure 1: Overview of Channel Binding Protocol

During network advertisement, selection, and authentication, the
authenticator presents unauthenticated information, labeled i1, about
the network to the peer. Message i1 could include an authenticator
identifier and the identity of the network it represents, in addition
to advertised network information such as offered services and
roaming information. Information may be communicated implicitly in
i1, such as the type of media in use. As there is no established
trust relationship between the peer and authenticator, there is no
way for the peer to validate this information.

Additionally, during the transaction the authenticator presents a
number of information properties in form of AAA attributes about
itself and the current request to the AAA infrastructure which may or
may not be valid. This information is labeled i2. Message i2 is the
information the AAA server receives from the last hop in the AAA
proxy chain which is not necessarily the authenticator.

AAA hops between the authenticator and AAA server can validate some
of I2. Whether the AAA server will be able to depend on this depends
significantly on the business relationship executed with these
proxies and on the structure of the AAA network.

The local database is perhaps the most important part of this system.

   In order for the EAP server or AAA server to know whether i1 and i2
   are correct, they need access to trustworthy information, since an
   authenticator could include false information in both i1 and i2.
   Additional reasons why such a database is necessary for channel
   bindings to work are discussed in the next subsection. The
   information contained within the database could involve wildcards.
   For example, this could be used to check whether WiFi access points
   on a particular IP subnet all use a specific SSID. The exact IP
   address is immaterial, provided it is on the correct subnet.

   During an EAP method execution with channel bindings, the peer sends
   i1 to the EAP server using the mechanism described in Section 5.3.
   the EAP server verifies the consistency of i1 provided by the peer,
   i2 provided by the authenticator, and the information in the local
   database. Upon the check, the EAP server sends a message to the peer
   indicating whether the channel binding validation check succeeded or
   failed and includes the attributes thatwere used in the check. The
   message flow is illustrated in Figure 1.

   Above, the EAP server is described as performing the channel binding
   validation. In most deployments, this will be a necessary
   implementation constraint. The EAP exchange needs to include an
   indication of channel binding success or failure. Most existing
   implementations do not have a way to have an exchange between the EAP
   server and another AAA entity during the EAP server's processing of a
   single EAP message. However another AAA entity can provide
   information to the EAP server to make its decision.

   If the compliance of i1 or i2 information with the authoritative
   policy source is mandatory and a consistency check failed, then after
   sending a protected indication of failed consistency, the EAP server
   MUST send an EAP-Failure message to terminate the session. If the
   EAP server is otherwise configured, it MUST allow the EAP session to
   complete normally, and leave the decision about network access up to
   the peer's policy. If i1 or i2 does not comply with policy, the EAP
   server MUST not list information that failed to comply in the set of
   information used to perform channel binding. In this case the EAP
   server SHOULD indicate channel binding failure; this requirement may
   be upgraded to a MUST in the future.

5.2. Channel Binding Consistency Check

   The validation check that is the core of the channel binding protocol
   described in the previous subsection, consists of two parts in which
   the server checks whether:

   1. the authenticator is lying to the peer, i.e. i1 contains false
       information,

   2. the authenticator or any entity on the AAA path to the AAA server
       provides false information in form of AAA attributes, i.e. i2
       contains false information,

   These checks enable the EAP server to detect lying NAS/authenticator
   in enterprise networks and lying providers in service provider
   networks.

   Checking the consistency of i1 and i2 is nontrivial, as has been
   pointed out already in [HC07]. First, i1 can contain any type of
   information propagated by the authenticator, whereas i2 is restricted
   to information that can be carried in AAA attributes. Second,
   because the authenticator typically communicates over different link
   layers with the peer and the AAA infrastructure, different type of
   identifiers and addresses may have been presented to both
   communication endpoints. Whether these different identifiers and
   addresses belong to the same device cannot be directly checked by the
   EAP server or AAA server without additional information. Finally, i2
   may be different from the original information sent by the
   authenticator because of en route processing or malicious
   modifications. As a result, in the service provider model, typically
   the i1 information available to the EAP server can only be verified
   against the last-hop portion of i2, or values propagated by proxy
   servers. In addition, checking the consistency of i1 and i2 alone is
   insufficient because an authenticator could lie to both, the peer and
   the EAP server, i.e. i1 and i2 may be consistent but both contain
   false information.

   A local database is required to leverage the above mentioned
   shortcomings and support the consistency and validation checks. In
   particular, information stored for each NAS/authenticator (enterprise
   scenario) or each roaming partner (service provider scenario) enables
   a comparison of any information received in i1 with AAA attributes in
   i2 as well as additionally stored AAA attributes that might have gone
   lost in transition. Furthermore, only such a database enables the
   EAP server and AAA server to check the received information against
   trusted information about the network including roaming agreements.

   Section 7 describes lower-layer specific properties that can be
   exchanged as a part of i1. Section 8 describes specific AAA
   attributes that can be included and evaluated in i2. The EAP server
   reports back the results from the channel binding validation check
   that compares the consistency of all the values with those in the
   local database. The challenges of setting up such a local database
   are discussed in Section 10.

5.3. EAP Protocol

   EAP methods supporting channel binding consistent with this
   specification provide a mechanism for carrying channel binding data
   from the peer to the EAP server and a channel binding response from
   the EAP server to the peer. The specifics of this mechanism are
   dependent on the method, although the content of the channel binding
   data and channel binding response are defined by this section.

   Typically the lower layer will communicate a set of attributes to the
   EAP implementation on the peer that should be part of channel
   binding. The EAP implementation may need to indicate to the lower
   layer that channel binding information cannot be sent. Reasons for
   failing to send channel binding information include an EAP method
   that does not support channel binding is selected, or channel binding
   data is too big for the EAP method selected. Peers SHOULD provide
   appropriate policy controls to select channel binding or mandate its
   success.

   The EAP server receives the channel binding data and performs the
   validation. The EAP method provides a way to return a response; the
   channel binding response uses the same basic format as the channel
   binding data.

   XXX Nice figure goes here. Both the channel binding data and
   response use the following format. The protocol starts with a one
   byte code; see Section 5.3.1. Then there is a set of attributes
   containing the following information:
   len: one octet of length, not including the length or the namespace
      ID; up to 255 octets of namespace-specific information can be
      included.
   NSID: One octet describing the namespace from which the attribute is
      drawn. See Section 5.3.3 describes how to encode RADIUS
      attributes in channel binding data and responses. RADIUS uses a
      namespace identifier of 0.
   NSSPECIFIC: Up to 255 octets of namespace specific data.

   In channel binding data, the code is set to 0 (channel binding data)
   and the full attributes and values are included. In a channel
   binding response, the server selects the code; see Section 5.3.1.
   The server includes any attributes that were considered in making the
   channel binding decision. The server SHOULD only include the
   attributes but not their values. In some cases, a server may not
   have information sufficient to know where the attribute stops and the
   value starts; as an example, consider attribute containers such as
   the RADIUS vendor-specific attribute. For this reason and for future
   extensibility, clients MUST ignore any values sent in the channel
   binding response.

5.3.1. Channel Binding Codes

```
+------+-------------------------------+
| Code | Meaning                       |
+------+-------------------------------+
| 0    | Channel Binding data from client |
| 1    | Channel binding success       |
| 2    | Channel binding failure       |
+------+-------------------------------+
```

5.3.2. Namespace Identifiers

```
+-----+------------+--------------+
| ID  | Namespace  | Reference    |
+-----+------------+--------------+
| 0   | RADIUS     | Section 5.3.3 |
| 255 | Private Use |              |
+-----+------------+--------------+
```

5.3.3. Radius Namespace

   RADIUS AVPs are encoded with a one-octet attribute type followed by
   the value of the RADIUS attribute being encoded. The length of the
   attribute is not included because it is already included prior to the
   namespace identifier.

   XXX nice figure goes here.

   The full value of an attribute is included in the channel binding
   data. For most attributes, none of the value is included in a
   channel binding response. However for attributes such as the vendor-
   specific attribute that include sub-attributes, enough information
   SHOULD be included to identify the sub-attribute. This means that
   the internal lengths SHOULD be 0 but the attribute information is
   included. XXX example figure of a RFC-style VSA in a channel binding
   response.

6. System Requirements

   This section defines requirements on components used to implement the
   channel bindings protocol.

   The channel binding protocol defined in this document must be
   transported after keying material has been derived between the EAP
   peer and server, and before the peer would suffer adverse affects
   from joining an adversarial network. This document describes a
   protocol for performing channel binding within EAP methods. As

discussed in Section 4.2, an alternative approach for meeting this
requirement is to perform channel bindings during the secure
association protocol of the lower layer.

6.1. General Transport Protocol Requirements

The transport protocol for carrying channel binding information MUST
support end-to-end (i.e. between the EAP peer and server) message
integrity protection to prevent the adversarial NAS or AAA device
from manipulating the transported data. The transport protocol
SHOULD provide confidentiality. The motivation for this that the
channel bindings could contain private information, including peer
identities, which SHOULD be protected. If confidentiality cannot be
provided, private information MUST NOT be sent as part of the channel
binding information.

One way to transport the single round-trip exchange is as a series of
TLVs formatted and encapsulated in EAP methods. These TLVs carry
different types of data. Since i2 messages are carried within a AAA
protocol it is useful to define one type of data carried as AAA AVPs,
but other types of data may be defined that are not carried in AAA
attributes and are only compared against the information stored in
the local database. This document describes some AAA attributes that
are useful for channel binding checks. Additionally, guidance on how
to perform consistency checks on those values will be provided.
Since the Diameter namespace contains the RADIUS namespace the TLVs
of AAA AVP type carry Diameter attributes.

Any transport needs to be careful not to exceed the MTU for its
lower-layer medium. In particular, if channel binding information is
exchanged within protected EAP method channels, these methods may or
may not support fragmentation. In order to work with all methods,
the channel binding messages must fit within the available payload.
For example, if the EAP MTU is 1020 octets, and EAP-GPSK is used as
the authentication method, and maximal-length identities are used, a
maximum of 384 octets are available for conveying channel binding
information. Other methods, such as EAP-TTLS, support fragmentation
and could carry significantly longer payloads.

6.2. EAP Method Requirements

If transporting data directly within an EAP method, it MUST be able
to carry integrity protected data from the EAP peer to server. EAP
methods SHOULD provide a mechanism to carry protected data from
server to peer. EAP methods MUST exchange channel binding data with
the AAA subsystem hosting the EAP server. EAP methods MUST be able
to import channel binding data from the lower layer on the EAP peer.

7. Channel Binding TLV

   This section defines some channel binding TLVs. While message i1 is
   not limited to AAA attributes, for the sake of tangible attributes
   that are already in place, this section discusses AAA AVPs that are
   appropriate for carrying channel bindings (i.e. data from i1 in
   Section 5). In particular, attributes for IEEE 802.11 are provided,
   which can be used as a template for developing bindings for other EAP
   lower-layer protocols.

   For any lower-layer protocol, network information of interest to the
   peer and server can be encapsulated in AVPs or other defined payload
   containers. The appropriate AVPs depend on the lower layer protocol
   as well as on the network type (i.e. enterprise network or service
   provider network) and its application. Additional TLV types can be
   defined beyond AAA AVPs. For example it may be useful to define TLVs
   that can carry 802.11 information elements.

7.1. Requirements for Lower-Layer Bindings

   Lower-layer protocols MUST support EAP in order to support EAP
   channel bindings. These lower layers MUST support EAP methods that
   derive keying material, as otherwise no integrity-protected channel
   would be available to execute the channel bindings protocol. Lower-
   layer protocols need not support traffic encryption, since this is
   independent of the authentication phase.

   Any binding value that is communicated in AAA MUST be encoded as a
   Diameter AVP. The data conveyed within the AVP type MUST NOT
   conflict with the externally-defined usage of the AVP. Additional
   TLV types SHOULD be defined for values that are not communicated
   within AAA attributes.

7.2. General Attributes

   This section lists AAA AVPs useful to all link-layers. The peer
   SHOULD transmit to the server the following fields, encapsulated
   within the appropriate Diameter AVPs:

   NAS-Port-Type: Indicates the underlying link-layer technology used
      to connect (e.g. IEEE 802.11, PPP, etc), and SHOULD be included
      by the EAP peer, and SHOULD be verified against the database and
      NAS-Port-Type received from the NAS.

Cost-Information: AVP from the Diameter Credit-Control Application
    [RFC4006] to the peer indicating how much peers will be billed for
    service and MAY be included by the EAP peer and verified against
    roaming profiles stored in the database.

7.3. IEEE 802.11

The peer SHOULD transmit to the server the following fields,
encapsulated within the appropriate Diameter AVPs:

Called-Station-Id: contains BSSID and SSID and SHOULD be verified
    against the database and Called-Station-Id received from the NAS

7.3.1. IEEE 802.11r

In addition to the AVPs for IEEE 802.11, an IEEE 802.11r client
SHOULD transmit the following additional field:

Mobility-Domain-Id: contains the identity of the mobility domain and
    SHOULD be verified against the database and Mobility-Domain-Id
    received from the NAS [I-D.aboba-radext-wlan]

8. AAA-Layer Bindings

This section discusses which AAA attributes in a AAA Accept-Request
messages can and should be validated by a AAA server (i.e. data from
i2 in Section 5). As noted before, this data can be manipulated by
AAA proxies either to enable functionality (e.g. removing realm
information after messages have been proxied) or maliciously (e.g. in
the case of a lying provider). As such, this data cannot always be
easily validated. However as thorough of a validation as possible
should be conducted in an effort to detect possible attacks.

User-Name: This value should be checked for consistency with the
    database and any method-specific user information. If EAP method
    identity protection is employed, this value typically contains a
    pseudonym or keyword.

NAS-IP-Address: This value is typically the IP address of the
    authenticator, but in a proxied connection it likely will not
    match the source IP address of an Access-Request. A consistency
    check MAY verify the subnet of the IP address was correct based on
    the last-hop proxy.

NAS-IPv6-Address: This value is typically the IPv6 address of the
   authenticator, but in a proxied connection it likely will not
   match the source IPv6 address of an Access-Request. A consistency
   check MAY verify the subnet of the IPv6 address was correct based
   on the last-hop proxy.

Called-Station-Id: This is typically the MAC address of the NAS. On
   an enterprise network, it MAY be validated against the MAC address
   is one that has been provisioned on the network.

Calling-Station-Id: This is typically the MAC address of the EAP
   peer, and verification of this is likely difficult, unless EAP
   credentials have been provisioned on a per-host basis to specific
   L2 addresses. It SHOULD be validated against the database in an
   enterprise deployment.

NAS-Identifier: This is an identifier populated by the NAS, and
   could be related to the MAC address, and should be validated
   similarly to the Called-Station-Id.

NAS-Port-Type: This specifies the underlying link technology. It
   SHOULD be validated against the value received from the peer in
   the information exchange, and against a database of authorized
   link-layer technologies.

9. Security Considerations

   This section discusses security considerations surrounding the use of
   EAP channel bindings.

9.1. Trust Model

   In the considered trust model, EAP peer and authentication server are
   honest while the authenticator is maliciously sending false
   information to peer and/or server. In the model, the peer and server
   trust each other, which is not an unreasonable assumption,
   considering they already have a trust relationship. The following
   are the trust relationships:

   o The server trusts that the channel binding information received
      from the peer is the information that the peer received from the
      authenticator.
   o The peer trusts the channel binding result received from the
      server.
   o The server trusts the information contained within its local
      database.

In order to establish the first two trust relationships during an EAP
execution, an EAP method needs to provide the following:

o mutual authentication between peer and server
o derivation of keying material including a key for integrity
   protection of channel binding messages
o sending i1 from peer to server over an integrity-protected channel
o sending the result and optionally i2 from server to peer over an
   integrity-protected channel

9.2. Consequences of Trust Violation

   If any of the trust relationships listed in Section 9.1 are violated,
   channel binding cannot be provided. In other words, if mutual
   authentication with key establishment as part of the EAP method as
   well as protected database access are not provided, then achieving
   channel binding is not feasible.

   Dishonest peers can only manipulate the first message i1 of the
   channel binding protocol. In this scenario, a peer sends i1' to the
   server. If i1' is invalid, the channel binding validation will fail.
   On the other hand if i1' passes the validation, either the original
   i1 was wrong and i1' corrected the problem or both i1 and i1'
   constitute valid information. A peer could potentially gain an
   advantage in auditing or charging if both are valid and information
   from i1 is used for auditing or charging. Such peers can be detected
   by including the information in i2 and checking i1 against i2.

   Dishonest servers can send EAP-Failure messages and abort the EAP
   authentication even if the received i1 is valid. However, servers
   can always abort any EAP session independent of whether channel
   binding is offered or not. On the other hand, dishonest servers can
   claim a successful validation even if i1 contains invalid
   information. This can be seen as collaboration of authenticator and
   server. Channel binding can neither prevent nor detect such attacks.
   In general such attacks cannot be prevented by cryptographic means
   and should be addressed using policies making servers liable for
   their provided information and services.

   Additional network entities (such as proxies) might be on the
   communication path between peer and server and may attempt to
   manipulate the channel binding protocol. If these entities do not
   possess the keying material used for integrity protection of the
   channel binding messages, the same threat analysis applies as for the
   dishonest authenticators. Hence, such entities can neither
   manipulate single channel binding messages nor the outcome. On the
   other hand, entities with access to the keying material must be
   treated like a server in a threat analysis. Hence such entities are

able to manipulate the channel binding protocol without being
detected. However, the required knowledge of keying material is
unlikely since channel binding is executed before the EAP method is
completed, and thus before keying material is typically transported
to other entities.

9.3. Privacy Violations

While the channel binding information exchanged between EAP peer and
EAP server (i.e. i1 and the optional result message) must always be
integrity-protected it may not be encrypted. In the case that these
messages contain identifiers of peer and/or network entities, the
privacy property of the executed EAP method may be violated. Hence,
in order to maintain the privacy of an EAP method, the exchanged
channel binding information must be encrypted. If encryption is not
available, private information is not sent as part of the channel
binding information, as described in Section 6.1.

10. Operations and Management Considerations

As with any extension to existing protocols, there will be an impact
on existing systems. Typically the goal is to develop an extension
that minimizes the impact on both development and deployment of the
new system, subject to the system requirements. This section
discusses the impact on existing devices that currently utilize EAP,
assuming the channel binding information is transported within the
EAP method execution.

The EAP peer will need an API between the EAP lower layer and the EAP
method that exposes the necessary information from the NAS to be
validated to the EAP peer, which can then feed that information into
the EAP methods for transport. For example, an IEEE 802.11 system
would need to make available the various information elements that
require validation to the EAP peer which would properly format them
and pass them to the EAP method. Additionally, the EAP peer will
require updated EAP methods that support transporting channel binding
information. While most method documents are written modularly to
allow incorporating arbitrary protected information, implementations
of those methods would need to be revised to support these
extensions. Driver updates are also required so methods can access
the required information.

No changes to the pass-through authenticator would be required.

The EAP server would need an API between the database storing NAS
information and the individual EAP server. The database may already
exist on the AAA server in which case the EAP server passes the

parameters to the AAA server for validation. The EAP methods need to
be able to export received channel binding information to the EAP
server so it can be validated.

11. IANA Considerations

   A new top level registry is created for "EAP Channel Binding
   Parameters." This registry consists of several sub registries.

   The "Channel Binding Codes" registry defines values for the code
   field in the channel binding data and channel binding response
   packet. See the table in Section 5.3.1 for initial registrations.
   This registry requires standards action [RFC5226] for new
   registrations. Early allocation is allowed. An additional reference
   column should be added to the table for the registry, pointing all
   codes in the initial registration to this specification.

   The "Channel Binding Namespaces" sub-registry contains registrations
   for the NSID field in the channel binding data and channel binding
   response. Initial registrations are found in the table in
   Section 5.3.2. Registrations in this registry require IETF review.

12. Acknowledgements

   The authors and editor would like to thank Bernard Aboba, Glen Zorn,
   Joe Salowey, and Klaas Wierenga for their valuable inputs that helped
   to improve and shape this document over the time.

   Sam hartman's work on this specification is funded by JANET(UK).

13. References

13.1. Normative References

   [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate
              Requirement Levels", BCP 14, RFC 2119, March 1997.

   [RFC3748] Aboba, B., Blunk, L., Vollbrecht, J., Carlson, J., and H.
              Levkowetz, "Extensible Authentication Protocol (EAP)",
              RFC 3748, June 2004.

   [RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an
              IANA Considerations Section in RFCs", BCP 26, RFC 5226,
              May 2008.

13.2. Informative References

   [I-D.aboba-radext-wlan]
              Aboba, B., Malinen, J., Congdon, P., and J. Salowey,
              "RADIUS Attributes for IEEE 802 Networks",
              draft-aboba-radext-wlan-13 (work in progress),
              February 2010.

   [I-D.clancy-emu-aaapay]
              Clancy, T., Lior, A., and G. Zorn, Ed., "EAP Method
              Support for Transporting AAA Payloads", Internet
              Draft draft-clancy-emu-aaapay-02, May 2009.

   [RFC4006] Hakala, H., Mattila, L., Koskinen, J-P., Stura, M., and J.
              Loughney, "Diameter Credit-Control Application", RFC 4006,
              August 2005.

   [RFC4017] Stanley, D., Walker, J., and B. Aboba, "Extensible
              Authentication Protocol (EAP) Method Requirements for
              Wireless LANs", RFC 4017, March 2005.

   [RFC5056] Williams, N., "On the Use of Channel Bindings to Secure
              Channels", RFC 5056, November 2007.

   [HC07]   Hoeper, K. and L. Chen, "Where EAP Security Claims Fail",
              ICST QShine, August 2007.

   [80211U-D4.01]
              "Information technology - Telecommunications and
              information exchange between systems - Local and
              metropolitan area networks - Specific requirements - Part
              11: Wireless LAN Medium Access Control (MAC) and Physical
              Layer (PHY) specifications - Amendment 7: Interworking
              with External Networks", IEEE Draft Standard 802.11u,
              November 2008.

   [I-D.ietf-abfab-gss-eap]
              Hartman, S. and J. Howlett, "A GSS-API Mechanism for the
              Extensible Authentication Protocol",
              draft-ietf-abfab-gss-eap-00 (work in progress),
              October 2010.

Appendix A. Attacks Prevented by Channel Bindings

   In the following it is demonstrated how the presented channel
   bindings can prevent attacks by malicious authenticators
   (representing the lying NAS problem) as well as malicious visited

   networks (representing the lying provider problem).

A.1. Enterprise Subnetwork Masquerading

   As outlined in Section 3, an enterprise network may have multiple
   VLANs providing different levels of security. In an attack, a
   malicious NAS connecting to a guest network with lesser security
   protection could broadcast the SSID of a subnetwork with higher
   protection. This could lead peers to believe that they are accessing
   the network over secure connections, and, e.g., transmit confidential
   information that they normally would not send over a weakly protected
   connection. This attack works under the conditions that peers use
   the same set of credentials to authenticate to the different kinds of
   VLANs and that the VLANs support at least one common EAP method. If
   these conditions are not met, the EAP server would not authorize the
   peers to connect to the guest network, because the peers used
   credentials and/or an EAP method that is associated with the
   corporate network.

A.2. Forced Roaming

   Mobile phone providers boosting their cell tower's transmission power
   to get more users to use their networks have occurred in the past.
   The increased transmission range combined with a NAS sending a false
   network identity lures users to connect to the network without being
   aware of that they are roaming.

   Channel bindings would detect the bogus network identifier because
   the network identifier send to the authentication server in i1 will
   neither match information i2 nor the stored data. The verification
   fails because the info in i1 claims to come from the peer's home
   network while the home authentication server knows that the
   connection is through a visited network outside the home domain. In
   the same context, channel bindings can be utilized to provide a "home
   zone" feature that notifies users every time they are about to
   connect to a NAS outside their home domain.

A.3. Downgrading attacks

   A malicious authenticator could modify the set of offered EAP methods
   in its Beacon to force the peer to choose from only the weakest EAP
   method(s) accepted by the authentication server. For instance,
   instead of having a choice between EAP-MD5-CHAP, EAP-FAST and some
   other methods, the authenticator reduces the choice for the peer to
   the weaker EAP-MD5-CHAP method. Assuming that weak EAP methods are
   supported by the authentication server, such a downgrading attack can
   enable the authenticator to attack the integrity and confidentiality
   of the remaining EAP execution and/or break the authentication and

key exchange. The presented channel bindings prevent such
downgrading attacks, because peers submit the offered EAP method
selection that they have received in the beacon as part of i1 to the
authentication server. As a result, the authentication server
recognizes the modification when comparing the information to the
respective information in its policy database.

A.4. Bogus Beacons in IEEE 802.11r

   In IEEE 802.11r, the SSID is bound to the TSK calculations, so that
   the TSK needs to be consistent with the SSID advertised in an
   authenticator's Beacon. While this prevents outsiders from spoofing
   a Beacon it does not stop a "lying NAS" from sending a bogus Beacon
   and calculating the TSK accordingly.

   By implementing channel bindings, as described in this draft, in IEEE
   802.11r, the verification by the authentication server would detect
   the inconsistencies between the information the authenticator has
   sent to the peer and the information the server received from the
   authenticator and stores in the policy database.

A.5. Forcing false authorization in IEEE 802.11i

   In IEEE 802.11i a malicious NAS can modify the beacon to make the
   peer believe it is connected to a network different from the on the
   peer is actually connected to.

   In addition, a malicious NAS can force an authentication server into
   authorizing access by sending an incorrect Called-Station-ID that
   belongs to an authorized NAS in the network. This could cause the
   authentication server to believe it had granted access to a different
   network or even provider than the one the peer got access to.

   Both attacks can be prevented by implementing channel bindings,
   because the server can compare the information that was sent to the
   peer, with information it received from the authenticator during the
   AAA communication as well as the information stored in the policy
   database.

Appendix B. Change History

   RFC editor, remove this section prior to publication.

B.1. Changes since Version 05

   The purpose of this revision is to provide a specific candidate
   protocol for channel binding data and channel binding responses.

B.2. Changes since version 04

   o Clarify examples in introduction.
   o In problem statement note that one EAP server may deal with both
      enterprise and provider networks.
   o Update discussion of the architecture. Talk about channel
      bindings as a mechanism to introduce levels of trust.
   o Indicate that this document is focusing on EAP channel bindings
      within methods while trying to do a better job of describing the
      SAP approach in more detail.
   o Claim that we're using the encoding from draft-clancy-emu-aaapay.
      The WG almost certainly doesn't have consensus on this, but in the
      interest of actually describing what the protocol might be like,
      it is a good straw-man proposal.
   o Update protocol description.

Authors' Addresses

   Sam Hartman (editor)
   Painless Security
   356 Abbott ST
   North Andover, MA 01845
   USA

   Email: hartmans-ietf@mit.edu

   T. Charles Clancy
   Laboratory for Telecommunications Sciences
   US Department of Defense
   College Park, MD 20740
   USA

   Email: clancy@LTSnet.net

   Katrin Hoeper
   Motorola, Inc.
   1301 E. Algonquin Road
   Schaumburg, IL 60196
   USA

   Email: khoeper@motorola.com