Network Working Group
INTERNET-DRAFT
Category: Standards Track
<draft-dekok-saag-dhcp-keys-00.txt>
24 October 2016

DeKok, Alan
Network RADIUS SARL

DHCP Keys via 802.1X

Abstract

While RFC 3118 made provisions for securing DHCP, it made no provisions for creating or distributing authentication keys. This specification describes how in some cases, DHCP keys can be automatically derived from 802.1X authentication. The pros and cons of this approach are also discussed

Status of this Memo

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

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt.

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

This Internet-Draft will expire on May 24, 2017.

Copyright Notice

Table of Contents

1. Introduction

There has been increased interest in, and awareness of, securing basic networking protocols such as DHCP [RFC2131]. While provisions were made in [RFC3118] for securing the protocol via secret keys, there is little discussion on how the secret keys are created or managed. This specification addresses that issue, for the limited case of DHCP which occurs after 802.1X authentication.

1.1. Problem Statement

This document addresses the situation where a client machine connects to the network via 802.1X / EAP, and where keying material is derived as part of the EAP conversation. Once the client machine authenticated to the network, it requests an IP address via DHCP.

However, there is essentially no communication or interaction between the AAA server which authenticates the client machine, and the DHCP server which allocates IP addresses. This lack of communication means that it is possible to attack the systems independent. That is, the two systems do not work together to increase security.

1.2. Proposed Solution

Have the AAA server derive a shared secret key for signing DHCP packets. And then use that key to sign DHCP packets.

1.3. Requirements Language

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].

2. Generating Keying Material

The algorithm used to generate the keying material is similar to that used for EAP methods, such as EAP-TLS ([RFC5216] Section 2.3), and TTLS ([RFC7542] Secton 8).

Upon successful conclusion of an EAP-TTLS negotiation, 128 octets of keying material are generated and exported for use in securing the data connection between client and access point. The first 4 octets of the keying material constitute the secret ID, the last 124 octets constitute the DHCP shared secret key.

The keying material is generated using the TLS PRF function [RFC5246], with inputs consisting of the TLS master secret, the ASCII-encoded constant string "dhcp keying material", the TLS client random, and the TLS server random. The constant string is not null-terminated.

```
Keying Material = PRF-128(SecurityParameters.master_secret, "dhcp
          keying material", SecurityParameters.client_random +
          SecurityParameters.server_random)

Secret ID = Keying Material [0..3]

DHCP Shared secret key = Keying Material [4..127]
```

We perhaps don't want to use the keying material directly in the Secret ID. Instead, maybe use the last 4 octets of the keying material? Which should give less information than the first 4 octets. Or, derive the secret ID from a different PRF? Or set it to a fixed ID, which indicates that the client is using this method for signing packets?

The lifetime of the key is the lifetime of the underlying authentication session. If the client machine re-authenticates, a new DHCP shared secret key is derived.

The lifetime of the key MUST be no longer than the lifetime of the underlying authentication session. That is, once the authentication session has expired, the client MUST discard the key along with the corresponding secret ID.

We should also do something useful with the RDM / Reply Detection fields.

2.1. Implementation Considerations

There has historically been little communication between DHCP servers and AAA servers. As such, there is no clear way for the AAA server to share the derived key with the DHCP server. We leave that problem for implementors to solve.

The DHCP server could receive a packet, and request the corresponding key from the AAA server. Or, it may hand the packet to the AAA server for verification and/or signing. Or, it may retrive the key from a secure co-located database. Or, the AAA server may pro-actively inform the DHCP server that a key exists, along with the keys value.

All of these scenarios are possible, and it is difficult to recommend any one in particular. We can, however make general security recommendations.

The keys are highly secret information. As such, any exchange of keys MUST be done in a secure manner. Keys MUST NOT be visible to any entity other than the DHCP server and AAA server. If keys are stored in a database, they MUST be encrypted with an encryption key known only to the DHCP server and AAA server.

2.2. What does the Signature mean?

While it is nice to sign a DHCP packet for security, it is not at all clear what is meant by the signature. The minimum we can say is that the signature means that the DHCP server has communicated with the AAA server, and obtained a copy of the key.

There is no way to know if the DHCP server is the correct one, or malicious, or under the control of an attacker. Though if that is true, there is no need for the DHCP server to obtain the key. It can just hand out any addresses it wants. And if you can compromise a DHCP server on the network, or run a rogue DHCP server, having signed packets is probably the least of your worries.

We suggest that the signature means (in the expected case), that the DHCP server is known to the AAA server, and is likely known to the network administrators, and is likely the correct DHCP server for the client to communicate with.

2.3. Open Questions

Q: Does this add security?

A: Perhaps. A larger discussion and analysis is necessary. It does

not appear to reduce security. i.e. forging the key is difficult to impossible. The most that can be done is to disable this mechanism, which reduces DHCP to it's current level of security.

Q: How does the AAA server indicate to the DHCP client that it has this capability?

A: There is no way for it to do so. The DHCP client can just sign packets speculatively.

Q: How does the DHCP server know to use the key?

A: RFC3118 has provisions for the client signalling that it has a key.

Q: What does a DHCP server do when it receives a message from the client indicating that the client has a key, but the DHCP server does not have the key?

A: RFC 3118 is silent on this topic. The likeliest response is for the DHCP server to ignore the signature.

Q: How many networks are technically capable of using 802.1X?

A: It's 2016, pretty much all of them.

Q: How many networks are administratively capable of using 802.1X?

A: Some. Not so much for home networks, WiFi hot spots, etc. Most enterprises, telcos (3G), and WiFI systems with Hotspot 2.0 should be capable.

Q: What is the cost of implementing this?

A: Unknown. Some modifications to AAA / DHCP servers. And communication between them via some method to be determined.

3. Security Considerations

This specification is concerned entirely with security. As such, additional security discussion is not necessary.

4. IANA Considerations

There are no IANA considerations for this document.

5. References

5.1. Normative References

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

[RFC2131]
Droms, R., "Dynamic Host Configuration Protocol", RFC 2131, March 1997.

[RFC2865]
Rigney, C., Willens, S., Rubens, A. and W. Simpson, "Remote Authentication Dial In User Service (RADIUS)", RFC 2865, June 2000.

[RFC3118]
Droms R. (Ed) and Arbaugh W. (Ed), "Authentication for DHCP Messages", RFC 3118, June 2001.

5.2. Informative References

[RFC5216]
Simon, D., Aboba, B., and Hurst, R, "The EAP-TLS Authentication Protocol", RFC 5216, March 2008.

[RFC5246]
Dierks, T., Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, August 2008.

[RFC5281]
Funk, P, and Blake-Wilson, S., "Extensible Authentication Protocol Tunneled Transport Layer Security Authenticated Protocol Version 0 (EAP-TTLSv0)", RFC 5281, August 2008

[RFC7542]
DeKok, A., "The Network Access Identifier", RFC 7542, May 2015.

Acknowledgments

None at this time.

Authors' Addresses

Alan DeKok
Network RADIUS SARL
100 Centrepointe Drive
Suite 200

Ottawa, ON
K2G 6B1
Canada

Email: aland@networkradius.com