Network Working Group
Request for Comments: 3118
Category: Standards Track

R. Droms, Editor
Cisco Systems
W. Arbaugh, Editor
University of Maryland

June 2001

Authentication for DHCP Messages

Status of this Memo

Copyright Notice

Abstract

This document defines a new Dynamic Host Configuration Protocol (DHCP) option through which authorization tickets can be easily generated and newly attached hosts with proper authorization can be automatically configured from an authenticated DHCP server. DHCP provides a framework for passing configuration information to hosts on a TCP/IP network. In some situations, network administrators may wish to constrain the allocation of addresses to authorized hosts. Additionally, some network administrators may wish to provide for authentication of the source and contents of DHCP messages.

1. Introduction

DHCP [1] transports protocol stack configuration parameters from centrally administered servers to TCP/IP hosts. Among those parameters are an IP address. DHCP servers can be configured to dynamically allocate addresses from a pool of addresses, eliminating a manual step in configuration of TCP/IP hosts.

Some network administrators may wish to provide authentication of the source and contents of DHCP messages. For example, clients may be subject to denial of service attacks through the use of bogus DHCP servers, or may simply be misconfigured due to unintentionally instantiated DHCP servers. Network administrators may wish to constrain the allocation of addresses to authorized hosts to avoid denial of service attacks in "hostile" environments where the network

medium is not physically secured, such as wireless networks or college residence halls.

This document defines a technique that can provide both entity authentication and message authentication. The current protocol combines the original Schiller-Huitema-Droms authentication mechanism defined in a previous work in progress with the "delayed authentication" proposal developed by Bill Arbaugh.

1.1 DHCP threat model

The threat to DHCP is inherently an insider threat (assuming a properly configured network where BOOTP ports are blocked on the enterprise's perimeter gateways.) Regardless of the gateway configuration, however, the potential attacks by insiders and outsiders are the same.

The attack specific to a DHCP client is the possibility of the establishment of a "rogue" server with the intent of providing incorrect configuration information to the client. The motivation for doing so may be to establish a "man in the middle" attack or it may be for a "denial of service" attack.

There is another threat to DHCP clients from mistakenly or accidentally configured DHCP servers that answer DHCP client requests with unintentionally incorrect configuration parameters.

The threat specific to a DHCP server is an invalid client masquerading as a valid client. The motivation for this may be for "theft of service", or to circumvent auditing for any number of nefarious purposes.

The threat common to both the client and the server is the resource "denial of service" (DoS) attack. These attacks typically involve the exhaustion of valid addresses, or the exhaustion of CPU or network bandwidth, and are present anytime there is a shared resource. In current practice, redundancy mitigates DoS attacks the best.

1.2 Design goals

These are the goals that were used in the development of the authentication protocol, listed in order of importance:

1. Address the threats presented in Section 1.1.
2. Avoid changing the current protocol.

3. Limit state required by the server.
4. Limit complexity (complexity breeds design and implementation errors).

1.3 Requirements Terminology

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 RFC 2119 [5].

1.4 DHCP Terminology

This document uses the following terms:

- "DHCP client"

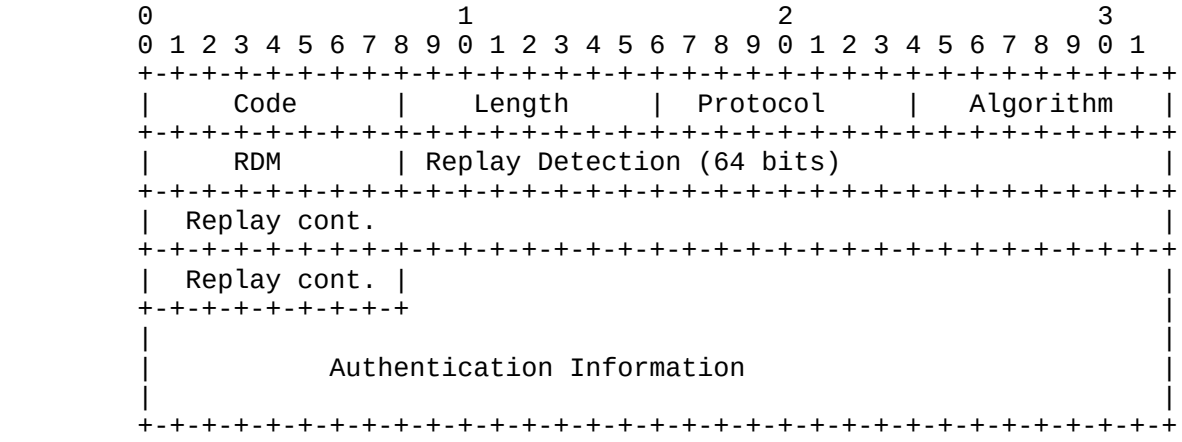
  A DHCP client or "client" is an Internet host using DHCP to obtain configuration parameters such as a network address.

- "DHCP server"

  A DHCP server or "server" is an Internet host that returns configuration parameters to DHCP clients.

2. Format of the authentication option

The following diagram defines the format of the DHCP authentication option:



The code for the authentication option is 90, and the length field contains the length of the protocol, RDM, algorithm, Replay Detection fields and authentication information fields in octets.

The protocol field defines the particular technique for authentication used in the option. New protocols are defined as described in Section 6.

The algorithm field defines the specific algorithm within the technique identified by the protocol field.

The Replay Detection field is per the RDM, and the authentication information field is per the protocol in use.

The Replay Detection Method (RDM) field determines the type of replay detection used in the Replay Detection field.

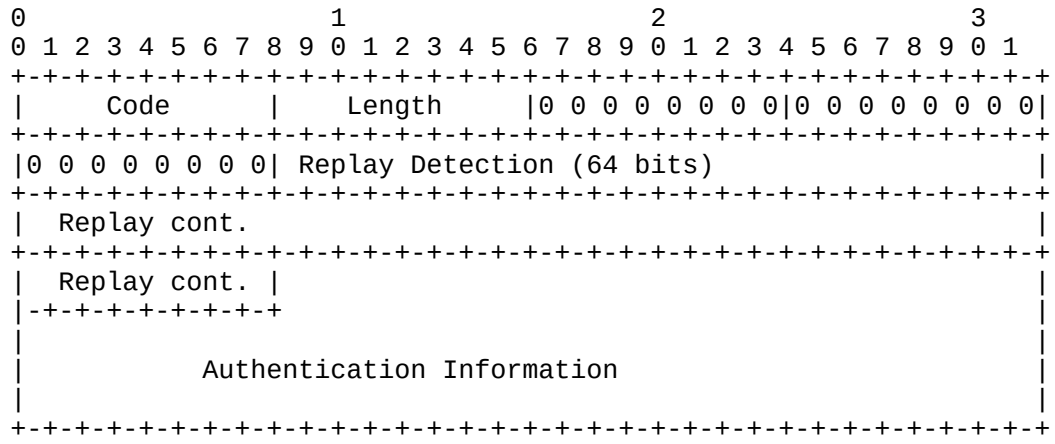
If the RDM field contains 0x00, the replay detection field MUST be set to the value of a monotonically increasing counter. Using a counter value such as the current time of day (e.g., an NTP-format timestamp [4]) can reduce the danger of replay attacks. This method MUST be supported by all protocols.

3. Interaction with Relay Agents

Because a DHCP relay agent may alter the values of the 'giaddr' and 'hops' fields in the DHCP message, the contents of those two fields MUST be set to zero for the computation of any hash function over the message header. Additionally, a relay agent may append the DHCP relay agent information option 82 [7] as the last option in a message to servers. If a server finds option 82 included in a received message, the server MUST compute any hash function as if the option were NOT included in the message without changing the order of options. Whenever the server sends back option 82 to a relay agent, the server MUST not include the option in the computation of any hash function over the message.

4. Configuration token

If the protocol field is 0, the authentication information field holds a simple configuration token:



The configuration token is an opaque, unencoded value known to both the sender and receiver. The sender inserts the configuration token in the DHCP message and the receiver matches the token from the message to the shared token. If the configuration option is present and the token from the message does not match the shared token, the receiver MUST discard the message.

Configuration token may be used to pass a plain-text configuration token and provides only weak entity authentication and no message authentication. This protocol is only useful for rudimentary protection against inadvertently instantiated DHCP servers.

DISCUSSION:

The intent here is to pass a constant, non-computed token such as a plain-text password. Other types of entity authentication using computed tokens such as Kerberos tickets or one-time passwords will be defined as separate protocols.

5. Delayed authentication

If the protocol field is 1, the message is using the "delayed authentication" mechanism. In delayed authentication, the client requests authentication in its DHCPDISCOVER message and the server replies with a DHCPOFFER message that includes authentication information. This authentication information contains a nonce value generated by the source as a message authentication code (MAC) to provide message authentication and entity authentication.

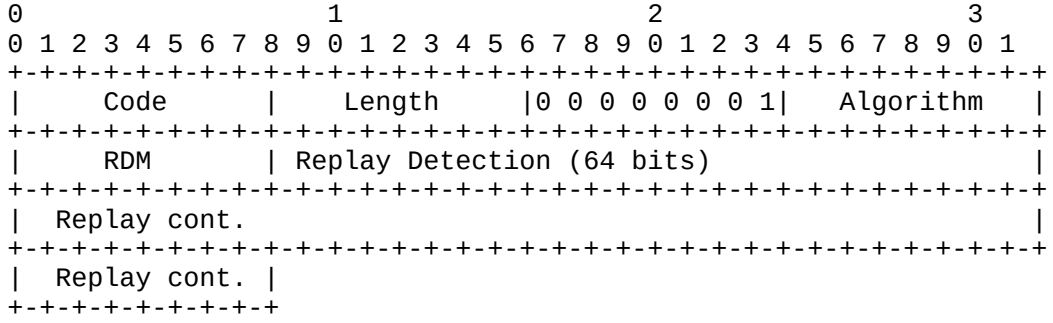
This document defines the use of a particular technique based on the HMAC protocol [3] using the MD5 hash [2].

5.1 Management Issues

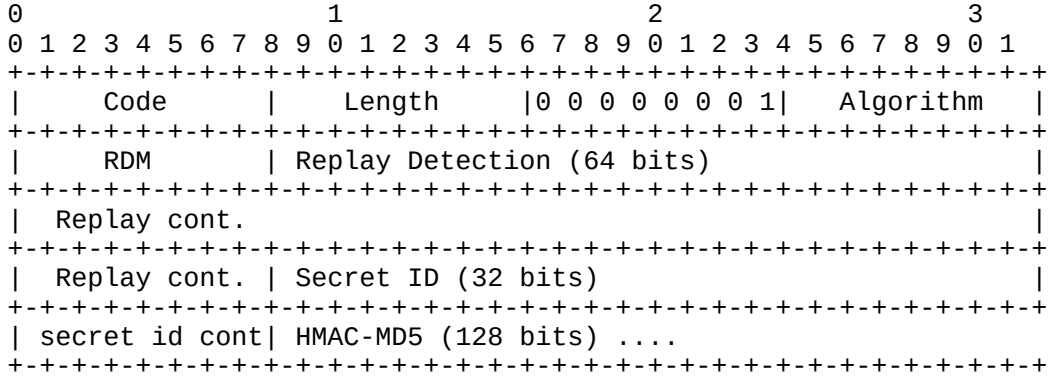
The "delayed authentication" protocol does not attempt to address situations where a client may roam from one administrative domain to another, i.e., interdomain roaming. This protocol is focused on solving the intradomain problem where the out-of-band exchange of a shared secret is feasible.

5.2 Format

The format of the authentication request in a DHCPDISCOVER or a DHCPINFORM message for delayed authentication is:



The format of the authentication information in a DHCPOFFER, DHCPREQUEST or DHCPACK message for delayed authentication is:



The following definitions will be used in the description of the authentication information for delayed authentication, algorithm 1:

Replay Detection	- as defined by the RDM field
K	- a secret value shared between the source and destination of the message; each secret has a unique identifier (secret ID)
secret ID	- the unique identifier for the secret value used to generate the MAC for this message
HMAC-MD5	- the MAC generating function [3, 2].

The sender computes the MAC using the HMAC generation algorithm [3] and the MD5 hash function [2]. The entire DHCP message (except as noted below), including the DHCP message header and the options field, is used as input to the HMAC-MD5 computation function. The 'secret ID' field MUST be set to the identifier of the secret used to generate the MAC.

DISCUSSION:

Algorithm 1 specifies the use of HMAC-MD5. Use of a different technique, such as HMAC-SHA, will be specified as a separate protocol.

Delayed authentication requires a shared secret key for each client on each DHCP server with which that client may wish to use the DHCP protocol. Each secret key has a unique identifier that can be used by a receiver to determine which secret was used to generate the MAC in the DHCP message. Therefore, delayed authentication may not scale well in an architecture in which a DHCP client connects to multiple administrative domains.

5.3 Message validation

To validate an incoming message, the receiver first checks that the value in the replay detection field is acceptable according to the replay detection method specified by the RDM field. Next, the receiver computes the MAC as described in [3]. The receiver MUST set the 'MAC' field of the authentication option to all 0s for computation of the MAC, and because a DHCP relay agent may alter the values of the 'giaddr' and 'hops' fields in the DHCP message, the contents of those two fields MUST also be set to zero for the computation of the MAC. If the MAC computed by the receiver does not match the MAC contained in the authentication option, the receiver MUST discard the DHCP message.

Section 3 provides additional information on handling messages that include option 82 (Relay Agents).

5.4 Key utilization

Each DHCP client has a key, K. The client uses its key to encode any messages it sends to the server and to authenticate and verify any messages it receives from the server. The client's key SHOULD be initially distributed to the client through some out-of-band mechanism, and SHOULD be stored locally on the client for use in all authenticated DHCP messages. Once the client has been given its key, it SHOULD use that key for all transactions even if the client's configuration changes; e.g., if the client is assigned a new network address.

Each DHCP server MUST know, or be able to obtain in a secure manner, the keys for all authorized clients. If all clients use the same key, clients can perform both entity and message authentication for all messages received from servers. However, the sharing of keys is strongly discouraged as it allows for unauthorized clients to masquerade as authorized clients by obtaining a copy of the shared key. To authenticate the identity of individual clients, each client MUST be configured with a unique key. Appendix A describes a technique for key management.

5.5 Client considerations

This section describes the behavior of a DHCP client using delayed authentication.

5.5.1 INIT state

When in INIT state, the client uses delayed authentication as follows:

1. The client MUST include the authentication request option in its DHCPDISCOVER message along with a client identifier option [6] to identify itself uniquely to the server.

2. The client MUST perform the validation test described in section 5.3 on any DHCPOFFER messages that include authentication information. If one or more DHCPOFFER messages pass the validation test, the client chooses one of the offered configurations.

   Client behavior if no DHCPOFFER messages include authentication information or pass the validation test is controlled by local policy in the client. According to client policy, the client MAY choose to respond to a DHCPOFFER message that has not been authenticated.

The decision to set local policy to accept unauthenticated messages should be made with care. Accepting an unauthenticated DHCPOFFER message can make the client vulnerable to spoofing and other attacks. If local users are not explicitly informed that the client has accepted an unauthenticated DHCPOFFER message, the users may incorrectly assume that the client has received an authenticated address and is not subject to DHCP attacks through unauthenticated messages.

A client MUST be configurable to decline unauthenticated messages, and SHOULD be configured by default to decline unauthenticated messages. A client MAY choose to differentiate between DHCPOFFER messages with no authentication information and DHCPOFFER messages that do not pass the validation test; for example, a client might accept the former and discard the latter. If a client does accept an unauthenticated message, the client SHOULD inform any local users and SHOULD log the event.

3. The client replies with a DHCPREQUEST message that MUST include authentication information encoded with the same secret used by the server in the selected DHCPOFFER message.

4. If the client authenticated the DHCPOFFER it accepted, the client MUST validate the DHCPACK message from the server. The client MUST discard the DHCPACK if the message fails to pass validation and MAY log the validation failure. If the DHCPACK fails to pass validation, the client MUST revert to INIT state and returns to step 1. The client MAY choose to remember which server replied with a DHCPACK message that failed to pass validation and discard subsequent messages from that server.

   If the client accepted a DHCPOFFER message that did not include authentication information or did not pass the validation test, the client MAY accept an unauthenticated DHCPACK message from the server.

5.5.2 INIT-REBOOT state

When in INIT-REBOOT state, the client MUST use the secret it used in its DHCPREQUEST message to obtain its current configuration to generate authentication information for the DHCPREQUEST message. The client MAY choose to accept unauthenticated DHCPACK/DHCPNAK messages if no authenticated messages were received. The client MUST treat the receipt (or lack thereof) of any DHCPACK/DHCPNAK messages as specified in section 3.2 of [1].

5.5.3 RENEWING state

When in RENEWING state, the client uses the secret it used in its initial DHCPREQUEST message to obtain its current configuration to generate authentication information for the DHCPREQUEST message. If client receives no DHCPACK messages or none of the DHCPACK messages pass validation, the client behaves as if it had not received a DHCPACK message in section 4.4.5 of the DHCP specification [1].

5.5.4 REBINDING state

When in REBINDING state, the client uses the secret it used in its initial DHCPREQUEST message to obtain its current configuration to generate authentication information for the DHCPREQUEST message. If client receives no DHCPACK messages or none of the DHCPACK messages pass validation, the client behaves as if it had not received a DHCPACK message in section 4.4.5 of the DHCP specification [1].

5.5.5 DHCPINFORM message

Since the client already has some configuration information, the client may also have established a shared secret value, K, with a server. Therefore, the client SHOULD use the authentication request as in a DHCPDISCOVER message when a shared secret value exists. The client MUST treat any received DHCPACK messages as it does DHCPOFFER messages, see section 5.5.1.

5.5.6 DHCPRELEASE message

Since the client is already in the BOUND state, the client will have a security association already established with the server. Therefore, the client MUST include authentication information with the DHCPRELEASE message.

5.6 Server considerations

This section describes the behavior of a server in response to client messages using delayed authentication.

5.6.1 General considerations

Each server maintains a list of secrets and identifiers for those secrets that it shares with clients and potential clients. This information must be maintained in such a way that the server can:

* Identify an appropriate secret and the identifier for that secret for use with a client that the server may not have previously communicated with

* Retrieve the secret and identifier used by a client to which the server has provided previous configuration information

Each server MUST save the counter from the previous authenticated message. A server MUST discard any incoming message which fails the replay detection check as defined by the RDM avoid replay attacks.

DISCUSSION:

The authenticated DHCPREQUEST message from a client in INIT-REBOOT state can only be validated by servers that used the same secret in their DHCPOFFER messages. Other servers will discard the DHCPREQUEST messages. Thus, only servers that used the secret selected by the client will be able to determine that their offered configuration information was not selected and the offered network address can be returned to the server's pool of available addresses. The servers that cannot validate the DHCPREQUEST message will eventually return their offered network addresses to their pool of available addresses as described in section 3.1 of the DHCP specification [1].

5.6.2 After receiving a DHCPDISCOVER message

The server selects a secret for the client and includes authentication information in the DHCPOFFER message as specified in section 5, above. The server MUST record the identifier of the secret selected for the client and use that same secret for validating subsequent messages with the client.

5.6.3 After receiving a DHCPREQUEST message

The server uses the secret identified in the message and validates the message as specified in section 5.3. If the message fails to pass validation or the server does not know the secret identified by the 'secret ID' field, the server MUST discard the message and MAY choose to log the validation failure.

If the message passes the validation procedure, the server responds as described in the DHCP specification. The server MUST include authentication information generated as specified in section 5.2.

5.6.4 After receiving a DHCPINFORM message

The server MAY choose to accept unauthenticated DHCPINFORM messages, or only accept authenticated DHCPINFORM messages based on a site policy.

When a client includes the authentication request in a DHCPINFORM message, the server MUST respond with an authenticated DHCPACK message. If the server does not have a shared secret value established with the sender of the DHCPINFORM message, then the server MAY respond with an unauthenticated DHCPACK message, or a DHCPNAK if the server does not accept unauthenticated clients based on the site policy, or the server MAY choose not to respond to the DHCPINFORM message.

6. IANA Considerations

Section 2 defines a new DHCP option called the Authentication Option, whose option code is 90.

This document specifies three new name spaces associated with the Authentication Option, which are to be created and maintained by IANA: Protocol, Algorithm and RDM.

Initial values assigned from the Protocol name space are 0 (for the configuration token Protocol in section 4) and 1 (for the delayed authentication Protocol in section 5). Additional values from the Protocol name space will be assigned through IETF Consensus, as defined in RFC 2434 [8].

The Algorithm name space is specific to individual Protocols. That is, each Protocol has its own Algorithm name space. The guidelines for assigning Algorithm name space values for a particular protocol should be specified along with the definition of a new Protocol.

For the configuration token Protocol, the Algorithm field MUST be 0. For the delayed authentication Protocol, the Algorithm value 1 is assigned to the HMAC-MD5 generating function as defined in section 5. Additional values from the Algorithm name space for Algorithm 1 will be assigned through IETF Consensus, as defined in RFC 2434.

The initial value of 0 from the RDM name space is assigned to the use of a monotonically increasing value as defined in section 2. Additional values from the RDM name space will be assigned through IETF Consensus, as defined in RFC 2434.

7. References

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

[2] Rivest, R., "The MD5 Message-Digest Algorithm", RFC 1321, April 1992.

[3] Krawczyk H., Bellare, M. and R. Canetti, "HMAC: Keyed-Hashing for Message Authentication", RFC 2104, February 1997.

[4] Mills, D., "Network Time Protocol (Version 3)", RFC 1305, March 1992.

[5] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", RFC 2219, March 1997.

[6] Alexander, S. and R. Droms, "DHCP Options and BOOTP Vendor Extensions", RFC 2132, March 1997.

[7] Patrick, M., "DHCP Relay Agent Information Option", RFC 3046, January 2001.

[8] Narten, T. and H. Alvestrand, "Guidelines for Writing and IANA Considerations Section in RFCs", BCP 26, RFC 2434, October 1998.

8. Acknowledgments

Jeff Schiller and Christian Huitema developed the original version of this authentication protocol in a terminal room BOF at the Dallas IETF meeting, December 1995. One of the editors (Droms) transcribed the notes from that discussion, which form the basis for this document. The editors appreciate Jeff’s and Christian’s patience in reviewing this document and its earlier drafts.

The "delayed authentication" mechanism used in section 5 is due to Bill Arbaugh. The threat model and requirements in sections 1.1 and 1.2 come from Bill’s negotiation protocol proposal. The attendees of an interim meeting of the DHC WG held in June, 1998, including Peter Ford, Kim Kinnear, Glenn Waters, Rob Stevens, Bill Arbaugh, Baiju Patel, Carl Smith, Thomas Narten, Stewart Kwan, Munil Shah, Olafur Gudmundsson, Robert Watson, Ralph Droms, Mike Dooley, Greg Rabil and Arun Kapur, developed the threat model and reviewed several alternative proposals.

The replay detection method field is due to Vipul Gupta.

Other input from Bill Sommerfield is gratefully acknowledged.

Thanks also to John Wilkins, Ran Atkinson, Shawn Mamros and Thomas Narten for reviewing earlier drafts of this document.

9. Security Considerations

This document describes authentication and verification mechanisms for DHCP.

9.1 Protocol vulnerabilities

The configuration token authentication mechanism is vulnerable to interception and provides only the most rudimentary protection against inadvertently instantiated DHCP servers.

The delayed authentication mechanism described in this document is vulnerable to a denial of service attack through flooding with DHCPDISCOVER messages, which are not authenticated by this protocol. Such an attack may overwhelm the computer on which the DHCP server is running and may exhaust the addresses available for assignment by the DHCP server.

Delayed authentication may also be vulnerable to a denial of service attack through flooding with authenticated messages, which may overwhelm the computer on which the DHCP server is running as the authentication keys for the incoming messages are computed.

9.2 Protocol limitations

Delayed authentication does not support interdomain authentication.

A real digital signature mechanism such as RSA, while currently computationally infeasible, would provide better security.

10. Editors' Addresses

Ralph Droms
Cisco Systems
300 Apollo Drive
Chelmsford, MA 01824

Phone: (978) 244-4733
EMail: rdroms@cisco.com

Bill Arbaugh
Department of Computer Science
University of Maryland
A.V. Williams Building
College Park, MD 20742

Phone: (301) 405-2774
EMail: waa@cs.umd.edu

Appendix A - Key Management Technique

To avoid centralized management of a list of random keys, suppose K for each client is generated from the pair (client identifier [6], subnet address, e.g., 192.168.1.0), which must be unique to that client. That is, K = MAC(MK, unique-id), where MK is a secret master key and MAC is a keyed one-way function such as HMAC-MD5.

Without knowledge of the master key MK, an unauthorized client cannot generate its own key K. The server can quickly validate an incoming message from a new client by regenerating K from the client-id. For known clients, the server can choose to recover the client's K dynamically from the client-id in the DHCP message, or can choose to precompute and cache all of the Ks a priori.

By deriving all keys from a single master key, the DHCP server does not need access to clear text passwords, and can compute and verify the keyed MACs without requiring help from a centralized authentication server.

To avoid compromise of this key management system, the master key, MK, MUST NOT be stored by any clients. The client SHOULD only be given its key, K. If MK is compromised, a new MK SHOULD be chosen and all clients given new individual keys.

Full Copyright Statement

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.