BFD Generic Cryptographic Authentication
draft-ietf-bfd-generic-crypto-auth-05

Abstract

   This document proposes an extension to Bidirectional Forwarding
   Detection (BFD) to allow the use of arbitary cryptographic
   authentication algorithms in addition to the already-documented
   authentication schemes described in the base specification. This
   document adds the basic infrastructure that is required for
   supporting algorithm and key agility for BFD.

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 RFC 2119 [RFC2119].

Table of Contents

1. Introduction

The base specification of bidirectional Forwarding Detection (BFD)
[RFC5880] defines five authentication schemes: Simple Password, Keyed
MD5 , Meticulous Keyed MD5, Keyed SHA-1, and Meticulous SHA-1. In
Simple Password, passwords are transferred in plaintext. An attacker
with physical access to the network can easily eavesdrop on the
password and compromise the security of the BFD packet exchanges. In
Keyed MD5 and Meticulous Keyed MD5, the BFD devices on the both sides
of a BFD session share a secret key which is used to generate a keyed
MD5 digest for each packet, and a monotonically increasing sequence
number scheme is used to prevent replay attacks. Keyed SHA-1 and
Meticulous SHA-1 modes are similar to MD5, and it uses SHA-1 instead
of MD5 to generate a digest for each packet.

A concern with existing authentication schemes of BFD is that the
security strength of the cryptographic algorithms adopted in the
schemes is relatively weak. Both the MD5 algorithm and the SHA-1
algorithm are known to be vulnerable to collision attacks. In

[MD5-attack] and [Dobb96a, Dobb96b], several methods of generating hash collisions for some applications of MD5 are proposed. Similar security vulnerabilities of SHA-1 are introduced in [SHA-1-attack1] and [SHA-1-attack2]. It is therefore desired that BFD must support newer algorithms that have not yet been broken. Additionally, the transition mechanism from one algorithm to the other must be seamless.

The other issue with the existing authentication schemes is the vulnerability to replay attacks. In non-meticulous authentication schemes, sequence numbers are only increased occasionally. This behavior can be taken advantage of by an attacker to perform intra-session replay attacks. In meticulous authentication schemes, sequence numbers are required to monotonically increase with each successive packet, which eliminates the possibility of intra-session replay attacks.

BFD session timers are often defined with the granularity of microseconds. Although in practice BFD devices send packets at millisecond intervals, they can potentially send packets at a much higher rate. Since the cryptographic sequence number space is only 32 bits, when using Meticulous Authentication, a sequence number used in a BFD session can reach its maximum value and roll over within a short period. For instance, if the value of a sequence number is increased by one every millisecond, then it will reach its maximum in less than 8 weeks. This can potentially be exploited to launch inter-session replay attacks.

In order to address the issues mentioned above, this document proposes two new authentication types that can be used to secure the BFD packets. The two authentication types are - Cryptographic Authentication (CRYPTO_AUTH) and Meticulous Cryptographic Authentication (MET_ CRYPTO_AUTH). Unlike earlier authentication types that were defined in BFD, the proposed authentication types are not tied to any particular authentication algorithm or construct. These can use different authentication algorithms and constructs like MD5, SHA-1, SHA-2, HMAC-SHA1, HMAC-SHA2, etc. to provide authentication and data integrity protection for BFD control packets.

The packet replay mechanism has also been enhanced to improve its capability in handling inter and intra-session replay attacks.

It should be noted that this document attempts to fix the security
issues raised by the manual key management procedure that currently
exists within BFD, as part of the Phase One described in KARP-design-
guide [I-D.ietf-karp-design-guide]. Therefore, only the pre-shared
keys is considered in this document. However, the solution described
in this document is generic and does not preclude the possibility of
supporting keys derived from an automated key management protocol.

2. BFD Security Association

The BFD protocol does not include an in-band mechanism to create or
manage BFD Security Associations (BFD SA). A BFD SA contains a set
of shared parameters between any two legitimate BFD devices.

The parameters associated with a BFD SA are listed as follows:

o Authentication Algorithm : This indicates the authentication
algorithm to be used with the BFD SA. This information SHOULD never
be sent in plaintext over the wire.

o Authentication Key : This indicates the cryptographic key
associated with this BFD SA. The length of this key is variable and
depends upon the authentication algorithm specified by the BFD SA.
Operators MUST ensure that this is never sent over the network in
clear-text via any protocol. Care should also be taken to ensure
that the selected key is unpredictable, avoiding any keys known to be
weak for the algorithm in use. [RFC4086] contains helpful
information on both key generation techniques and cryptographic
randomness.

o Authentication Key Identifier (Key ID) : This is a two octet
unsigned integer used to uniquely identify the BFD SA. This ID could
be manually configured by the network operator (or, in the future,
possibly by some key management protocol specified by the IETF). The
receiver determines the active SA by looking at this field in the
incoming packet. The sender puts this Key ID in the BFD packet based
on the active configuration. Using Key IDs makes changing keys while
maintaining protocol operation convenient. Normally, an
implementation would allow the network operator to configure a set of
keys in a key chain, with each key in the chain having fixed
lifetime. The actual operation of these mechanisms is outside the
scope of this document.

A key ID indicates a tuple of an authentication key and an associated
authentication algorithm. If a key is expected to be applied with
different algorithms, different Key IDs must be used to identify the
associations of the key with its authentication algorithms
respectively. However, the application of a key for different

purposes must be very careful, since it may make an adversary easier
to collect more material to compromise the key.

o Not Before Time : The time point before which the key should not be
used.

o Not After Time : The time point after which the key should not be
used.

3. Authentication Procedures

In the proposed authentication extension, an optional authentication
section (Generic Authentication Section) and two authentication types
(Generic Cryptographic Authentication and Generic Meticulous
Cryptographic Authentication) are specified.

3.1. Authentication Types

The Authentication section is only present in a BFD packet if the
Authentication Present (A) bit is set in the packet header. The Auth
Type in the Authentication section is set to 6 when Generic
Cryptographic Authentication is in use, while it is set to 7 when
Generic Meticulous Cryptographic Authentication is in use.

Both the authentication types use a monotonically increasing sequence
number to protect the BFD session against reply attacks. The only
difference between the two types is that the sequence number is
occasionally incremented in the Cryptographic Authentication mode, as
against the Meticulous Cryptographic Authentication mode, where it is
incremented on every packet.

As a result of this, in the Cryptographic Authentication scheme, a
replay attack is possible till the next sequence number is sent out.

3.2. Authentication Section Format

A new authentication type, 6 or 7, indicating the generic
cryptographic authentication mechanism in use, is inserted in the
first octet of Authentication Section of the BFD control packet.

For a BFD packet, if the Authentication Present (A) bit is set in the
header and the Authentication Type field is 6 (Generic Cryptographic
Authentication) or 7 (Generic Meticulous Cryptographic
Authentication), the Authentication Section has the following format:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

```
|   Auth Type   |   Auth Len    |          Auth Key ID          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|              Sequence Number (High Order 32 Bits)            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|              Sequence Number (Low Order 32 Bits)             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                              |
|                  Authentication Data (Variable)              |
|                                                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

o Auth Type: The Authentication Type, which in this case is 6
   (Cryptographic Authentication) or 7 (Meticulous Cryptographic
   Authentication).

o Auth Len: The length of the Authentication Section.

o Auth Key ID: The Key ID of the authentication key used for this
   packet, enabling multiple keys to be active simultaneously.

o Sequence Number: A 64-bit sequence number that is used to prevent
   replay attacks. For Cryptographic Authentication this value is
   incremented occasionally. For Meticulous Cryptographic
   Authentication, this value is incremented for each successive
   packet transmitted for a session.

o Authentication Data: This field carries the digest computed by
   whatever Cryptographic Authentication algorithm is being used to
   authenticate the BFD control packet.

3.3. Procedures at the Sending Side

Before a BFD device sends a BFD packet out, the device needs to
select an appropriate BFD SA from its local key database if a keyed
digest for the packet is required. If no appropriate SA is
available, the BFD packet MUST be discarded.

If an appropriate SA is available, the device then derives the key
and the associated authentication algorithm from the SA.

The device sets the Authentication Present (A) bit in the packet
header.

The device MUST fill the Auth Type and the Auth Len fields before the
authentication data is computed. The Sequence Number field MUST be
set to bfd.XmitAuthSeq.

The Auth Len field in the Authentication section is set as per the
authentication algorithm that is being used.

The Key ID field is filled.

The computation of the digest is performed. The computing process
can be various when different algorithms are adopted and is out of
the scope of this document.

The generated digest is placed in the Authentication Data field.

3.4. Procedure at the Receiving Side

When a BFD Control packet is received, the following procedure MUST
be followed, in the order specified.

If the Authentication Present (A) bit is set in the packet header and
the receiver will try to find a appropriate BFD SA in its local key
table to process the packet. The BFD SA is identified by the Key ID
field in the Authentication Section of the incoming BFD packet.

If the Auth Key ID field does not match the ID of any configured
authentication key or the associated key is not in its valid period,
the received packet MUST be discarded.

If bfd.AuthSeqKnown is 1, examine the Sequence Number field. For
Cryptographic Authentication, if the Sequence Number lies outside of
the range of bfd.RcvAuthSeq to bfd.RcvAuthSeq+(3*Detect Mult)
inclusive (when treated as an unsigned 32 bit circular number space),
the received packet MUST be discarded. For Meticulous Cryptographic
Authentication, if the Sequence Number lies outside of the range of
bfd.RcvAuthSeq+1 to bfd.RcvAuthSeq+(3*Detect Mult) inclusive (when
treated as an unsigned 32 bit circular number space, the received
packet MUST be discarded.

The device then prepares for generating a digest of the packet.
First of all, the authentication data in the Authentication Value
field needs to be saved somewhere else. Then the Authentication
Value field is set with a pre-specified value (which may be various
in different security algorithms) according the authentication
algorithm indicated in the SA. After this, the device starts
performing the digest generating operations. The work of defining
actual digest generating operations is out of the scope of this
document.

The calculated data is compared with the received authentication data
in the packet and the packet MUST be discarded if the two do not
match. In such a case, an error event SHOULD be logged.

An implementation MAY have a transition mode where it includes
CRYPTO_AUTH or the MET_CRYPTO_AUTH information in the packets but
does not verify this information. This is provided as a transition
aid for networks in the process of migrating to the new CRYPTO_AUTH
and MET_CRYPTO_AUTH based authentication schemes.

3.5. Key Selection for BFD Packet Transmission

In [I-D.ietf-karp-crypto-key-table], a conceptual key database called
"key table" is introduce. A key table is located in the middle of
key management protocols and security protocols so that a security
protocol can derive long-term keys from the key table but does not
have to know the details of key management. This section describes
how the proposed security solution selects long-lived keys from key
tables [I-D.ietf-karp-crypto-key-table].

Assume that a device R1 tries to send a unicast BFD packet from its
interface I1 to the interface R2 of a remote device R2 at time T.
Because the key should be shared by the by both R1 and R2 to protect
the communication between I1 and I2, R1 needs to provide a protocol
("BFD"), an interface identifier (I1) and a peer identifier (R2) into
the key selection function. Any key that satisfies the following
conditions may be selected:

o The Peer field includes the device ID of R2.

o the Protocol field matches "BFD"

o The PeerKeyName field is not "unknown".

o The Interface field includes I1 or "all".

o The Direction field is either "out" or "both".

o SendNotBefore <= current time <= SendNotAfter.

After a set of keys are provided, a BFD implementation should support
selection of keys based on algorithm preference.

Upon R2 receives the BFD packet from R1, R2 provides the protocol
("BFD"), the peer identifier (R1), the key identifier derived from
the incoming packet (L), and the interface (I2) to the key table.
Any key that satisfies the following conditions may be selected:

o The Peer field includes the device ID of R1.

o the Protocol field matches "BFD"

o the LocalKeyName is L

o The Interface field includes I2 or "all".

o The Direction field is either "out" or "both".

o SendNotBefore <= current time <= SendNotAfter.

3.6. Replay Protection using Extended Sequence Numbers

As described in Section 1, if the BFD packets in a session are
transferred with a high frequency, a 32-bit sequence number may reach
its maximum and have to roll back before the session finishes. A
attacker thus can replay the packets intercepted before the sequence
number wrapped without being detected. To address this problem, the
length of the sequence number in the proposed authentication section
has been extended to 64 bits. After the extension, the sequence
number space of a device will not be exhausted within half of a
million years even if the device sends out a BFD packet in every
micro-second. Therefore, the replay attack risks caused by the
limited sequence number space can be largely addressed. However, in
Generic Cryptographic Authentication, the sequence number is only
required to increase occasionally. Therefore, a replayed packet may
be regarded as a legal one until the packet with a larger sequence
number is received. This type of intra-session replay attack cannot
be addressed only by extending the length of sequence numbers.

An anti-replay solution for BFD also needs to consider the scenarios
where a BFD device loses its prior sequence number state (e.g.,
system crash, loss of power). In such cases, a BFD device has to re-
initialize its sequence number. Taking this opportunity, an attacker
may be able to replay the antique packets intercepted in previous
sessions without being detected.

To address this problem, in the proposed solution, the most
significant 32-bit value of the sequence number is used to contain a
boot count, and the remainder 32-bit value is used as an ordinary
32-bit monotonically increasing sequence number. In Generic
Cryptographic Authentication, the remainder 32-bit value is required
to increase occasionally, while in Generic Meticulous Cryptographic
Authentication, the lower order 32-bit sequence number MUST be
incremented for every BFD packet sent by a BFD device. The BFD
implementations are required to retain the boot count in non-volatile
storage for the deployment life the BFD device. The boot count
increases each time when the BFD device loses its prior sequence
number state. The SNMPv3 snmpEngineBoots variable [RFC4222] MAY be
used for this purpose. However, maintaining a separate boot count
solely for BFD sequence numbers has the advantage of decoupling SNMP

re-initialization and BFD re-initialization. Also, in the rare event
that the lower order 32- bit sequence number wraps, the boot count
can be incremented to preserve the strictly increasing property of
the aggregate sequence number. Hence, a separate BFD boot count is
RECOMMENDED.

4. IANA Considerations

This document currently defines a value of 6 to be used to denote
Cryptographic Authentication mechanism for authenticating BFD control
packets and 7 for Meticulous Cryptographic Authentication.

5. Security Considerations

The proposed sequence number extension offers most of the benefits of
of more complicated mechanisms involving challenges. There are,
however, a couple drawbacks to this approach. First, it requires the
BFD implementation to be able to save its boot count in non-volatile
storage. If the non-volatile storage is ever repaired or upgraded
such that the contents are lost or the BFD device is replaced with a
model, the keys MUST be changed to prevent replay attacks. Second,
if a device is taken out of service completely (either intentionally
or due to a persistent failure), the potential exists for re-
establishment of a BFD adjacency by replaying the entire BFD session
establishment. This scenario is however, extremely unlikely and can
be easily avoided. For instance, after recovering from a system
failure, a BFD device has to re-establish BFD sessions. At this
stage, if the device randomly selects its discriminators to identify
new BFD sessions, the possibility of reestablishing a BFD session by
replaying the entire BFD session establishment will be eliminated.
For the implementations in which discriminators are not randomly
selected, this issue can be largely mitigated by integrating the boot
count of the remote BFD router in the generation of the
authentication data for outgoing BFD packets. Of course, this attack
could also be thwarted by changing the relevant manual keys.

There is a transition mode suggested where devices can ignore the
CRYPTO_AUTH or the MET_CRYPTO_AUTH information carried in the
packets. The operator must ensure that this mode is only used when
migrating to the new CRYPTO_AUTH/MET_CRYPTO_AUTH based authentication
scheme as this leaves the device vulnerable to an attack.

6. Acknowledgements

7. References

7.1. Normative References

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

7.2. Informative References

   [Dobb96a]   Dobbertin, H., "Cryptanalysis of MD5 Compress", May 1996.

   [Dobb96b]   Dobbertin, H., "The Status of MD5 After a Recent Attack",
               CryptoBytes", 1996.

   [I-D.ietf-karp-crypto-key-table]
               Housley, R., Polk, T., Hartman, S., and D. Zhang,
               "Database of Long-Lived Symmetric Cryptographic Keys",
               draft-ietf-karp-crypto-key-table-08 (work in progress),
               July 2013.

   [I-D.ietf-karp-design-guide]
               Lebovitz, G. and M. Bhatia, "Keying and Authentication for
               Routing Protocols (KARP) Design Guidelines", draft-ietf-
               karp-design-guide-10 (work in progress), December 2011.

   [MD5-attack]
               Wang, X., Feng, D., Lai, X., and H. Yu, "Collisions for
               Hash Functions MD4, MD5, HAVAL-128 and RIPEMD", August
               2004.

   [RFC1321]   Rivest, R., "The MD5 Message-Digest Algorithm", RFC 1321,
               April 1992.

   [RFC4086]   Eastlake, D., Schiller, J., and S. Crocker, "Randomness
               Requirements for Security", BCP 106, RFC 4086, June 2005.

   [RFC4222]   Choudhury, G., "Prioritized Treatment of Specific OSPF
               Version 2 Packets and Congestion Avoidance", BCP 112, RFC
               4222, October 2005.

   [RFC5880]   Katz, D. and D. Ward, "Bidirectional Forwarding Detection
               (BFD)", RFC 5880, June 2010.

   [SHA-1-attack1]
               Wang, X., Yin, Y., and H. Yu, "Finding Collisions in the
               Full SHA-1", 2005.

   [SHA-1-attack2]
               Wang, X., Yao, A., and F. Yao, "New Collision Search for
               SHA-1", 2005.

Authors' Addresses

   Manav Bhatia
   Alcatel-Lucent
   Bangalore
   India

   Email: manav.bhatia@alcatel-lucent.com

   Vishwas Manral
   Hewlett-Packard Co.
   19111 Pruneridge Ave.
   Cupertino, CA 95014
   USA

   Email: vishwas.manral@hp.com

   Dacheng Zhang
   Huawei
   Beijing
   China

   Email: zhangdacheng@huawei.com