Workgroup: Network Working Group
Internet-Draft:
draft-ietf-bfd-secure-sequence-numbers-10
Updates: 5880 (if approved)
Published: 9 March 2023
Intended Status: Standards Track
Expires: 10 September 2023
Authors: A. Dekok M. Jethanandani
Network RADIUS SARL Kloud Services
S. Agarwal A. Mishra A. Saxena
Cisco Systems, Inc O3b Networks Ciena Corporation

Secure BFD Sequence Numbers

Abstract

This document describes a new BFD Authentication mechanism, Meticulous Keyed ISAAC. This mechanism can be used to authenticate BFD packets with less CPU time cost than using MD5 or SHA1, with the tradeoff of decreased security. This mechanism cannot be used to signal state changes, but it can be used as an authenticated signal to maintain a session in the the "Up" state. This document updates RFC 5880.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

BFD [RFC5880] defines a number of authentication mechanisms, including Simple Password (Section 6.7.2), and various other methods based on MD5 and SHA1 hashes. The benefit of using cryptographic hashes is that they are secure. The downside to cryptographic hashes is that they are expensive and time consuming on resource-constrained hardware.

When BFD packets are unauthenticated, it is possible for an attacker to forge, modify, and/or replay packets on a link. These attacks have a number of side effects. They can cause parties to believe that a link is down, or they can cause parties to believe that the link is up when it is, in fact, down. The goal of this specification is to use a simple method to prevent spoofing of the BFD session being "Up". We therefore define a fast Auth Type method which allows parties securely signal that they are still in the Up state..

This document proposes the use of an Authentication method which provides meticulous keying, but which has less impact on resource constrained systems. The algorithm chosen is a seeded pseudo-random number generator named ISAAC [ISAAC]. ISAAC has been subject to significant cryptanalysis in the past thirty years, and has not yet

been broken. It requires only a few CPU operations per generated 32-bit number, can take a large secret key as a seed, and it has an extremely long period. These properties make it ideal for use in BFD.

2. 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].

3. Meticulous Keyed ISAAC

If the Authentication Present (A) bit is set in the header, and the State (Sta) field equals 3 (Up), and the Authentication Type field contains TB1 (Meticulous Keyed ISAAC), 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  |   Reserved    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Sequence Number                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                             Seed                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                            Auth-Key                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Auth Type

The Authentication Type, which in this case is TB1 (Meticulous Keyed ISAAC). If the State (Sta) field value is not 3 (Up), then Meticulous Keyed ISAAC MUST NOT be used.

Auth Len

The length of the Authentication Section, in bytes. For Meticulous Keyed ISAAC authentication, the length is 16.

Auth Key ID

The authentication key ID in use for this packet. This allows multiple keys to be active simultaneously.

Reserved

This byte MUST be set to zero on transmit, and ignored on receipt.

Sequence Number

The sequence number for this packet. For Meticulous Keyed ISAAC Authentication, this value is incremented for each successive packet transmitted for a session. This provides protection against replay attacks.

Seed

A 32-bit (4 octet) seed which is used in conjunction with the shared key in order to configure and initialize the ISAAC pseudo-random-number-generator (PRNG). It is used to identify and distinguish different "streams" of random numbers which are generated by ISAAC.

Auth-Key

This field carries the 32-bit (4 octet) ISAAC output which is associated with the Sequence Number. The ISAAC PRNG MUST be configured and initialized as given in section TBD, below.

Note that the Auth-Key here does not include any summary or hash of the packet. The packet itself is completely unauthenticated.

When the receiving party receives a BFD packet with an expected sequence number and the correct corresponding ISAAC output in the Auth Key field, it knows that only the authentic sending party could have sent that message. The sending party is therefore "Up", and is the only one who could have sent the message.

While the rest of the contents of the BFD packet are unauthenticated and may be modified by an attacker, the same is true of stronger Auth Types, such as MD5 or SHA1. The Auth Type methods are not designed to prevent such attacks. Instead, they are designed to prevent an attacker from spoofing identities, and an attacker from artificially keeping a session "Up".

4. Operation

BFD requires fast and reasonably secure authentication of messages which are exchanged. Methods using MD5 or SHA1 are CPU intensive, and can negatively impact systems with limited CPU power.

We use ISAAC here as a way to generate an infinite stream of pseudo-random numbers. With Meticulous Keyed ISAAC, these numbers are used as a signal that the sending party is authentic. That is, only the sending party can generate the numbers. Therefore if the receiving party sees a correct number, then only the sending party could have generated that number. The sender is therefore authentic, even if the packet contents are not necessarily trusted.

Note that since the packets are not signed with this authentication type, the Meticulous Keyed ISAAC method MUST NOT be used to signal BFD state changes. For BFD state changes, and a more optimized way to authenticate packets, please refer to BFD Authentication [I-D.ietf-bfd-optimizing-authentication]. Instead, the packets containing Meticulous Keyed ISAAC are only a signal that the sending party is still alive, and that the sending party is authentic. That is, these Auth Type methods must only be used when bfd.SessionState=Up, and the State (Sta) field equals 3 (Up).

4.1. Seeding and Operation of ISAAC

The ISAAC PRNG state is initialized using the 32-bit Seed and the secret key, as defined below.

The origin of the Seed field is discussed later in this document. For now, we note that each time a new Seed is used, the bfd.XmitAuthSeq value MUST be set to zero. The Seed MUST be changed when a BFD session transitions into the "Up" state. In order to prevent continuous rekeying, the Seed MUST NOT be changed while a session is in the "Up" state.

Once the state has been initialized, the standard ISAAC initial mixing function is run. Once this operation has been performed, ISAAC will be able to produce 256 random numbers at near-zero cost. When all 256 numbers are consumed, the ISAAC mixing function is run, which then results in another set of 256 random numbers

ISAAC can be thought of here as producing an infinite stream of numbers, based on a secret key, where the numbers are produced in "pages" of 256 32-bit values. This property of ISAAC allows for essentially zero-cost "seeking" within a page. The expensive operation of mixing is performed only once per 256 packets, which means that most BFD packet exchanges can be fast and efficient.

The Sequence number is used to "seek" within a the stream of 32-bit numbers produced by ISAAC. The sending party increments the Sequence Number on every packet sent, to indicate to the receiving party where it is in the sequence.

The receiving party can then look at the Sequence Number to determine which particular PRNG value is being used in the packet. The Sequence Number thus permits the two parties to synchronise if/ when a packet or packets are lost. Incrementing the Sequence Number for every packet also prevents the re-use of any individual pseudo-random number which was derived from ISAAC.

The Sequence Number can increment without bounds, though it can wrap once it reaches the limit of the 32-bit counter field. ISAAC has a

cycle length of 2^8287, so there is no issue with using more than 2^32 values from it.

The result of the above operation is an infinite series of numbers which are unguessable, and which can be used to authenticate the sending party.

Each system sending BFD packets chooses its own seed, and generates its own sequence of pseudo-random numbers using ISAAC, and place those values into the Auth Key field. Each system receiving BFD packets runs a separate pseudo-random number generator, and verifies that the received packets contain the expected Auth Key.

4.2. Secret Key

For interoperability, the management interface by which the key is configured MUST accept ASCII strings, and SHOULD also allow for the configuration of any arbitrary binary string in hexadecimal form. Other configuration methods MAY be supported.

The secret Key is mixed with the Seed before being used in ISAAC, as described below. If instead ISAAC was initialized without a Seed, then an attacker could pre-compute ISAAC states for many keys, and perform an off-line dictionary attack. The addition of the Seed makes these attacks infeasable.

The Secret Key MUST be at least eight (8) octets in length, and SHOULD NOT be more than 128 octets in length.

As a result, it is believed to be safe to use the same secret Key for the Auth Types defined here, and also for other Auth Types. However, it is RECOMMENDED to use differnet secret Keys for each Auth Type.

4.3. Seeding ISAAC

The value of the Seed field SHOULD be derived from a secure source. Exactly how this can be done is outside of the scope of this document.

The Seed value MUST remain the same for the duration of a BFD session. The Seed value MUST change when the BFD state changes.

The string used to initialize the ISAAC PRNG is taken from the following structure:

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                             Seed                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Your Discriminator                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Secret Key ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Where the "Your Discriminator" field is taken from the BFD packet defined in RFC5880 Section 4.1 [RFC5880]. This field is taken from the respective values used by a sending system. For recieving systems, the field are taken from the received packet. The length of the Secret Key MUST be 1016 octets or less.

The data is padded to 1024 octets using zeroes, and then is processed throught the "randinit()" function of ISAAC. Pseudo-random numbers are then produced by calling the "isaac()" function.

The following figure give Seed and Your-Discriminator as 32-bit hex values, and the Secret Key as an eleven-character string. The subsequent figure shows the first eight Sequence numbers and corresponding Auth Key values which were generated using the above initial values.

```
Seed    0x0bfd5eed
Y-Disc  0x4002d15c
Key     RFC5880June

Sequence Auth Key
00000000 739ba88a
00000001 901e5075
00000002 8e84991c
00000003 93e534cd
00000004 fc213b4b
00000005 f78fc6e6
00000006 3a44db86
00000007 7dda6e6a
```

Note that this construct requires that the "Your Discriminator" field not change during a session. However, it does allow the "My Discriminator" field to change as permitted by RFC5880 Section 6.3 [RFC5880]

This construct provides for 64 bits of entropy, of which 32 bits is controlled by each party in a BFD session. For security, each implemention SHOULD randomize their discrimator fields at the start of a session, as discussed in RFC5880 Section 10 [RFC5880].

There is no way to signal or negotiate Seed changes. The receiving party MUST remember the current Seed value, and then detect if the Seed changes. Note that the Seed value MUST NOT change unless sending party has signalled a BFD state change with a a packet that is authenticated using a more secure Auth Type method.

5. Meticulous Keyed ISAAC Authentication

In this method of authentication, one or more secret keys (with corresponding key IDs) are configured in each system. One of the keys is used to seed the ISAAC PRNG. The output of ISAAC (I) is used to signal that the sender is authentic. To help avoid replay attacks, a sequence number is also carried in each packet. For Meticulous Keyed ISAAC, the sequence number is incremented on every packet.

The receiving system accepts the packet if the key ID matches one of the configured Keys, the Auth-Key derived from the selected Key, Seed, and Sequence Number matches the Auth-Key carried in the packet, and the sequence number is strictly greater than the last sequence number received (modulo wrap at 2^32)

Transmission Using Meticulous Keyed ISAAC Authentication

The Auth Type field MUST be set to TBD1 (Meticulous Keyed ISAAC). The Auth Len field MUST be set to 16. The Auth Key ID field MUST be set to the ID of the current authentication key. The Sequence Number field MUST be set to bfd.XmitAuthSeq.

The Seed field MUST be set to the value of the current seed used for this sequence.

The Auth-Key field MUST be set to the output of ISAAC, which depends on the secret Key, the current Seed, and the Sequence Number.

For Meticulous Keyed ISAAC, bfd.XmitAuthSeq MUST be incremented on each packet, in a circular fashion (when treated as an unsigned 32-bit value). The bfd.XmitAuthSeq MUST NOT be incremented by more than one for a packet.

Receipt using Meticulous Keyed ISAAC Authentication

If the received BFD Control packet does not contain an Authentication Section, or the Auth Type is not correct (TBD2 for Meticulous Keyed ISAAC), then the received packet MUST be discarded.

If the Auth Key ID field does not match the ID of a configured authentication key, the received packet MUST be discarded.

If the Auth Len field is not equal to 16, the packet MUST be discarded.

If the Seed field does not match the current Seed value, the packet MUST be discarded.

If bfd.AuthSeqKnown is 1, examine the Sequence Number field. For Meticulous keyed ISAAC, 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.

Calculate the current expected output of ISAAC, which depends on the secret Key, the current Seed, and the Sequence Number. If the value does not matches the Auth-Key field, then the packet MUST be discarded.

Note that in some cases, calculating the expected output of ISAAC will result in the creation of a new "page" of 256 numbers. This process will irreversible, and will destroy the current "page". As a result, if the generation of a new output will create a new "page", the receiving party MUST save a copy of the entire ISAAC state before proceeding with this calculation. If the outputs match, then the saved copy can be discarded, and the new ISAAC state is used. If the outputs do not match, then the saved copy MUST be restored, and the modified copy discarded, or cached for later use.

6. IANA Considerations

This document asks that IANA allocate a new entry in the "BFD Authentication Types" registry.

Address - TBD1

BFD Authentication Type Name - Meticulous Keyed ISAAC

Reference - this document

Note to RFC Editor: this section may be removed on publication as an RFC.

7. Security Considerations

The security of this proposal depends strongly on the length of the Secret Key, and on its entropy. It is RECOMMENDED that the key be 16 octets in length or more.

The dependency on the Secret Key for security is mitigated through the use of two 32-bit random numbers, with one generated by each

party to a BFD session. An attacker cannot simply perform an off-line brute-force dictionary attack to discover the key. Instead, any analysis has to include the particular 64 bits of entropy used for a particular session. As a result, dictionary attacks are more difficult than they would be if the PRNG generator depended on nothing more than the Secret Key.

The security of this proposal depends strongly on ISAAC. This generator has been analyzed for almost three decades, and has not been broken. Research shows that there are few other CSRNGs which are as simple and as fast as ISAAC. For example, many other generators are based on AES, which is infeasibe for resource constrained systems.

In a keyed algorithm, the key is shared between the two systems. Distribution of this key to all the systems at the same time can be quite a cumbersome task. BFD sessions running a fast rate may require these keys to be refreshed often, which poses a further challenge. Therefore, it is difficult to change the keys during the operation of a BFD session without affecting the stability of the BFD session. Therefore, it is recommended to administratively disable the BFD session before changing the keys.

That is, while the Auth Key ID field provides for the use of multiple keys simultaneously, there is no way for each party to signal which Key IDs are supported.

The Auth Type method defined here allows the BFD end-points to detect a malicious packet, as the calculated hash value will not match the value found in the packet. The behavior of the session, when such a packet is detected, is based on the implementation. A flood of such malicious packets may cause a BFD session to be operationally down.

7.1. Spoofing

When Meticulous Keyed ISAAC is used, it is possible for an attacker who can see the packets to observe a particular Auth Key value, and then copy it to a different packet as a "man-in-the-middle" attack. However, the usefulness of such an attack is limited by the requirements that these packets must not signal state changes in the BFD session, and that the Auth Key changes on every packet.

Performing such an attack would require an attacker to have the following information and capabilities:

This is man-in-the-middle active attack.

The attacker has the contents of a stable packet

The attacker has managed to deduce the ISAAC key and knows which per-packet key is being used.

The attack is therefore limited to keeping the BFD session up when it would otherwise drop.

However, the usual actual attack which we are protecting BFD from is availability. That is, the attacker is trying to shut down then connection when the attacked parties are trying to keep it up. As a result, the attacks here seem to be irrelevant in practice.

7.2. Re-Use of keys

The strength of the Auth-Type methods is significantly different between the strong one like SHA-1 and ISAAC. While ISAAC has had cryptanalysis, and has not been shown to be broken, that analysis is limited. The question then is whether or not it is safe to use the same key for both Auth Type methods (SHA1 and ISAAC), or should we require different keys for each method?

If we recommend different keys, then it is possible for the two keys to be configured differently on each side of a BFD lin. For example. the strong key can be properly provisioned, which allows to the BFD state machine to advance to Up, Then, when we switch to the weaker Auth Type which uses a different key, that key may not match, and the session will immediatly drop.

We believe that the use of the same key is acceptable, as the Auth Type defined for ISAAC depend on 64 bits of random data. The use of this randomness increases the difficulty of breaking the key, and makes off-line dictionary attacks infeasible.

8. Acknowledgements

The authors would like to thank Jeff Haas and Reshad Rahman for their reviews of and suggestions for the document.

9. References

9.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC5880] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD)", RFC 5880, DOI 10.17487/RFC5880, June 2010, <https://www.rfc-editor.org/info/rfc5880>.

9.2. Informative References

[I-D.ietf-bfd-optimizing-authentication]
Jethanandani, M., Mishra, A., Saxena, A., and M. Bhatia, "Optimizing BFD Authentication", Work in Progress, Internet-Draft, draft-ietf-bfd-optimizing-authentication-13, 1 August 2021, <https://datatracker.ietf.org/doc/html/draft-ietf-bfd-optimizing-authentication-13>.

[ISAAC] Jenkins, R. J., "ISAAC", http://www.burtleburtle.net/bob/rand/isaac.html, 1996.

Authors' Addresses

Alan DeKok
Network RADIUS SARL
100 Centrepointe Drive #200
Ottawa ON K2G 6B1
Canada

Email: aland@freeradius.org

Mahesh Jethanandani
Kloud Services

Email: mjethanandani@gmail.com

Sonal Agarwal
Cisco Systems, Inc
170 W. Tasman Drive
San Jose, CA 95070
United States of America

Email: agarwaso@cisco.com
URI: www.cisco.com

Ashesh Mishra
O3b Networks

Email: mishra.ashesh@gmail.com

Ankur Saxena
Ciena Corporation
3939 North First Street
San Jose, CA 95134
United States of America

Email: ankurpsaxena@gmail.com