Network Working Group
Request for Comments: 5386
Category: Standards Track

N. Williams
Sun
M. Richardson
SSW

November 2008

Better-Than-Nothing Security: An Unauthenticated Mode of IPsec

Status of This Memo

Copyright Notice

Abstract

This document specifies how to use the Internet Key Exchange (IKE) protocols, such as IKEv1 and IKEv2, to setup "unauthenticated" security associations (SAs) for use with the IPsec Encapsulating Security Payload (ESP) and the IPsec Authentication Header (AH). No changes to IKEv2 bits-on-the-wire are required, but Peer Authorization Database (PAD) and Security Policy Database (SPD) extensions are specified. Unauthenticated IPsec is herein referred to by its popular acronym, "BTNS" (Better-Than-Nothing Security).

Table of Contents

1. Introduction

Here we describe how to establish unauthenticated IPsec SAs using IKEv2 [RFC4306] and unauthenticated public keys. No new on-the-wire protocol elements are added to IKEv2.

The [RFC4301] processing model is assumed.

This document does not define an opportunistic BTNS mode of IPsec whereby nodes may fall back to unprotected IP when their peers do not support IKEv2, nor does it describe "leap-of-faith" modes or "connection latching".

See [RFC5387] for the applicability and uses of BTNS and definitions of these terms.

This document describes BTNS in terms of IKEv2 and [RFC4301]'s concepts. There is no reason why the same methods cannot be used with IKEv1 [RFC2408], [RFC2409], and [RFC2401]; however, those specifications do not include the PAD concepts, and therefore it may not be possible to implement BTNS on all compliant RFC2401 implementations.

1.1. Conventions Used in This Document

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. BTNS

The IPsec processing model is hereby modified as follows:

- A new ID type is added: 'PUBLICKEY'. IDs of this type have public keys as values. This ID type is not used on the wire.

- PAD entries that match on PUBLICKEY IDs are referred to as "BTNS PAD entries". All other PAD entries are referred to as "non-BTNS PAD entries".

- BTNS PAD entries may match on specific peer PUBLICKEY IDs (or public key fingerprints) or on all peer public keys. The latter is referred to as the "wildcard BTNS PAD entry".

- BTNS PAD entries MUST logically (see below) follow all other PAD entries (the PAD being an ordered list).

- At most one wildcard BTNS PAD entry may appear in the PAD, and, if present, MUST be the last entry in the PAD (see below).

- Any peer that uses an IKEv2 AUTH method involving a digital signature (made with a private key to a public key cryptosystem) may match a BTNS PAD entry, provided that it matches no non-BTNS PAD entries. Suitable AUTH methods as of August 2007 are: RSA Digital Signature (method #1) and DSS Digital Signature (method #3); see [RFC4306], Section 3.8.

- A BTNS-capable implementation of IPsec will first search the PAD for non-BTNS entries matching a peer's ID. If no matching non-BTNS PAD entries are found, then the peer's ID MUST be coerced to be of 'PUBLICKEY' type with the peer's public key as its value. The PAD is then searched again for matching BTNS PAD entries. This ensures that BTNS PAD entries logically follow non-BTNS PAD entries. A single PAD search that preserves these semantics is allowed.

- A peer that matches a BTNS PAD entry is referred to as a "BTNS peer". Such a peer is "authenticated" by verifying the signature in its IKEv2 AUTH payload with the public key from the peer's CERT payload.

- Of course, if no matching PAD entry is found, then the IKE SA is rejected as usual.

- A new flag for SPD entries: 'BTNS_OK'. Traffic to/from peers that match the BTNS PAD entry will match only SPD entries that have the BTNS_OK flag set. The SPD may be searched by address or by ID (of type PUBLICKEY for BTNS peers), as per the IPsec processing model [RFC4301]. Searching by ID in this case requires creation of SPD entries that are bound to public key values. This could be used to build "leap-of-faith" [RFC5387] behavior (see Section 4.2), for example.

Nodes MUST reject IKE_SA proposals from peers that match non-BTNS PAD entries but fail to authenticate properly.

Nodes wishing to be treated as BTNS nodes by their peers MUST include bare public key CERT payloads. Currently only bare RSA public key CERT payloads are defined, which means that BTNS works only with RSA public keys at this time (see "Raw RSA Key" in Section 3.6 of [RFC4306]). Nodes MAY also include any number of certificates that bind the same public key. These certificates do not need to be pre-shared with their peers (e.g., because ephemeral, self-signed). RSA keys for use in BTNS may be generated at any time, but connection latching [ConnLatch] requires that they remain constant between IKEv2 exchanges that are used to establish SAs for latched connections.

To preserve standard IPsec access control semantics:

- BTNS PAD entries MUST logically follow all non-BTNS PAD entries,

- the wildcard BTNS PAD entry MUST be the last entry in the PAD, logically, and

- the wildcard BTNS PAD entry MUST have ID constraints that do not logically overlap those of other PAD entries.

As described above, the logical PAD ordering requirements can easily be implemented by searching the PAD twice at peer authentication time: once using the peer-asserted ID, and if that fails, once using the peer's public key as a PUBLICKEY ID. A single pass implementation that meets this requirement is permitted.

The BTNS entry ID constraint non-overlap requirement can easily be implemented by searching the PAD twice: once when BTNS peers authenticate, and again when BTNS peers negotiate child SAs. In the first pass, the PAD is searched for a matching PAD entry as described above. In the second, it is searched to make sure that BTNS peers' asserted child SA traffic selectors do not conflict with non-BTNS PAD entries. Single pass implementations that preserve these semantics are feasible.

3. Usage Scenarios

In order to explain the above rules, a number of scenarios will be examined. The goal here is to persuade the reader that the above rules are both sufficient and necessary.

This section is informative only.

To explain the scenarios, a reference diagram describing an example network will be used. It is as follows:

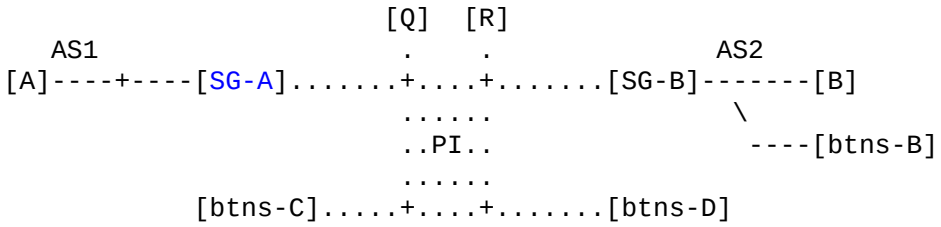

Figure 1: Reference Network Diagram

In this diagram, there are eight systems. Six systems are end-nodes (A, B, C, D, Q, and R). Two are security gateways (SG-A, SG-B) protecting networks on which [A] and [B] reside. Node [Q] is IPsec and BTNS capable. Node [R] is a simple node, with no IPsec or BTNS capability. Nodes [C] and [D] are BTNS capable.

Nodes [C] and [Q] have fixed addresses. Node [D] has a non-fixed address.

We will examine how these various nodes communicate with node [SG-A] and/or how [SG-A] rejects communications with some such nodes. In the first example, we examine [SG-A]'s point of view. In the second example, we look at [Q]'s point of view. In the third example, we look at [C]'s point of view.

PI is the Public Internet ("The Wild").

3.1. Example #1: A Security Gateway

The machine that we will focus on in this example is [SG-A], a firewall device of some kind that we wish to configure to respond to BTNS connections from [C].

[SG-A] has the following PAD and SPD entries:

Rule	Remote ID	Child SA IDs allowed	SPD Search by
1	<B's ID>	<B's network>	by-IP
2	<Q's ID>	<Q's host>	by-IP
3	PUBLICKEY:any	ANY	by-IP

The last entry is the BTNS entry.

Figure 2: [SG-A] PAD Table

Note that [SG-A]'s PAD entry has one and only one wildcard PAD entry: the BTNS catch-all PAD entry as the last entry, as described in Section 2.

<Child SA IDs allowed> and <SPD Search by> are from [RFC4301], Section 4.4.3.

Rule	Local addr	Remote addr	Next Layer Protocol	BTNS ok	Action
1	[A]	[R]	ANY	N/A	BYPASS
2	[A]	[Q]	ANY	no	PROTECT(ESP,tunnel,AES, SHA256)
3	[A]	B-net	ANY	no	PROTECT(ESP,tunnel,AES, SHA256)
4	[A]	ANY	ANY	yes	PROTECT(ESP,transport, integr+conf)

Figure 3: [SG-A] SPD Table

The processing by [SG-A] of SA establishment attempts by various peers is as follows:

- [Q] does not match PAD entry #1 but does match PAD entry #2. PAD processing stops, then the SPD is searched by [Q]'s ID to find entry #2. CHILD SAs are then allowed that have [SG-A]'s and [Q]'s addresses as the end-point addresses.

- [SG-B] matches PAD entry #1. PAD processing stops, then the SPD is searched by [SG-B]'s ID to find SPD entry #3. CHILD SAs are then allowed that have [SG-A]'s address and any addresses from B's network as the end-point addresses.

- [R] does not initiate any IKE SAs; its traffic to [A] is bypassed by SPD entry #1.

- [C] does not match PAD entries #1 or #2 but does match entry #3, the BTNS wildcard PAD entry. The SPD is searched by [C]'s address, and SPD entry #4 is matched. CHILD SAs are then allowed that have [SG-A]'s address and [C]'s address as the end-point addresses, provided that [C]'s address is neither [Q]'s nor any of [B]'s (see Section 2). See the last bullet item below.

- A rogue BTNS node attempting to assert [Q]'s or [B]'s addresses will either match the PAD entries for [Q] or [B] and fail to authenticate as [Q] or [B], in which case they are rejected, or they will match PAD entry #3 but will not be allowed to create CHILD SAs with [Q]'s or [B]'s addresses as traffic selectors.

- A rogue BTNS node attempting to establish an SA whereby the rogue node asserts [C]'s address will succeed at establishing such an SA. Protection for [C] requires additional bindings of [C]'s specific BTNS ID (that is, its public key) to its traffic flows through connection latching and channel binding or through leap-of-faith, none of which are described here.

3.2. Example #2: A Mixed End-System

[Q] is an NFSv4 server.

[Q] is a native IPsec implementation, and its NFSv4 implementation is IPsec-aware.

[Q] wants to protect all traffic with [A]. [Q] also wants to protect NFSv4 traffic with all peers. Its PAD and SPD are configured as follows:

Rule	Remote ID	Child SA IDs allowed	SPD Search by
1	<[A]'s ID>	<[A]'s address>	by-IP
2	PUBLICKEY:any	ANY	by-IP

The last entry is the BTNS entry.

Figure 4: [Q] PAD Table

Rule	Local addr	Remote addr	Next Layer Protocol	BTNS ok	Action
1	[Q]	[A]	ANY	no	PROTECT(ESP,tunnel,AES, SHA256)
2	[Q] with port 2049	ANY	ANY	yes	PROTECT(ESP,transport, integr+conf)

Figure 5: [Q] SPD Table

The same analysis shown above in Section 3.1 applies here with respect to [SG-A], [C], and rogue peers. The second SPD entry permits any BTNS-capable node to negotiate a port-specific SA to port 2049, the port on which NFSv4 runs. Additionally, [SG-B] is treated as a BTNS peer as it is not known to [Q], and therefore any host behind [SG-B] can access the NFSv4 service on [Q]. As [Q] has no formal relationship with [SG-B], rogues can impersonate [B] (i.e., assert [B]'s addresses).

3.3. Example #3: A BTNS-Only System

[C] supports only BTNS and wants to use BTNS to protect NFSv4 traffic. Its PAD and SPD are configured as follows:

Rule	Remote ID	Child SA IDs allowed	SPD Search by
1	PUBLICKEY:any	ANY	by-IP

The last (and only) entry is the BTNS entry.

Figure 6: [Q] PAD Table

Rule	Local addr	Remote addr	Next Layer Protocol	BTNS ok	Action
1	[C] with port 2049	ANY	ANY	yes	PROTECT(ESP,transport, integr+conf)
2	[C]	ANY	ANY	N/A	BYPASS

Figure 7: [SG-A] SPD Table

The analysis from Section 3.1 applies as follows:

- Communication with [Q] on port 2049 matches SPD entry number 1. This causes [C] to initiate an IKEv2 exchange with [Q]. The PAD entry on [C] causes it to not care what identity [Q] asserts. Further authentication (and channel binding) could occur within the NFSv4 protocol.

- Communication with [A], [B], or any other internet machine (including [Q]), occurs in the clear, so long as it is not on port 2049.

- All analysis about rogue BTNS nodes applies, but they can only assert SAs for port 2049.

3.4. Miscellaneous Comments

If [SG-A] were not BTNS capable, then it would not have PAD and SPD entries #3 and #4, respectively, in example #1. Then [C] would be rejected as usual under the standard IPsec model [RFC4301].

Similarly, if [Q] were not BTNS capable, then it would not have PAD and SPD entries #2 in example #2. Then [C] would be rejected as usual under the standard IPsec model [RFC4301].

4. Security Considerations

Unauthenticated security association negotiation is subject to man-in-the-middle (MITM) attacks and should be used with care. Where security infrastructures are lacking, this may indeed be better than nothing.

Use with applications that bind authentication at higher network layers to secure channels at lower layers may provide one secure way to use unauthenticated IPsec, but this is not specified herein.

The BTNS PAD entry must be last and its child SA ID constraints must be non-overlapping with any other PAD entry, as described in Section 2. This will ensure that no BTNS peer can impersonate another IPsec non-BTNS peer.

4.1. Connection Latching and Channel Binding

BTNS is subject to MITM attacks. One way to protect against MITM attacks subsequent to initial communications is to use "connection latching" [ConnLatch]. In connection latching, upper layer protocols (ULPs) cooperate with IPsec to bind discrete packet flows to

sequences of similar SAs. Connection latching requires native IPsec implementations.

MITMs can be detected by using application-layer authentication frameworks and/or mechanisms, such as the GSS-API [RFC2743], with channel binding [RFC5056]. IPsec "channels" are nothing other than latched connections.

4.2. Leap-of-Faith (LoF) for BTNS

"Leap of faith" is the term generally used when a user accepts the assertion that a given key identifies a peer on the first communication (despite a lack of strong evidence for that assertion), and then remembers this association for future communications. Specifically this is a common mode of operation for Secure Shell [RFC4251] clients. When a server is encountered for the first time, the Secure Shell client may ask the user whether to accept the server's public key. If so, the client records the server's name (as given by the user) and public key in a database.

Leap of faith can work in a similar way for BTNS nodes, but it is currently still being designed and specified by the IETF BTNS WG.

5. Acknowledgements

Thanks to the following reviewer: Stephen Kent.

6. References

6.1. Normative References

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

[RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, December 2005.

6.2. Informative References

[ConnLatch] Williams, N., "IPsec Channels: Connection Latching", Work in Progress, April 2008.

[RFC2401] Kent, S. and R. Atkinson, "Security Architecture for the Internet Protocol", RFC 2401, November 1998.

[RFC2408] Maughan, D., Schneider, M., and M. Schertler, "Internet Security Association and Key Management Protocol (ISAKMP)", RFC 2408, November 1998.

[RFC2409] Harkins, D. and D. Carrel, "The Internet Key Exchange (IKE)", RFC 2409, November 1998.

[RFC2743] Linn, J., "Generic Security Service Application Program Interface Version 2, Update 1", RFC 2743, January 2000.

[RFC4251] Ylonen, T. and C. Lonvick, "The Secure Shell (SSH) Protocol Architecture", RFC 4251, January 2006.

[RFC4306] Kaufman, C., "Internet Key Exchange (IKEv2) Protocol", RFC 4306, December 2005.

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

[RFC5387] Touch, J., Black, D., and Y. Wang, "Problem and Applicability Statement for Better-Than-Nothing Security (BTNS)", RFC 5387, November 2008.

Authors' Addresses

Nicolas Williams
Sun Microsystems
5300 Riata Trace Ct
Austin, TX 78727
US

EMail: Nicolas.Williams@sun.com

Michael C. Richardson
Sandelman Software Works
470 Dawson Avenue
Ottawa, ON K1Z 5V7
CA

EMail: mcr@sandelman.ottawa.on.ca
URI: http://www.sandelman.ottawa.on.ca/