Network Working Group
Request for Comments: 5387
Category: Informational

J. Touch
USC/ISI
D. Black
EMC
Y. Wang
Microsoft

November 2008

Problem and Applicability Statement for Better-Than-Nothing Security (BTNS)

Status of This Memo

Copyright Notice

Abstract

The Internet network security protocol suite, IPsec, requires authentication, usually of network-layer entities, to enable access control and provide security services. This authentication can be based on mechanisms such as pre-shared symmetric keys, certificates with associated asymmetric keys, or the use of Kerberos (via Kerberized Internet Negotiation of Keys (KINK)). The need to deploy authentication information and its associated identities can be a significant obstacle to the use of IPsec.

This document explains the rationale for extending the Internet network security protocol suite to enable use of IPsec security services without authentication. These extensions are intended to protect communication, providing "better-than-nothing security" (BTNS). The extensions may be used on their own (this use is called Stand-Alone BTNS, or SAB) or may be used to provide network-layer security that can be authenticated by higher layers in the protocol stack (this use is called Channel-Bound BTNS, or CBB). The document also explains situations for which use of SAB and/or CBB extensions are applicable.

Table of Contents

1. Introduction

Network security is provided by a variety of protocols at different layers in the stack. At the network layer, the IPsec protocol suite (consisting of IKE (Internet Key Exchange protocol), ESP (Encapsulating Security Payload), and AH (Authentication Header)) is used to secure IP traffic. IPsec, including IKE, offers high levels of security that provide protection from a wide array of possible threats, but authentication is required [5][7][8]. In turn, authentication requires deployment of authentication identities and credentials, which can be an obstacle to IPsec usage. This document discusses this dependency and introduces "Better-Than-Nothing Security" (BTNS) as one solution, whose goal is to provide a generally useful means of applying IPsec security services without requiring network-layer authentication.

1.1. Authentication

There are two primary architectural approaches to authentication: employing out-of-band communications and using pre-deployed information. Out-of-band authentication can be done via a trusted third party, via a separate communication channel to the peer, or via the same channel as the communications to be secured but at a higher layer. Out-of-band authentication requires mechanisms and interfaces to bind the authenticated identities to the secure communication channels, and is out of scope for this document (although it may be possible to extend the channel binding mode of BTNS to work with such mechanisms). Pre-deployed information includes identities, pre-shared secrets, and credentials that have been authenticated by trusted authorities (e.g., a certificate and its corresponding private key).

This form of authentication often requires manual deployment and coordination among communicating peers. Furthermore, obtaining and deploying credentials such as certificates signed by certification authorities (CA) involves additional protocol and administrative actions that may incur significant time and effort to perform.

These factors increase the work required to use IKE with IPsec for peer authentication. Consequently, some users and applications do not use IPsec to protect traffic at the network layer, but rely instead on higher-layer security protocols (e.g., TLS [4]) or operate without any security. As Section 2.2.1 describes, higher-layer security protocols may not be enough to protect against some network-layer attacks.

To improve the situation, one could either reduce the hurdles to obtain and configure authentication information or remove the requirement for authentication in IPsec. The latter approach is the core idea of BTNS, which provides anonymous (unauthenticated) keying for IPsec to create security associations (SAs) with peers that do not possess requisite authentication credentials. This requires extensions to the IPsec architecture. As the new BTNS modes for IPsec relax the authentication requirement, the impacts, tradeoffs, and risks must be thoroughly understood before applying BTNS to any communications. More specifically, this document addresses the issues of whether and when network-layer authentication can be omitted, the risks of using BTNS, and finally, the impacts to the existing IPsec architecture.

BTNS employs a weaker notion of authenticated identity by comparison to most authentication protocols; this weaker notion is bootstrapped from the security association itself. This notion, called "continuity of association", doesn’t mean "Bill Smith" or "owner of shared secret X2YQ", but means "the entity with which I have been communicating on connection #23". Continuity of association is only invariant within a single SA; it is not invariant across SAs, and hence can only be used to provide protection during the lifetime of an SA. This is a core notion used by BTNS, particularly in the absence of higher-layer authentication. Continuity of association can be viewed as a form of authentication in which an identity is not authenticated across separate associations or out-of-band, but does not change during the lifetime of the SA.

1.2. IPsec Channels and Channel Binding

When IPsec security services are used by higher-layer protocols, it is important to bind those services to higher-layer protocol sessions in order to ensure that the security services are consistently applied to the higher-layer traffic involved. The result of this binding is an "IPsec channel", and the act of creating an IPsec channel is an instance of channel binding. Channel binding is discussed in RFC 5056 [27] and in an associated connection latching document [26]. This subsection summarizes the portions of these documents that are essential to understanding certain aspects of BTNS.

A secure channel is a packet, datagram, octet stream connection, or sequence of connections between two endpoints that affords cryptographic integrity and, optionally, confidentiality to data exchanged over it [27]. Applying this concept to IPsec, an "IPsec channel" is a packet flow associated with a higher-layer protocol session, such as a TCP connection, where all the packets are protected by IPsec SAs such that:

- o the peer’s identity is the same for the lifetime of the packet flow, and

- o the quality of IPsec protection used for the packet flow’s individual packets is the same for all of them for the lifetime of the packet flow [26].

The endpoints of an IPsec channel are the higher-layer protocol endpoints, which are beyond the endpoints of the IPsec SAs involved. This creates a need to bind each IPsec SA to the higher-layer protocol session and its endpoints. Failure to do this binding creates vulnerabilities to man-in-the-middle (MITM) attacks, where what appears to be a single IPsec SA for the higher-layer protocol traffic is actually two separate SAs concatenated by the attacker acting as a traffic-forwarding proxy.

The combination of connection latching [26] with channel binding [27] creates IPsec channels and binds IPsec SAs to higher-layer protocols. Connection latching creates an IPsec channel by associating IPsec SAs to higher-layer protocol sessions, and channel binding enables a higher-layer protocol to bind its authentication to the IPsec SAs. Caching of this "latch" across higher-layer protocol sessions is necessary to counter inter-session spoofing attacks, and the channel binding authentication should be performed on each higher-layer protocol session. Connection latching and channel binding are useful not only for BTNS but also for IPsec SAs whose peers are fully authenticated by IKE during creation of the SA.

Channel binding for IPsec is based on information obtained from the SA creation process that uniquely identifies an SA pair. Channel binding can be accomplished by adding this identifying information to higher-layer authentication mechanisms based on one-way hashes, key exchanges, or (public key) cryptographic signatures; in all three cases, the resulting higher-layer authentication resists man-in-the-middle attacks on SA creation. When each IKE peer uses a public-private key pair for IKE authentication to create an SA pair, the pair of public keys used (one for each peer) suffices for channel binding; strong incorporation of this information into higher-layer authentication causes that higher-layer authentication to fail when an MITM attacker has concatenated separate SAs by acting as a traffic-forwarding proxy.

1.3. BTNS Methods

There are two classes of scenarios in which BTNS may be used to apply IPsec services without network-layer authentication:

1. Protection of traffic for a higher-layer protocol that does not use authentication. The resulting protection is "better than nothing" because once an unauthenticated SA is successfully created without an MITM, that SA's IPsec security services resist subsequent MITM attacks even though the absence of authentication allows the initial creation of the BTNS-based security association (SA) to be subverted by an MITM. This method of using BTNS is called Stand-Alone BTNS (SAB) because it does not rely on any security services outside of IPsec.

2. Protection of traffic generated by a higher-layer protocol that uses authentication. The "better-than-nothing" protection in this case relies on the strength of the higher-layer protocol's authentication and the channel binding of that authentication with the BTNS-based SAs. The level of protection may be comparable to the level afforded by the use of network-layer IKE authentication when the higher-layer protocol uses strong authentication and strong channel binding is employed to associate the BTNS-based SA with that higher-layer authentication. This method of using BTNS is called Channel-Bound BTNS (CBB) when the combination of the higher-layer authentication and channel binding is sufficient to detect an MITM attack on creation of a BTNS-based SA.

It is possible to combine IKE authentication for one end of an SA pair with BTNS's absence of network-layer authentication for the other end. The resulting asymmetric authentication creates asymmetric modes of BTNS that are discussed further in Section 3.2 below.

1.4. BTNS Scope

The scope of BTNS is to provide a generally useful means of applying IPsec security services that does not require network-level authentication credentials. The following areas are outside this scope of BTNS and hence are not discussed further in this document:

1. Use of security frameworks other than IPsec to provide security services for higher-layer protocols. There are a variety of security service frameworks other than IPsec, such as TLS [4], Simple Authentication and Security Layer (SASL) [11], and Generic Security Service Application Program Interface (GSS-API) [10], as well as a variety of non-IPsec security mechanisms, such as TCP

MD5 [6], that are described in other documents. BTNS is based on IPsec by design; it will not always be the most appropriate solution.

2. Use of the Extensible Authentication Protocol (EAP) for IKE authentication. Section 1.3 of RFC 3748 clearly restricts EAP's applicability to network access protocols [1]:

   > "EAP was designed for use in network access authentication, where IP layer connectivity may not be available. Use of EAP for other purposes, such as bulk data transport, is NOT RECOMMENDED."

   Hence, EAP authentication for IKE is only applicable to situations where IKE is being used to establish network access (e.g., create a Virtual Private Network (VPN) connection). In contrast, the BTNS goal of general applicability encompasses many areas other than network access and specifically includes protocols that transfer large amounts of data, such as iSCSI [19] and NFSv4 [21].

3. Manual keying is not considered for BTNS because manual keying is unsafe for protocols that transfer large amounts of data (e.g., RFC 3723 forbids use of manual keying with the IP Storage protocols, including iSCSI, for this reason [2]).

1.5. Structure of This Document

The next section discusses the motivations for BTNS, primarily based on the implications of IKE's requirements for network-layer authentication. Section 3 provides a high level overview of BTNS, both SAB and CBB. Section 3 also includes descriptions of the security services offered and the BTNS modes of operation (based on combinations of SAB, CBB, and/or IKE authentication). Section 4 explores the applicability of all of the modes of BTNS. This is followed by a discussion of the risks and other security considerations in Section 5. Section 6 briefly describes other related efforts.

2. Problem Statement

This section describes the problems that motivated the development of BTNS. The primary concern is that IPsec is not widely utilized despite rigorous development effort and emphasis on network security by users and organizations. There are also differing viewpoints on which layer is best for securing network communications and how security protocols at different layers should interact. The following discussion roughly categorizes these issues by layers: network layer and higher layers.

2.1. Network Layer

At the network layer, one of the hurdles is to satisfy the authentication requirements of IPsec and IKE. This section discusses some drawbacks of network-layer authentication and the results of these requirements.

2.1.1. Authentication Identities

Current IPsec authentication supports several types of identities in the Peer Authorization Database (PAD): IPv4 addresses, IPv6 addresses, DNS names, Distinguished Names, RFC 822 email addresses, and Key IDs [8]. All require either certificates or pre-shared secrets to authenticate. The identities supported by the PAD can be roughly categorized as network-layer identifiers or other identifiers.

The first three types of identifiers -- IPv4 addresses, IPv6 addresses and DNS names -- are network-layer identifiers. The main deficiency of IP addresses as identifiers is that they often do not consistently represent the same physical systems due to the increasing use of dynamic address assignments (DHCP) and system mobility. The use of DNS names is also affected because the name to address mapping is not always up to date as a result. Stale mapping information can cause inconsistencies between the IP address recorded in the DNS for a named system and the actual IP address of that system, leading to problems if the DNS is used to cross-check the IP address from which a DNS name was presented as an identifier. DNS names are also not always under the control of the endpoint owner.

There are two main drawbacks with the other, non-network-layer identifiers defined for the PAD. The PAD functionality can be overly restrictive because there are other forms of identifiers not covered by the PAD specification (EAP does not loosen these restrictions in general; see Section 1.4). Use of any non-network-layer identifiers for IPsec authentication may result in multiple authentications for the same or different identifiers at different layers, creating a need to associate authentications and new error cases (e.g., one of two authentications for the same identifier fails). These issues are also related to channel binding and are further discussed later in this document.

2.1.2. Authentication Methods

As described earlier, certificates and pre-shared secrets are the only methods of authentication accepted by current IPsec and IKE specifications. Pre-shared secrets require manual configuration and out-of-band communications. The verification process for certificates is cumbersome, plus there are administrative and potential monetary costs in obtaining certificates. These factors are among the possible reasons why IPsec is not widely used outside of environments with the highest security requirements.

2.1.3. Current IPsec Limits on Unauthenticated Peers

Pre-configuration of Security Policy Database (SPD) "bypass" entries to enable communication with unauthenticated peers only works if the peer IP addresses are known in advance. The lack of unauthenticated IPsec modes often prevents secure communications at the network layer with unauthenticated or unknown peers, even when they are subsequently authenticated in a higher-layer protocol or application. The lack of a channel binding API between IPsec and higher-layer protocols may further force such communications to completely bypass IPsec, leaving the network layer of such communications unprotected.

2.2. Higher-Layer Issues

For higher layers, the next subsection focuses on whether IPsec is necessary if transport layer security is already in use. The use of IPsec in the presence of transport security provides further motivation for reducing the administrative burdens of using IPsec. This is followed by a discussion of the implications of using authentication at both the network layer and a higher layer for the same connection.

2.2.1. Transport Protection from Packet Spoofing

Consider the case of transport protocols. Increases in network performance and the use of long-lived connections have resulted in increased vulnerability of connection-oriented transport protocols to certain forms of attacks. TCP, like many other protocols, is susceptible to off-path third-party attacks, such as injection of a TCP RST [24]. The Internet lacks comprehensive ingress filtering to discard such spoofed traffic before it can cause damage. These attacks can affect BGP sessions between core Internet routers, and are thus of significant concern [3][12]. As a result, a number of proposed solutions have been developed, most of which are at the transport layer.

Some of these solutions augment the transport protocol by improving its own security, e.g., TCP MD5 [6]. Others modify the core TCP processing rules to make it harder for off-path attackers to inject meaningful packets either during the initial handshake (e.g., SYN cookies) or after a connection is established (e.g., TCPsecure) [15][23]. Some of these approaches are new to TCP, but have already been incorporated into other transport protocols (e.g., Stream Control Transmission Protocol (SCTP) [22]) or intermediate (so-called layer 3.5) protocols (e.g., Host Identity Protocol (HIP) [14]).

TCP MD5 and its potential successor, TCP Auth [25], are based on authentication; TCP-specific modifications that lack authentication are, at best, temporary patches to the ubiquitous vulnerability to spoofing attacks. The obvious solution to spoofing is end-to-end validation of the traffic, either at the transport layer or the network layer. The IPsec suite already provides authentication of a network-layer packet and its contents, but the costs of an authentication infrastructure required for the use of IPsec can be prohibitive. Similarly, TCP MD5 requires pre-shared keys, which can likewise be prohibitive. TCP Auth is currently under development, and may include a BTNS-like mode.

Protecting systems from spoofed packets is ultimately an issue of authentication, ensuring that a receiver's interpretation of the source of a packet is accurate. Authentication validates the identity of the source of the packet. The current IPsec suite assumes that identity is validated either by a trusted third party -- e.g., a certification authority -- or by a pre-deployed shared secret. Such an identity is unique and invariant across associations (pair-wise security configuration), and can be used to reject packets that are not authentic.

With regard to BGP in particular, it has been understood that the use of appropriate network- or transport-layer authentication is the preferred protection from TCP spoofing attacks [3]. Authentication at one router by itself does not provide overall BGP security because that router remains at the mercy of all routers it peers with, since it depends on them to also support authentication [25]. The reality is that few Internet routers are configured to support authentication at all, and the result is the use of unsecured TCP for sending BGP packets. BTNS allows an individual router to relax the need for authentication in order to enable the use of protected sessions that are not authenticated. The latter is "better than nothing" in cases where "nothing" is the alternative. Although the routing community has chosen solutions other than BTNS for protection of BGP's TCP connections (e.g., TCP MD5), the discussion of BGP remains in this document because it was a motivation for the development of BTNS.

2.2.2. Authentication at Multiple Layers

Some existing protocols used on the Internet provide authentication above the network and transport layers but rely on the IPsec suite for packet-by-packet cryptographic integrity and confidentiality services. Examples of such protocols include iSCSI [19] and the remote direct data placement (RDDP) protocols [16][20]. With the current IPsec suite, the result is two authentication operations: one at the IPsec layer using an identity for IKE and an associated secret or key, and another by the higher-layer protocol using a higher-layer identity and secret or key. With the current IPsec specifications, this redundant authentication is necessary because the identity and key formats differ between IPsec and the higher-layer protocol and/or because there is no standard interface to pass authentication results from IPsec up to the higher layer. End-node software is then responsible for ensuring that the identities used for these two authentication operations are consistent in some fashion; determining whether these identities are consistent is an authorization policy decision.

Failure of the end-node software to enforce appropriate consistency across authentication operations at different layers creates man-in-the-middle attack opportunities at the network layer. An attacker may exploit this omission by interposing as a proxy; rather than impersonate the attacked endpoints, the attacker need only authenticate with identities that are acceptable to the attacked endpoints. The resulting success enables the attacker to obtain full access to the higher-layer traffic by passing the higher-layer authentication operation through without modification. In the complete absence of consistency checks on the identities used at different layers, higher-layer traffic may be accessible to any entity that can successfully authenticate at the network layer.

In principle, a single authentication operation should suffice to protect the higher-layer traffic, removing the need for:

- the second authentication operation,

- configuration and management of the identities and secrets or keys for the second authentication (even if the identities and secrets or keys are the same, the two authentication operations may employ different repositories for identities, secrets, and keys), and

- determining in some fashion that the two authenticated identities are consistent. As noted above, there are significant potential MITM vulnerabilities if this is not done.

IPsec may not always be present for these higher-layer protocols, and even when present, may not always be used. Hence, if there is a choice, the higher-layer protocol authentication is preferable as it will always be available for use, independent of IPsec.

A "better-than-nothing" security approach to IPsec can address this problem by setting up an IPsec security association without an authentication, and then using an extended form of the higher-layer authentication to establish that the higher-layer protocol session is protected by a single IPsec SA. This counters man-in-the-middle (MITM) attacks on BTNS IPsec session establishment by terminating the higher-layer session via an authentication failure when such an attack occurs. The result is that a single authentication operation validates not only the higher-layer peer's identity but also continuity of the security association to that peer. This higher-layer check for a single IPsec SA is referred in this document as "channel binding", thus the name Channel-Bound BTNS (CBB) [27].

3. BTNS Overview and Threat Models

This section provides an overview of BTNS and the IPsec security services that are offered when BTNS is used. It also describes the multiple operating modes of BTNS.

3.1. BTNS Overview

This is an overview of what is needed in IPsec to enable BTNS. The detailed specifications of the extensions are addressed by the relevant protocol specifications.

The main update to IPsec is adding extensions to security policy that permit secure communications with unauthenticated peers. These extensions are necessary for both IPsec and IKE. For IPsec, the first extension applies to the PAD, which specifies the forms of authentication allowed for each IKE peer. In addition to existing forms of authentication, such as X.509 certificates and pre-shared secrets, the extension adds an unauthenticated category in which the public key presented by the peer serves as its identity (and is authenticated by the peer demonstrating knowledge of the corresponding private key) [28]. The second extension is that a flag is added to each SPD entry to indicate whether BTNS lack of authentication is acceptable for that SPD entry.

The changes to enable channel binding between IPsec and higher-layer protocols or applications are more complex than the policy extensions above. They require specifying APIs and interactions between IPsec and higher-layer protocols. This document assumes such provisions will be developed, but does not address their details.

3.2. BTNS and IPsec Security Services

The changes and extensions of BTNS primarily affect IPsec policy as described above. Other parts of IPsec and IKE specifications are unchanged. BTNS does not require a separate IPsec implementation, as BTNS can be integrated with any IPsec implementation in a system. The scope of BTNS functionality applies only to the SAs matching the policies that explicitly specify or enable BTNS modes in the PAD and for which the corresponding SPD entries allow BTNS. All other non-BTNS policy entries, including entries in the SPD and the PAD, and non-BTNS SAs are not affected by BTNS.

In principle, the result of removing the requirement that all SAs be authenticated is that BTNS can establish secure IPsec connections in a fashion similar to fully authenticated IKE, but BTNS cannot verify or authenticate the peer identities of these SAs. The following is a list of security services offered by the IPsec protocol suite with notes that address the differences created by the addition of BTNS.

1. Access Control

   BTNS extends IPsec's access control services to allow unauthenticated connections. These extensions are integrated with the IPsec PAD and SPD in a fashion that does not affect the access controls associated with entries that do not use the BTNS extensions. For Channel-Bound BTNS, the authentication that applies to the SA is performed at a higher layer in a fashion that links higher-layer access control policy to IPsec's network-layer access control mechanisms.

2. Data Origin Authentication

   Stand-Alone BTNS weakens data origin authentication to continuity of association, namely the assurance that traffic on an SA continues to originate from the same unauthenticated source.

   Channel-Bound BTNS relies on higher-layer authentication to provide data origin authentication of protected network traffic.

3. Connectionless Integrity

4. Anti-Replay Protection

5. Confidentiality

6. (Limited) Traffic Flow Confidentiality

   For the security services offered by IPsec that are listed in items 3 through 6, it is possible to establish secure IPsec connections with rogue peers via BTNS because authentication is not required. On the other hand, once a secure connection is established, the communication is protected by these security services in the same fashion as a connection established by conventional IPsec means.

3.3. BTNS and IPsec Modes

The previous sections have described two ways of using BTNS: Stand-Alone (SAB) and Channel-Bound (CBB). Both of these can also be used either symmetrically, where neither party authenticates at the network layer, or asymmetrically, where only one party does not authenticate at the network layer. There are a number of cases to consider, based on combinations of the endpoint security capabilities of SAB, CBB, and conventional IKE authentication of an identity (denoted as AUTH below). The following tables show all of the combinations based on the capabilities of the two security endpoints:

```
     | AUTH  |  SAB  |                | CB-AUTH |   CBB   |
-----+-------+-------+         -------+---------+---------+
     |       |       |                |         |         |
AUTH | AUTH  | A-SAB |         CB-AUTH| CB-AUTH |  A-CBB  |
     |       |       |                |         |         |
-----+-------+-------+         -------+---------+---------+
     |       |       |                |         |         |
SAB  | A-SAB | S-SAB |           CBB  |  A-CBB  |  S-CBB  |
     |       |       |                |         |         |
-----+-------+-------+         -------+---------+---------+

  No Channel Binding                With Channel Binding
```

There are six operating modes that result from the combinations. The first three modes consist of network-layer authentication schemes used without channel binding to higher-layer authentication:

1. AUTH: both parties provide and authenticate conventional, IKE-supported identities.

2. Symmetric SAB (S-SAB): neither party authenticates with a conventional, IKE-supported identity.

3. Asymmetric SAB (A-SAB): one party does not authenticate with a conventional, IKE-supported identity, but the other side does authenticate with such an identity.

The following three modes combine the network-layer behaviors with channel binding to higher-layer authentication credentials:

4. CB-AUTH: channel binding is used and both parties authenticate with conventional, IKE-supported identities.

5. Symmetric CBB (S-CBB): neither party authenticates with a conventional, IKE-supported identity, but channel binding is used to bind the SAs to higher-layer authentication operations.

6. Asymmetric CBB (A-CBB): asymmetric SAB (A-SAB) used with channel binding; at the network layer, one party does not authenticate with a conventional, IKE-supported identity, but the other party does authenticate with such an identity. Channel binding is used to bind the SA to higher-layer authentication operations.

There are three security mechanisms involved in BTNS with channel binding:

1. BTNS and IPsec at the network layer,

2. higher-layer authentication, and

3. the connection latching plus channel binding mechanisms that bind the higher-layer authentication credentials with the secure IPsec channel.

Authentication at both the network and higher layers can be either bidirectional (both peers are authenticated) or unidirectional (one of the two peers does not authenticate). In contrast, when channel binding is used, it must be applied at both ends of the communication to prevent MITM attacks. Existing channel binding mechanisms and APIs for this purpose (e.g., as defined in GSS-API [10]) mandate the exchange and verification of the channel binding values at both ends to ensure that correct, non-spoofed channel characteristics are bound to the higher-layer authentication.

Note: When any Stand-Alone BTNS (SAB) or Channel-Bound BTNS (CBB) is used without being qualified as symmetric or asymmetric, the symmetric mode is the intended default meaning.

4. Applicability Statement

BTNS is intended for services open to the public but for which protected associations are desired, and for services that can be authenticated at higher layers in the protocol stack. BTNS can also provide some level of protection for private services when the alternative BTNS is no protection at all.

BTNS uses the IPsec protocol suite, and therefore should not be used in situations where IPsec and specifically IKE are unsuitable. IPsec and IKE incur additional computation overhead, and IKE further requires message exchanges that incur round-trip latency to setup security associations. These may be undesirable in environments with limited computational resources and/or high communication latencies.

This section provides an overview of the types of applications suitable for various modes of BTNS. The next two sections describe the overall benefits and vulnerabilities, followed by the applicability analysis for each BTNS mode. The applicability statement covers only the four BTNS-specific modes; the AUTH and CB-AUTH modes are out of scope for this discussion.

4.1. Benefits

BTNS protects security associations after they are established by reducing vulnerability to attacks from parties that are not participants in the association. BTNS-based SAs protect network and transport layers without requiring network-layer authentication. BTNS can be deployed without pre-deployment of authentication material for IPsec or pre-shared information and can protect all transport layer protocols using a common mechanism.

BTNS also helps protect systems from low-effort attacks on higher-layer sessions or connections that disrupt valuable services or resources. BTNS raises the level of effort for many types of network- and transport-layer attacks. Simple transport layer packet attacks are rejected because the malicious packet or packets are not part of an IPsec SA. The attacker is instead forced to establish an unauthenticated IPsec SA and a transport connection for SAB, requiring the attacker to perform as much work as a host engaging in the higher-layer communication. SAB thus raises the effort for a DDoS (Distributed Denial of Service) attack to that of emulating a flash crowd. For open services, there may be no way to distinguish such a DDoS attack from an actual flash crowd.

BTNS also allows individual security associations to be established for protection of higher-layer traffic without requiring pre-deployed authentication credentials.

4.2. Vulnerabilities

BTNS removes the requirement that every IPsec SA be authenticated. Hosts connecting to BTNS hosts are vulnerable to communicating with a masquerader throughout the association for SAB, or until higher layers provide additional authentication for CBB. As a result, authentication data (e.g., passwords) sent to a masquerading peer could be disclosed to an attacker. This is a deliberate design tradeoff; in BTNS, network- and transport-layer access is no longer controlled by the identity presented by the other host, opening hosts to potential masquerading and flash crowd attacks. Conversely, BTNS can secure connections to hosts that are unable to authenticate at the network layer, so the network and transport layers are more protected than can be achieved via higher-layer authentication alone.

Lacking network-layer authentication information, other means must be used to provide access control for local resources. Traffic selectors for the BTNS SPD entries can be used to limit which interfaces, address ranges, and port ranges can access BTNS-enabled services. Rate limiting can further restrict resource usage. For SAB, these protections need to be considered throughout associations, whereas for CBB they need be present only until higher-layer protocols provide the missing authentication. CBB also relies on the effectiveness of the binding of higher-layer authentication to the BTNS network association.

4.3. Stand-Alone BTNS (SAB)

SAB is intended for applications that are unable to use IKE-compatible authentication credentials and do not employ higher-layer authentication or other security protection. SAB is also suitable when the identities of either party are not important or are deliberately omitted, but IPsec security services are desired (see Section 3.2). SAB is particularly applicable to long-lived connections or sessions for which assurance that the entity at the other end of the connection has not changed may be a good enough substitute for the lack of authentication. This section discusses symmetric and asymmetric SAB.

4.3.1. Symmetric SAB

Symmetric SAB (S-SAB) is applicable when both parties lack network-layer authentication information and that authentication is not available from higher-layer protocols. S-SAB can still provide some forms of protection for network and transport protocols, but does not provide authentication beyond continuity of association. S-SAB is useful in situations where transfer of large files or use of other long-lived connections would benefit from not being interrupted by attacks on the transport connection (e.g., via a false TCP RST), but the particular endpoint identities are not important.

Open services, such as web servers, and peer-to-peer networks could utilize S-SAB when their identities need not be authenticated but their communication would benefit from protection. Such services might provide files that are either not validated or validated by other means (e.g., published hashes). These transmissions present a target for off-path attacks that could be mitigated by S-SAB. S-SAB may also be useful for protecting voice-over-IP (VoIP) traffic between peers, such as direct calls between VoIP clients.

S-SAB is also useful in protecting any transport protocol when the endpoints do not deploy authentication, for whatever reason. This is the case for BGP TCP connections between core routers, where the protection afforded by S-SAB is better than no protection at all, even though BGP is not intended as an open service.

S-SAB can also serve as an intermediate step towards S-CBB. S-SAB is the effective result when an IPsec channel is used (via connection latching), but the higher-layer authentication is not bound to the IPsec SAs within the channel.

4.3.2. Asymmetric SAB

Asymmetric SAB (A-SAB) allows one party lacking network-layer authentication information to establish associations with another party that possesses authentication credentials for any applicable IKE authentication mechanism.

Asymmetric SAB is useful for protecting transport connections for open services on the Internet, e.g., commercial web servers, etc. In these cases, the server is typically authenticated by a widely known CA, as is done with TLS at the application layer, but the clients need not be authenticated [4]. Although this may result in IPsec and TLS being used on the same connection, this duplication of security services at different layers is necessary when protection is required from the sorts of spoofing attacks described in Section 2 (e.g., TLS cannot prevent a spoofed TCP RST, as the RST is processed by TCP rather than being passed to TLS).

A-SAB can also secure transport for streaming media such as would be used by webcasts for remote education and entertainment.

4.4. Channel-Bound BTNS (CBB)

CBB allows hosts without network-layer authentication information to cryptographically bind BTNS-based IPsec SAs to authentication at higher layers. CBB is intended for applications that employ higher-layer authentication but that also benefit from additional network-layer security. CBB provides network-layer security services without requiring authentication at the network layer. This enables IPsec security services for applications that have IKE-incompatible authentication credentials. CBB allows IPsec to be used with authentication mechanisms not supported by IKE and frees higher-layer applications and protocols from duplicating security services already available in IPsec.

Symmetric CBB integrates channel binding with S-SAB, as does asymmetric CBB with A-SAB. In both cases, the target applications have similar characteristics at the network layer to their non-channel-binding counterparts. The only significant difference is the binding of authentication credentials at a higher layer to the resulting IPsec channels.

Although the modes of CBB refer to the authentication at the network layer, higher-layer authentication can also be either asymmetric (one-way) or symmetric (two-way). Asymmetric CBB can be used to complement one-way authentication at a higher layer by providing one-way authentication of the opposite direction at the network layer. Consider an application with one-way, client-only authentication. The client can utilize A-CBB where the server must present IKE-authenticated credentials at the network layer. This form of A-CBB achieves mutual authentication, albeit at separate layers. Many remote file system protocols, such as iSCSI and NFS, fit into this category and can benefit from channel binding with IPsec for better network-layer protection, including prevention of MITM attacks.

Mechanisms and interfaces for BTNS channel binding with IPsec are discussed in further detail in [26].

4.5. Summary of Uses, Vulnerabilities, and Benefits

The following is a summary of the properties of each type of BTNS, based on the previous subsections:

	SAB	CBB
Uses	Open services Peer-to-peer Zero-config Infrastructure	Same as SAB but with higher-layer auth., e.g., iSCSI [19], NFSv4 [21]
Vuln.	Masqueraders Needs data rate limit Load on IPsec Exposure to open access	Masqueraders until bound Needs data rate limit Load on IPsec
Benefit	Protects L3 & L4 Avoids all auth. keys	Protects L3 & L4 Avoids L3 auth. keys Full auth. once bound

Most of the potential vulnerabilities in the above table have been discussed in previous sections of this document; some of the more general issues, such as the increased load on IPsec processing, are addressed in the Security Considerations section of this document.

5. Security Considerations

This section describes the threat models for BTNS and discusses other security issues based on the threat models for different modes of BTNS. Some of the issues were mentioned previously in the document but are listed again for completeness.

5.1. Threat Models and Evaluation

BTNS is intended to protect sessions from a variety of threats, including on-path, man-in-the-middle attacks after key exchange, and off-path attacks. It is intended to protect the contents of a session once established, but does not protect session establishment itself. This protection has value because it forces the attacker to target connection establishment as opposed to waiting for a more convenient time; this is of particular value for long-lived sessions.

BTNS is not intended to protect the key exchange itself, so this presents an opportunity for a man-in-the-middle attack or a well-timed attack from other sources. Furthermore, Stand-Alone BTNS is not intended to protect the endpoint from nodes masquerading as legitimate clients of a higher-layer protocol or service. Channel-Bound BTNS can protect from such masquerading, though at a later point after the security association is established, as a masquerade attack causes a client authentication failure at a higher layer.

BTNS is also not intended to protect from DoS (Denial of Service) attacks that seek to overload a CPU performing authentication or other security computations, nor is BTNS intended to provide protection from configuration mistakes. These latter two threat assumptions are also the case for IPsec.

The following sections discuss the implications of the threat models in more details.

5.2. Interaction with Other Security Services

As with any aspect of network security, the use of BTNS must not interfere with other security services. Within IPsec, the scope of BTNS is limited to the SPD and PAD entries that explicitly specify BTNS and to the resulting SAD entries. It is incumbent on system administrators to deploy BTNS only where safe, preferably as an alternative to the use of "bypass" SPD entries that exempt specified traffic from IPsec cryptographic protection. In other words, BTNS should be used only as a substitute for no security, rather than as a substitute for stronger security. When the higher-layer authentication required for CBB is not available, other methods, such as IP address filtering, can help reduce the vulnerability of SAB to exposure to anonymous access.

5.3. MITM and Masquerader Attacks

Previous sections have described how CBB can counter MITM and masquerader attacks, even though BTNS does not protect key exchange and does not authenticate peer identities at the network layer. Nonetheless, there are some security issues regarding CBB that must be carefully evaluated before deploying BTNS.

For regular IPsec/IKE, a man in the middle cannot subvert IKE authentication, and hence an attempt to attack an IPsec SA via use of two SAs concatenated by the attacker acting as a traffic-forwarding proxy will cause an IKE authentication failure. On the other hand, a man-in-the-middle attack on IPsec with CBB is discovered later. With CBB, the IKE protocol will succeed because it is unauthenticated, and the security associations will be set up. The man in the middle will not be discovered until the higher-layer authentication fails. There are two security concerns with this approach: possible exposure of sensitive authentication information to the attackers, and resource consumption before attacks are detected.

The exposure of information depends on the higher-layer authentication protocols used in applications. If the higher-layer authentication requires exchange of sensitive information (e.g., passwords or password-derived materials) that are directly useful or can be attacked offline, an attacker can gain such information even though the attack can be detected. Therefore, CBB must not be used with higher-layer protocols that may expose sensitive information during authentication exchange. For example, Kerberos V AP exchanges would leak little other than the target’s krb5 principal name, while Kerberos V AS exchanges using PA-ENC-TIMESTAMP pre-authentication would leak material that can then be attacked offline. The latter should not be used with BTNS, even with Channel Binding. Further, the ways in which BTNS is integrated with the higher-layer protocol must take into consideration vulnerabilities that could be introduced in the APIs between these two systems or in the information that they share.

The resource consumption issue is addressed in the next section on DoS attacks.

5.4. Denial of Service (DoS) Attacks and Resource Consumptions

A consequence of BTNS deployment is that more traffic requires cryptographic operations; these operations increase the computation required in IPsec implementations that receive protected traffic and/or verify incoming traffic. That additional computation raises vulnerability to overloading, which may be the result of legitimate flash crowds or a DoS or DDoS attack. Although this may itself present a substantial impediment to deployment, it is an issue for all cryptographically protected communication systems. This document does not address the impact BTNS has on such increases in required computation.

The effects of the increased resource consumption are twofold. The consumption raises the level of effort for attacks such as MITM, but also consumes more resources to detect such attacks and to reject spoofed traffic. At the network layer, proper limits and access controls for resources should be set up for all BTNS SAs. CBB SAs may be granted increased resource access after the higher-layer authentications succeed. The same principles apply to the higher-layer protocols that use CBB SAs. Special care must be taken to avoid excessive resource usage before authentication is established in these applications.

5.5. Exposure to Anonymous Access

The use of SAB by a service implies that the service is being offered for open access, since network-layer authentication is not performed. SAB should not be used with services that are not intended to be openly available.

5.6. ICMP Attacks

This document does not consider ICMP attacks because the use of BTNS does not change the existing IPsec guidelines on ICMP traffic handling [8]. BTNS focuses on the authentication part of establishing security associations. BTNS does not alter the IPsec traffic processing model and protection boundary. As a result, the entire IPsec packet processing guidelines, including ICMP processing, remain applicable when BTNS is added to IPsec.

5.7. Leap of Faith

BTNS allows systems to accept and establish security associations with peers without authenticating their identities. This can enable functionality similar to "Leap of Faith" authentication utilized in other security protocols and applications such as the Secure Shell Protocol (SSH) [29].

SSH implementations are allowed to accept unknown peer credentials (host public keys) without authentication, and these unauthenticated credentials may be cached in local databases for future authentication of the same peers. Similar to BTNS, such measures are allowed due to the lack of "widely deployed key infrastructure" [29] and to improve ease of use and end-user acceptance.

There are subtle differences between SSH and BTNS regarding Leap of Faith, as shown in the following table:

	SSH	BTNS
Accept unauthenticated credentials	Allowed	Allowed
Options/Warnings to reject unauthenticated credentials	Yes	No
Cache unauthenticated credential for future refs	Required	Allowed

SSH requires proper warnings and options in applications to reject unauthenticated credentials, while BTNS accepts such credentials automatically when they match the corresponding policy entries. Once SSH accepts a credential for the first time, that credential should be cached and can be reused automatically without further warnings. BTNS credentials can be cached for future use, but there is no security advantage to doing so, as a new unauthenticated credential that is allowed by the policy entries will be automatically accepted.

In addition, BTNS does not require IPsec to reuse credentials in a manner similar to SSH. When IPsec does reuse unauthenticated credentials, there may be implementation advantages to caching them.

SSH-style credential caching for reuse with SAB could be addressed by future extension(s) to BTNS; such extension(s) would need to provide warnings about unauthenticated credentials and a mechanism for user acceptance or rejection of them in order to establish a level of authentication assurance comparable to SSH’s "Leap of Faith". Such extension(s) would also need to deal with issues caused by the absence of identities in BTNS. At best, a cached BTNS credential reauthenticates the network-layer source of traffic when the credential is reused -- in contrast, SSH credential reuse reauthenticates an identity.

Network-layer reauthentication for SAB is further complicated by:

- o the ability of NATs to cause multiple independent network-layer sources of traffic to appear to be one source (potentially requiring acceptance and caching of multiple BTNS credentials),

- o the ability of multihoming to cause one network-layer source of traffic to appear to be multiple sources (potentially triggering unexpected warnings and requiring re-acceptance of the same BTNS credential), and

- o interactions with both mobility and address ownership changes (potentially requiring controlled BTNS credential reassignment and/or invalidation).

These issues are left to be addressed by possible future work on the addition of "Leap of Faith" functionality to BTNS.

In contrast, for CBB, credential caching and verification are usually done at the higher-layer protocols or applications. Caching credentials for CBB at the BTNS level is not as important because the channel binding will bind whatever credentials are presented (new or cached) to the higher-layer protocol identity.

5.8. Connection Hijacking through Rekeying

Each IPsec SA has a limited lifetime (defined as a time and/or byte count) and must be rekeyed or terminated when the lifetime expires. Rekeying an SA provides a small window of opportunity where an on-path attacker can step in and hijack the new SA created by rekeying by spoofing the victim during rekeying. BTNS, and particularly SAB, simplify this attack by removing the need for the attacker to authenticate as the victim or via the same non-BTNS PAD entry that was used by the victim for the original SA. CBB, on the other hand, can detect such attacks by detecting the changes in the secure channel properties.

This vulnerability is caused by the lack of inter-session binding or latching of IKE SAs with the corresponding credentials of the two peers. Connection latching, together with channel binding, enables such binding but requires higher-layer protocols or applications to verify consistency of identities and authentication across the two SAs.

5.9. Configuration Errors

BTNS does not address errors of configuration that could result in increased vulnerability; such vulnerability is already possible using "bypass" SPD entries. SPD entries that allow BTNS must be explicitly flagged, and hence can be kept separate from SPD entries that do not allow BTNS, just as "bypass" SPD entries are separate from entries that create SAs with more conventional, stronger security.

6. Related Efforts

There have been a number of related efforts in the IETF and elsewhere to reduce the configuration effort of deploying the Internet security suite.

The IETF PKI4IPsec effort focused on providing an automatic infrastructure for the configuration of Internet security services, e.g., to assist in deploying signed certificates and CA information [9]. The IETF KINK effort focused on adapting Kerberos [13] for IKE, enabling IKE to utilize the Kerberos key distribution infrastructure rather than requiring certificates or shared private keys [18]. KINK takes advantage of an existing architecture for automatic key management in Kerberos. Opportunistic Encryption (OE) is a system for automatic discovery of hosts willing to do a BTNS-like encryption, with authentication being exchanged by leveraging existing use of the DNS [17]. BTNS differs from all three in that BTNS is intended to avoid the need for such infrastructure altogether, rather than to automate it.

7. Acknowledgments

This document was inspired by discussions on the IETF TCPM WG about the spoofed RST attacks on BGP routers and various solutions, as well as discussions in the NFSv4 and IPS WGs about how to better integrate with IPsec. The concept of BTNS was the result of these discussions as well as discussions with USC/ISI’s T. Faber, A. Falk, and B. Tung, and discussions on the IETF SAAG (Security Area open meeting) mailing list and IPsec mailing list. The authors would like to thank the members of those WGs and lists, as well as the IETF BTNS BOFs and WG and its associated ANONsec mailing list (http://www.postel.org/anonsec) for their feedback -- in particular, Steve Kent, Sam Hartman, Nicolas Williams, and Pekka Savola.

This document was prepared using 2-Word-v2.0.template.dot.

8. Informative References

[1] Aboba, B., Blunk, L., Vollbrecht, J., Carlson, J., and H. Levkowetz, Ed., "Extensible Authentication Protocol (EAP)", RFC 3748, June 2004.

[2] Aboba, B., Tseng, J., Walker, J., Rangan, V., and F. Travostino, "Securing Block Storage Protocols over IP", RFC 3723, April 2004.

[3] CERT Vulnerability Note VU#415294, "The Border Gateway Protocol relies on persistent TCP sessions without specifying authentication requirements", 4/20/2004.

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

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

[6] Heffernan, A., "Protection of BGP Sessions via the TCP MD5 Signature Option", RFC 2385, August 1998.

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

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

[9] Korver, B., "The Internet IP Security PKI Profile of IKEv1/ISAKMP, IKEv2, and PKIX", RFC 4945, August 2007.

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

[11] Melnikov, A., Ed., and K. Zeilenga, Ed., "Simple Authentication and Security Layer (SASL)", RFC 4422, June 2006.

[12] Murphy, S., "BGP Security Vulnerabilities Analysis", RFC 4272, January 2006.

[13] Neuman, C., Yu, T., Hartman, S., and K. Raeburn, "The Kerberos Network Authentication Service (V5)", RFC 4120, July 2005.

[14] Moskowitz, R., Nikander, P., Jokela, P., Ed., and T. Henderson, "Host Identity Protocol", RFC 5201, April 2008.

[15] Ramaiah, A., R Stewart, M. Dalal, "Improving TCP’s Robustness to Blind In-Window Attacks", Work in Progress, January 2008.

[16] Recio, R., Metzler, B., Culley, P., Hilland, J., and D. Garcia, "A Remote Direct Memory Access Protocol Specification", RFC 5040, October 2007.

[17] Richardson, M. and D. Redelmeier, "Opportunistic Encryption using the Internet Key Exchange (IKE)", RFC 4322, December 2005.

[18] Sakane, S., Kamada, K., Thomas, M., and J. Vilhuber, "Kerberized Internet Negotiation of Keys (KINK)", RFC 4430, March 2006.

[19] Satran, J., Meth, K., Sapuntzakis, C., Chadalapaka, M., and E. Zeidner, "Internet Small Computer Systems Interface (iSCSI)", RFC 3720, April 2004.

[20] Shah, H., Pinkerton, J., Recio, R., and P. Culley, "Direct Data Placement over Reliable Transports", RFC 5041, October 2007.

[21] Shepler, S., Callaghan, B., Robinson, D., Thurlow, R., Beame, C., Eisler, M., and D. Noveck, "Network File System (NFS) version 4 Protocol", RFC 3530, April 2003.

[22] Stewart, R., Ed., "Stream Control Transmission Protocol", RFC 4960, September 2007.

[23] TCP SYN-cookies, http://cr.yp.to/syncookies.html

[24] Touch, J., "Defending TCP Against Spoofing Attacks", RFC 4953, July 2007.

[25] Touch, J., A. Mankin, R. Bonica, "The TCP Authentication Option", Work in Progress, November 2007.

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

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

[28] Williams, N. and M. Richardson, "Better-Than-Nothing Security: An Unauthenticated Mode of IPsec", RFC 5386, November 2008.

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

Authors' Addresses

Joe Touch
USC/ISI
4676 Admiralty Way
Marina del Rey, CA 90292-6695
U.S.A.

Phone: +1 (310) 448-9151
EMail: touch@isi.edu

David L. Black
EMC Corporation
176 South Street
Hopkinton, MA 01748
USA

Phone: +1 (508) 293-7953
EMail: black_david@emc.com

Yu-Shun Wang
Microsoft
One Microsoft Way
Redmond, WA 98052
U.S.A.

Phone: +1 (425) 722-6980
EMail: yu-shun.wang@microsoft.com