

IPsec Analysis in an IPv6 Context – v01

By: Merike Kaeo, Double Shot Security

Introduction

IPsec deployments for IPv4 have had a less-than-ideal historical evolution. However, as networks and implementations have evolved in the last decade, many of the negative IPsec perceptions are now myths which need to be corrected. This is especially true for the IPv6 context since IPv6 relies heavily on the IPsec standard(s) for security.

The issue of whether to mandate AH for IPv6 IPsec deployments has come up. The IPsec architecture uses two security protocols to provide traffic security services: the Authentication Header (AH) and the Encapsulating Security Payload (ESP). The AH protocol provides data integrity, data origin authentication and optional anti-replay features for the entire IP packet, including the header. ESP offers a similar set of services with the addition of confidentiality. The ESP protocol could be used without confidentiality to provide only data integrity, data origin authentication and optional anti-replay protection - the main difference however, is that the ESP data integrity and data origin authentication are not computed over the IP header.

Currently, all of the IETF standards detailing IPsec usage in IPv6 environments are specifying ESP w/Null encryption to be implemented as a MUST and AH as a MAY or SHOULD. A historical perspective of why this is the case is enumerated in the next section 'IETF Historical Perspective on AH vs ESP/Null'. This document will analyze the issues that need to be investigated to ensure that this is the appropriate decision for IPv6 networks. The considerations will encompass analysis from both a technical and operational perspective. The resulting analysis is intended as a discussion point between varying vendors and network architects to help reach the appropriate decision as to whether AH must be supported in IPv6 IPsec implementations. [This document is considered a work-in-progress until such a time that a consensus is reached and a final version of this document is available.]

IETF Historical Perspective on AH vs ESP/Null

The main reason for separating the authentication/integrity and encryption portions of IPsec date back to before US export controls were relaxed in the mid 1990s. Before 1996, there were US export controls on any encryption keys greater than 40 bits and it was unclear whether ESP would ever be exportable. The restrictions were more relaxed for any cryptographic methods used solely for authentication purposes. The thought was that AH would at least provide authenticated IP addresses. Most systems on the Internet

at that time were using some form of rhost-like IP-based access control. SSH did not yet exist, just kerberized rlogin and kerberized telnet.

When IPsec implementations started shipping in 1997, many networks had already started to adopt the use of NATs. The need for ESP/Null [ESP with null encryption, RFC 2410] became apparent at the Cisco/ICSA IPsec & IKE March 1998 Interoperability Workshop. AH would not work with NATs while ESP/Null was NAT friendly since it would only provide integrity protection for the payload and not the IP header fields.

Around that time, in mid 1999, researchers from Counterpane published an analysis paper on IPsec: <http://www.schneier.com/paper-ipsec.pdf>. The paper recommended a number of changes to IPsec, including the removal of AH from the protocol. The main reasoning behind the recommendations was to reduce the complexity of the protocol with the reasoning that a complex security protocol introduces more errors and bugs. Note that this paper was criticized within the IPsec mailing list although it was acknowledged that only the mailing list archives and working group minute notes gave any clue as to why certain decisions were made which lead to the existing set of standards.

Due to the problems of AH in a NATed environment and the recommendation to remove AH by the Counterpane paper there were many discussions on deprecating AH altogether. An IETF draft document which was never completed pointed out security properties of IPsec using IKEv1.¹ This was an attempt to help counter some points made in the Counterpane researcher's paper as well as to provide the reasoning for making future changes to the existing IPsec protocol. In that document the following statement is made:

When tunnel mode is being used, AH has the same effective coverage as ESP, because the outer header is merely a transient routing header. If AH is being used to ensure that the header of the IP packet remains uncorrupted during transit, this is really only useful if any of the intermediate routers which interpret the header are also privy to the AH key.

A more detailed analysis on the above statements is in the next section 'Security Tradeoffs between AH vs ESP w/Null Encryption'.

In the 53rd IETF IPsec meeting (March 2002), there was a discussion about whether or not AH should be deprecated². It was pointed out that groups like VRRP may have a concern with that. However an informal pool of the room indicated that most people were in favor of discarding it. Another comment was made that while deprecating AH as an IPsec requirement could be decided on in the IPsec group, AH could still have a life as an independent protocol. Additionally there was a comment that IDS vendors would like to quickly look at packets and determine if it was encrypted or not – AH would provide

¹ <http://tools.ietf.org/id/draft-ietf-ipsec-properties-02.txt>

² <http://www3.ietf.org/proceedings/02mar/192.htm>

that capability.

Currently in the IETF, the new IPsec RFCs and any IPv6 RFCs and drafts that explicitly discuss the use of IPsec refer to ESP/Null as the mandatory to implement protocol while AH is always a MAY or SHOULD.

Usage Scenarios for IPv6/IPsec

While the security models of current IPv4 production networks must be evolved for IPv6 networks, there are features in IPv6 and new trends in networking that should lead us to changing security paradigms. End-to-end security between hosts has had limited operational practicality in IPv4-based networks but is a key feature of IPv6. A return to the end-to-end network model should be architected into any dual stacked transition architecture with careful consideration for not compromising IPv4 security. In addition, IPv4 security practices should not limit or compromise IPv6 network architectures. This is especially true for situations where IPv6 traffic may be tunneled over IPv4 or IPv4 traffic tunneled over IPv6.

Some of the operational concerns of whether AH is required over ESP/Null can come back to the controversy of host based security versus network based security. It is always going to be the case that a layered security approach is necessary and a hybrid model that encompasses both host and network security will be used in IPv6 network architectures. How much an environment relies on application, host and/or network-based security is most likely environment dependent since it relies on multiple factors including installed base, vendor implementations and operational experience.

The following are sample scenarios that will be required for using IPsec to protect IPv6 traffic:

1. End-to-end IPv6 communications with encryption and integrity services
2. End-to-end IPv6 communication with only integrity services
3. Network-based filtering capability
4. Network-based load balancing capability
5. Network-based QoS differentiation
6. Network-based flow detection
7. Network-based IDS
8. Scenarios 1 through 7 where IPv6 is tunneled over IPv4
9. Scenarios 1 through 7 where IPv4 is tunneled over IPv6

[Author note: How and where these all fit together is interesting at best. Needs more input and fleshing out. The main architectural concern for IPv6 networks is that the security model will not (should not) blindly mimic IPv4 security architectures. With IPv6 there will be environments that will want more true end-to-end secured communications. But where does the network-based intelligence come in? Which users

will prefer end-to-end security over network-based control? Which users will prefer network-based intelligence over true end-to-end? There will be both.....]

For any networking device that wants to verify and take action on AH-protected fields, that device must act as an IPsec peer to at least the device whose traffic it is trying to verify. For performance reasons the network-based device could perform the AH ICV check once and then rely on subsequent cached entries for all subsequent packets of the same flow. Since most flows are relatively short and to insert invalid traffic that matches the flow is highly unlikely, this seems a valid performance / security tradeoff.

If a network device is acting on a bidirectional flow, the network device will act as an IPsec peer to both the sending and the receiving node and effectively act as a proxy between the two communicating parties.

Note also that if a networking device had access to the peer credentials that were used for the authentication, then there could be a scenario where a device can validate the ICV and act on information without being an actual IPsec peer proxying between the two end-to-end communicating parties.

Security Tradeoffs between AH vs ESP w/ NULL encryption

Many IETF IPv6 documents reference RFC2401bis or RFC4301 (which obsoletes RFC2401) when discussing how to protect IPv6 traffic using the required IPsec protocol. RFC4301 states the following:

IPsec implementations **MUST** support ESP and **MAY** support AH. (Support for AH has been downgraded to **MAY** because experience has shown that there are very few contexts in which ESP cannot provide the requisite security services. Note that ESP can be used to provide only integrity, without confidentiality, making it comparable to AH in most contexts.)

Although it may be intuitive for IPsec implementers how and why using ESP is comparable to using AH for most contexts, the following is a detailed analysis of what fields are left unprotected when using either protocol in an IPv6 environment and how it would affect operational security issues. Both transport and tunnel modes are taken into consideration.

Transport Mode AH or ESP/Null

Transport mode AH or ESP is used when the cryptographic endpoints are the source and destination of the data packet. While most individuals would immediately think of client-to-server models based on PCs, laptops and servers running a myriad of varying

operating systems, transport mode IPsec can also be used between networking devices that need to communicate with other devices. For example, syslog servers, TFTP servers, SNMP servers, etc. I.e. in these instances a router or firewall networking devices would also become an IPsec endpoints.

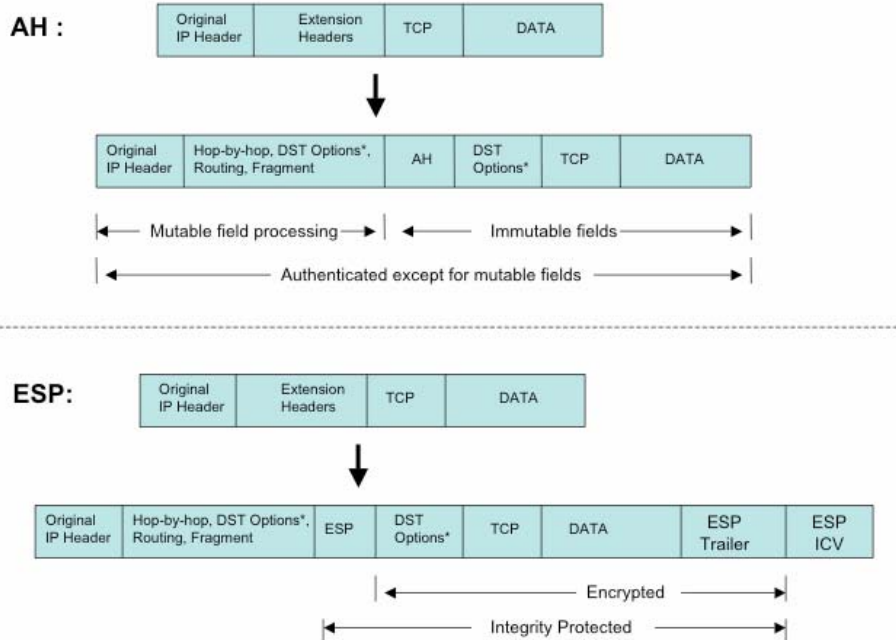
In IPv6, the transport mode security protocol header appears after the base IP header and selected extension headers. The base header is a fixed 40 bytes while the number of extension headers is variable. RFC2460 provides the following recommendation for ordering the extension headers:

1. IPv6 Base Header
2. Hop-by-Hop Options Header: the only extension header that must be processed by routers. It is used for the support of jumbo grams (RFC2675) or, with the router alert option, used for IPv6 multicast and IPv6 RSVP
3. Destination Options Header: used to specify a process that needs to be performed by the destination node; used in IPv6 mobility
4. Routing Header: used to specify routing path; used in IPv6 mobility and in source routing
5. Fragmentations Header: denotes fragmented packets
6. Authentication Header: provides data authentication, integrity and optional anti-replay services
7. Encapsulating Security Payload Header: provides data authentication, integrity, an anti-replay service and traffic flow confidentiality
8. Destination Options Header: used to specify a process that needs to be performed by the destination node; used in IPv6 mobility
9. Upper-layer headers: TCP, UDP, SCTP, etc.

There is also a mobility header which would most likely follow the AH/ESP headers (?). The mobility header is used by mobile nodes, correspondent nodes and home agents in all messaging related to the creation and management of bindings.

The only MUST requirement in RFC2460 is that the Hop-by-Hop extension header has to be first. The security protocol header(s) may appear before or after destination options but must appear before next layer protocols (e.g., TCP, UDP, SCTP).

IPv6 IPsec AH/ESP in Transport Mode



AH Protection

The main difference between using transport mode AH vs ESP/Null is that the IPv6 header and optional extension headers are not part of the hash computation when using ESP/Null. However, for AH, the integrity check value (ICV) is not computed over every field in the IPv6 header or the optional extension headers. Any field which is mutable, i.e. may be modified during transit, is set to zero during the AH ICV computation and is therefore not included in the integrity check. Note that any field which is mutable but predictable at the receiving end is included in the integrity check calculation. Everything after AH in an IPv6 packet is assumed to be immutable during transit and therefore is protected.

The following list the protected, mutable but predictable (protected) and mutable (not protected) fields in the IPv6 base and extension headers. (from RFC4302)

Base IPv6 Header

Protected:

- Version
- Payload Length
- Next Header
- Source IP Address
- Destination IP Address (without Routing Extension Header)

Mutable but Predictable (protected):

- Destination IP Address (with Routing Extension Header)

Mutable (not protected):

- DSCP (Differentiated Service Codepoint) [RFC2474]
- ECN (Explicit Congestion Notification) [RFC3168]
- Flow Label
- Hop Limit

Note that RFC4302 deliberately left the flow label mutable even though RFC3697 states that the flow label value set by the source **MUST** be delivered unchanged to the destination node(s). The reasoning was that the flow label described in AHv1 was mutable, and in RFC 2460 was potentially mutable. To retain compatibility with existing AH implementations, the flow label is not included in the ICV in AHv2. [Is this an issue that should get revisited? How much will it break in existing AH deployments?]

Hop-by-Hop Options Header

This header contains a bit that indicates whether the option might unpredictably change during transit. All options for which the bit indicates immutability are included in the ICV calculation. For any option for which contents may change during transit between sending and receiving party, the entire option data field will be treated as zero-valued octets when computing or verifying the ICV. However, the Option Type and Option Data Length are still included in the ICV calculation.

Destination Options Header

This header contains a bit that indicates whether the option might unpredictably change during transit. All options for which the bit indicates immutability are included in the ICV calculation. For any option for which contents may change during transit between sending and receiving party, the entire option data field will be treated as zero-valued

octets when computing or verifying the ICV. However, the Option Type and Option Data Length are still included in the ICV calculation.

Routing Header

Protected: N/A

Mutable but Predictable (protected): Type 0 Routing Header

Mutable (not protected): N/A

[what about type 2 routing header used for mobility?]

Fragmentations Header

There is no protection since fragmentation occurs after outbound IPsec processing and reassembly occurs before inbound IPsec processing.

Mobility Header

Protected: ??

Mutable but Predictable (protected): ??

Mutable (not protected): ??

[since mobility folks want ESP and encryption and then there's rfc4285, would the AH vs ESP/Null issue even be relevant with the mobility header?!?]

AH Validation

Any device that wants to modify or perform an action on fields that are protected by transport mode AH, for example filter on IP addresses, must validate the AH ICV. From a true end-to-end protection perspective, this is obvious. However, for any networking device that wants to verify and take action on AH-protected fields, that device must act as an IPsec peer to at least the device whose traffic it is trying to verify. If it does not perform any verification, then there is no difference in assuring that the IP header fields have been modified and there is no practical security advantage over using ESP/Null.

It can also be likely that the network device will act as a peer to both the sending and the receiving node and effectively act as a proxy between the two communicating parties. Alternatively, if a networking device had access to the peer credentials that were used for the authentication, then there could be a scenario where a device can validate the ICV and act on information without being an actual IPsec peer proxying between the two end-to-end communicating parties.

ESP/Null vs AH Operational Risk Impact

Because in transport mode ESP/Null does not provide any integrity protection for the IPv6 base header and optional extension header fields that are protected by AH, what would be the operational risk from a realistic threat perspective?

I.e. what threats can be realized using ESP/Null protection end-to-end rather than AH end-to-end?

The Security Parameter Index (SPI) plays an important role in some of this evaluation. The SPI has the following characteristics:

- helps receiver identify the SA to which an incoming packet is bound
- for unicast it can be used by itself to specify an SA or in conjunction with the protocol type (AH or ESP)
- for unicast SAs the SPI value is generated by the receiver
- demultiplexing algorithm is used for IPsec implementations that support multicast for mapping inbound IPsec packets to SAs
- SPI assignment is not negotiated or coordinated with the key management subsystems that may also be part of a multicast group – it is possible that a group security association and a unicast security association simultaneously use the same SPI

IP Address Spoofing

IPsec Security Associations (SAs) are established based on Source/Destination IP addresses and protocol port numbers. The receiver matches packets against SPI and inbound selectors associated with the SA. Successful verification implies that the packet came from the correct source and destination IP addresses. [Note that this is the same source address that established the SA, not necessarily who the receiver thought should have set up the SA – is this an issue between AH vs ESP/Null? In both cases anyone can spoof an address and it will get dropped when an SPD check is done]

Are there instances where a source or destination address can be spoofed and the packet is erroneously processed? Where is AH protection adding value over ESP/Null protection when looking at SPD processing? Do IKEv1 vs IKEv2 identity protection mechanisms add any value here?

If a network device acts as an IPsec proxy, i.e. if it is participating in the SA set-up, will spoofed addresses ever get through?

If the network device had access to the peer credentials that were used for the authentication and could validate the AH ICV and act on information without being an actual IPsec peer then there would be an advantage of using AH. With ESP/Null you have no mechanism to validate the IP addresses this way. What is likelihood that this scenario is warranted?

Hop-by-Hop Options Header

The Option Type and Option Data Length are included in the AH ICV calculation. The option data itself may or may not be included depending on whether it gets modified in transit.

Using ESP/Null there is no validation that the option type and/or option length are valid until processing is done at the receiving node.

[what is real operational risk impact?]

Destination Options Header

The Option Type and Option Data Length are included in the AH ICV calculation. The option data itself may or may not be included depending on whether it gets modified in transit.

Using ESP/Null there is no validation that the option type and/or option length are valid until processing is done at the receiving node.

[what is real operational risk impact?]

Routing Header

The type 0 routing header is protected which is used in scenarios for source routing. However, source routing is currently considered a security issue and is usually not used.

Some may argue that this is due to the inability to authenticate the origin of the packet. It is intended to be a transit-provider selection tool, but since packets could be spoofed, the transit-provider can't tell if it should accept this packet (and this route) as being from a real customer. [Is it realistic to assume that AH would fix that if there were a way for

the transit-relay to acquire the authentication materials? This would require ISP buy-in to use IPsec.....only likely if configurations are A LOT simpler and there is no operational / performance impact . How does it compare and/or relate to the SIDR work?]

[what about type 2 routing header used for mobility?]

Fragmentation Header

There is no advantage/disadvantage using AH. There is no protection since fragmentation occurs after outbound IPsec processing and reassembly occurs before inbound IPsec processing.

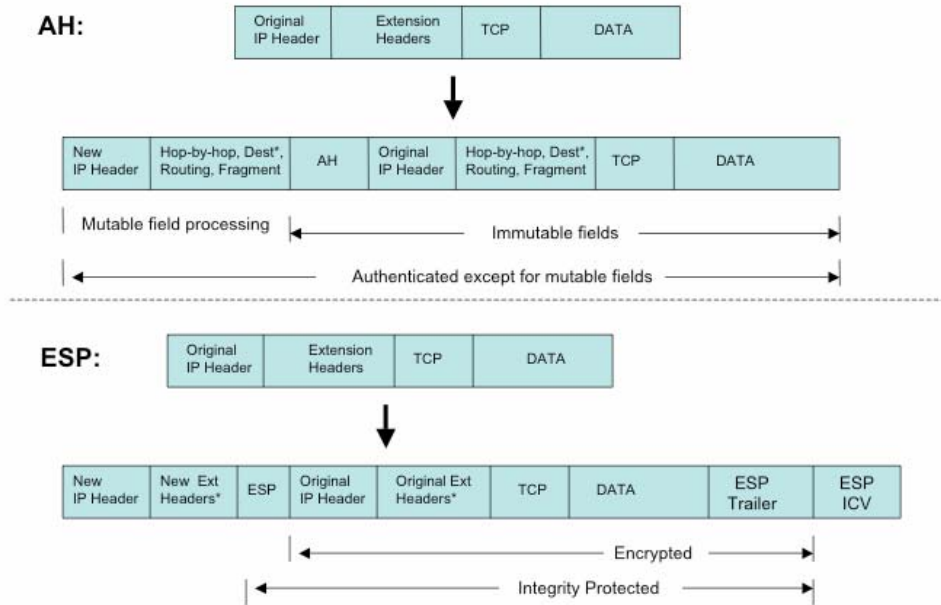
Mobility

[what is operational risk impact using AH vs ESP/Null?]

Tunnel Mode AH or ESP/Null

Tunnel mode AH or ESP is used when the cryptographic endpoints are not the IPsec peers. Typically tunnel mode is used when devices provide IPsec protection services on behalf of traffic not sourced by or destined to the IPsec peers.

IPv6 IPsec AH/ESP in Tunnel Mode



The main difference between using tunnel mode AH vs ESP/Null is that the outer IPv6 header and optional extension headers are not part of the hash computation when using tunnel mode ESP/Null but are included for AH. Note however that for tunnel mode AH, the integrity check value (ICV) is not computed over every field in the new IPv6 header or the new optional extension headers.

In both the tunnel mode AH and ESP/Null case, the original IPv6 base header and optional extension headers are integrity protected.

The analysis for tunnel mode follows the same logic as the analysis for transport mode. The analysis is simply based on the new IPv6 header and the new optional extension headers.

Differentiating Between IPsec Integrity Protected Traffic vs IPsec Encrypted Traffic

What are the requirements for having the capability to differentiate between encrypted vs unencrypted traffic? For transport mode IPsec, if there is no need for network-based actions there is no requirement. However, if a network-based device wants to add some

network intelligence then being able to differentiate between encrypted vs unencrypted traffic will give that device the capability to quickly process the unencrypted traffic, as in the case of IDS devices.

Additionally, end users may have requirements that restrict encrypted traffic use, and therefore network devices could quickly discern and drop any encrypted traffic.

In today's environment you can easily differentiate between AH integrity protected traffic versus ESP traffic since they use different protocol numbers. An argument can be made that AH is about establishing trusted values in the part of the packet that the network is supposed to look at and act on. On the contrary, ESP (either encrypted or not) is about a relationship between the endpoints. If relying on ESP/Null, the network has to use heuristic or IKE snooping to establish any trust about what is happening on a given packet stream. In either case for ESP there is a non-zero probability of getting it wrong which would result in improperly passing or blocking packets, or applying the wrong QoS values.

However, it is not a valid assumption that all ESP traffic is encrypted. How do you then enforce a policy which may only allow encrypted traffic through? There may be a need to come up with an interoperable mechanism to differentiate between ESP/Null and ESP encrypted traffic anyway.

Some thoughts on how to differentiate between ESP/Null and ESP encrypted traffic:

- use assigned SPI numbers: cannot do this since the SPI is randomly generated by the receiver and this mechanism would completely break the existing protocol
- use a new protocol number: this may be hard since there are only 256 protocol numbers and IANA may not want to give another one to just integrity protected traffic since 51 already exists for AH
- others?!?

Hardware Performance Concerns Between AH vs ESP w/ NULL encryption

What are the perceptions versus real issues with hardware performance? [this section needs more input from hardware specialists to validate any claims]

The most prominent hardware performance concerns with using ESP/Null today is having to pick up the trailer to determine if the packet is encrypted or not (which requires parsing of the entire packet) and the variability in the ICV length. While some argue that the silicon buffers in today's implementations are not big enough to handle an entire IPv6 packet, a contrary view is that for reliable deep content IPS/IDS type inspection you would need to have the capability to check the entire packet anyhow.

Is it true that if the ESP/Null packet were read in and instructions parallelized, that performance of parsing an ESP/Null packet versus parsing an AH packet may not be so different?

You do not need to know the protocol number before you start parsing. With hardware you can do parallel checking of the TCP/UDP and ICMP headers. For TCP you just check the flags to see if this is a new connection or not. If it is a new connection then the packet will be short anyways. So the protocol number will be available even in the short buffer. If it is an existing flow then the IP addresses and port numbers can be searched from the flow table and if found, you can check the sequence number, etc. If those match then you can continue doing the rest of the checking, like checking window size, etc. To verify the checksum the whole packet needs to be checked anyway. The probability for the random packet to have valid TCP (or UDP) header is not that big. The final check and committing new flows should be delayed until the protocol number can be checked from the end.

Assumes a known ICV size. SHA 256 is already a problem, and we need to design a system that accommodates the eventual replacement of SHA256 with likely another new ICV length. SHA-256 is a big issue when parsing. The variability of the ICV length between these and 256 assures match failures. [needs more detail on this]

If an IPS/IDS system is deployed and checks even deeper into the packet then the probability of the packet passing all the tests before reaching the protocol number is minimal unless the packet is really UDP or TCP packet. In an IPS/IDS case, the device will usually check the full packet anyway so it will not matter what the protocol number is,

Why do we want to know the protocol number in the first place?

Assumes that encrypted traffic is to be filtered. Prevents the use of e2e encrypted where the network is trying to react to the unencrypted flows.

If we want to filter out all non-TCP/UDP/ICMP traffic based on the protocol and nothing else, then there will be a pass rate of 3 out of 256 when checking the protocol number in case encrypted ESP traffic is seen instead of ESP/Null. That will still cause the encrypted ESP traffic to be unusable, thus effectively preventing the valid users of using encrypted ESP.

As none of the contents is authenticated and the one who wants to bypass those checks can always use some other method of encapsulation those users are not really issue here.

Normal firewall type or rules will drop the pass rate of encrypted traffic to very small (i.e. the probability that random encrypted ESP packet will have valid TCP/UDP flow type header (valid port numbers) is $n+m$ out of 2^{32} , where n is the number of TCP flows between those IP-addresses and m is number of UDP flows between those same IP-addresses. As the number of flows between those specific IP addresses is small this

probability gets small very quickly.

Operational Usability Considerations

Since AH was being deprecated given to its lack of utility in IPv4, there hasn't been much testing even with current implementations – mindset will have to be changed.

If AH is deemed to be necessary for IPv6 traffic integrity, what are the actual cases where it should be used? Always when integrity/authentication is needed? If ESP is used for encryption would AH be also always be used? Would you ever use ESP/Null for IPv6 traffic? [need more input here]

It is critical to have vendor collaboration on default behavior to avoid more complexity and confusion **IF** AH is to be the IPsec integrity approach for IPv6. It is also important to ensure that operators understand that AH will not be re-introduced for IPv4 since it breaks NAT environments. This is only a consideration for IPv6 networks.

Tunneling IPv6 over IPv4 and IPv4 over IPv6

Many transition mechanisms will require that IPv6 traffic be tunneled over IPv4 or that some IPv4 traffic will get tunneled over IPv6. It is clear that any IPv6 traffic that gets tunneled over IPv4 will prohibit the use of AH due to NAT concerns.

Summary

To summarize the issues brought forth in this paper, the decision on whether to mandate using AH for IPv6 IPsec implementations rely on the following:

- Does AH provide enough of a security benefit over using ESP/Null (in either transport or tunnel mode) for IPv6 network architectures?
- If there were a mechanism to more easily and reliably differentiate between ESP/Null and ESP encrypted traffic, would that affect the reasoning to mandate AH for IPv6?
- Are the hardware performance concerns for parsing AH versus ESP/Null packets for deeper packet inspection based on perception or empirical data?
- What will be the effect on practical operational deployment issues if AH is used to provide integrity protection for IPv6 traffic? How will easy configurations and out-of-the-box interoperability be assured?

Acknowledgments

The author would like to acknowledge the following individuals for help with initial review and understanding of the issues being raised: Derrel Piper, Tero Kivinen, Tony Hain, Patrick Grossetete.

Conclusion

TBD.