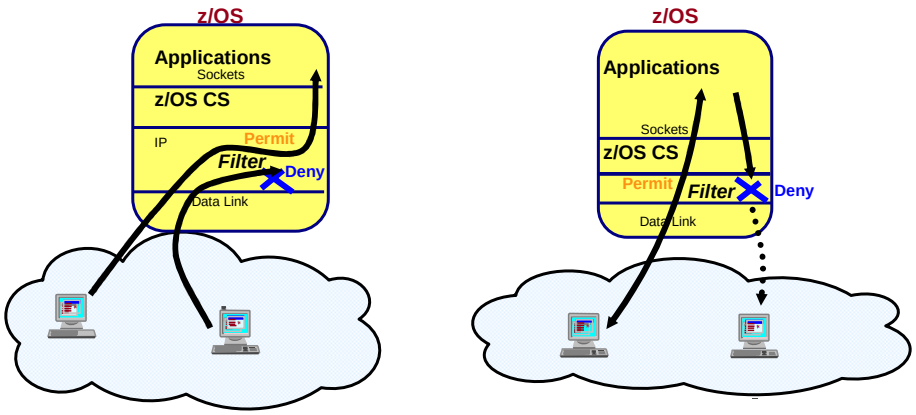
This presentation covers changes made for IPSec RFC currency

Background information – IP filtering



Many of the changes made for IPSec RFC Currency impact the IP filtering function. For a brief review, the IP filtering function of IPSec provides for control over IP packets that are sent and received by z/OS. Administrators can define a list of *filter rules* that are examined for every packet that is sent and received. The filter rules specify a set of selectors, or criteria, that indicate what sort of packet to look for; and they specify an *action* to take for such packets.

Consider this example. A simple set of filter rules might allow all traffic for TCP port 80. It might also encrypt all traffic to and from subnet 192.168.1.0/24, and allow packets to be routed through this system from address 192.168.2.1 to address 192.168.3.1.

Because filter rules might be overlapping in their specifications, the order of filter rules is important. The possible actions for a filter are to permit the packet, deny the packet, or permit the packet but ensure it is encrypted or authenticated using IPSec.

RFCs

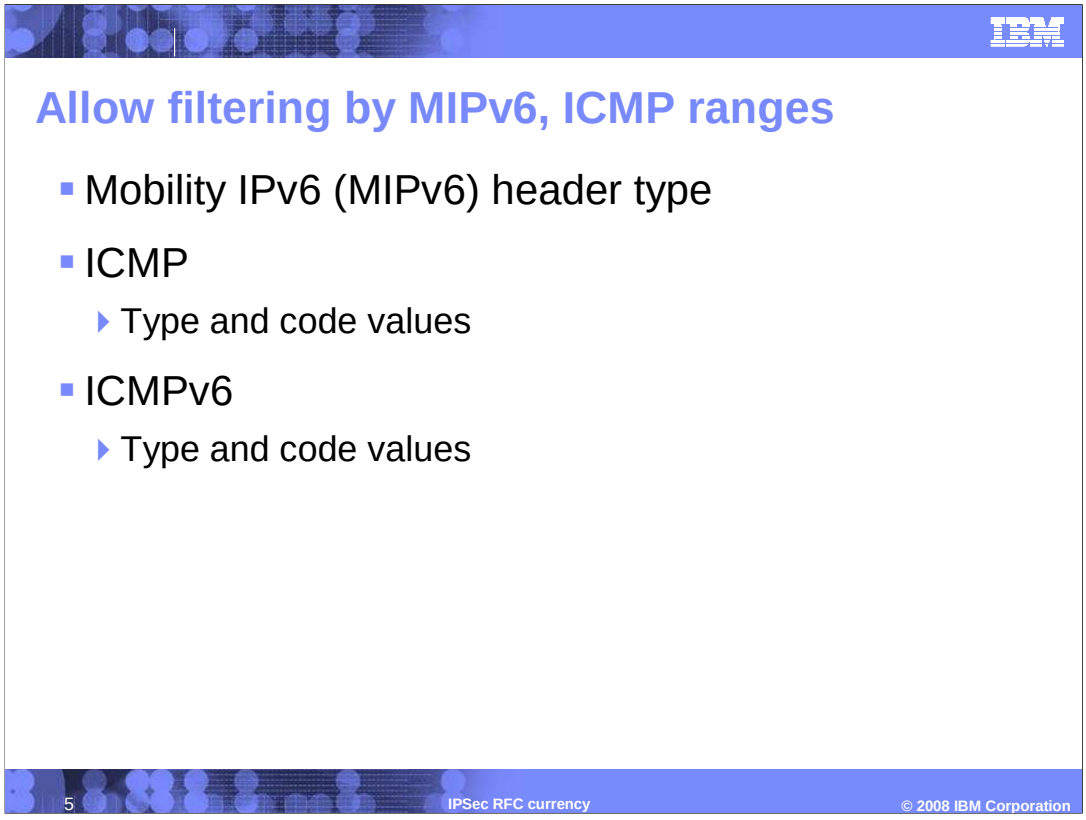
- Many of the RFCs that define IPSec were drafts when implemented by Communications Server for z/OS.
- Have since become standards, but in changed form

Many of the RFCs that define IPSec were drafts when implemented by Communications Server for z/OS. They have since become standards, but in changed form, so in V1R10 Communications server implemented support for these updated IPsec RFCs

RFCs

- In V1R10 Communications server implemented support for these updated IPsec RFCs:
  - RFC 4301 – Security Architecture for the Internet Protocol
  - RFC 4302 – IP Authentication Header
  - RFC 4303 – IP Encapsulating Security Payload
  - RFC 4304 – Extended Sequence Number . . .
  - RFC 4308 – Cryptographic Suites for IPsec
  - RFC 4835 – Cryptographic Algorithm Implementation Requirements . . .

With the exception of functions that require the use of the IKEv2 protocol, the mandatory requirements of the IETF IPsec RFCs listed on this chart are supported in z/OS Communications Server V1R10.



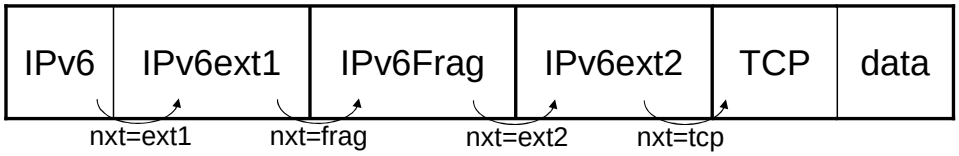
RFC 4301 requires support for filtering by the Mobility IPv6 header *type* field, including ranges of types. It also requires support for filtering by ranges of ICMP *type* and *code* values and ICMPv6 *type* and *code* values. Support for all these is added to z/OS Communications Server V1R10, including the z/OS Configuration Assistant GUI.

Some example ICMP types are echo request (8), echo reply (0) and destination unreachable (3). For ICMP destination unreachable messages, there are codes such as host unreachable (1) and port unreachable (3). ICMPv6 is similar in function to ICMP but generally uses different values; its echo request is type 128, echo reply type 129, and destination unreachable type 1. For destination unreachable it has similar codes, such as address unreachable (3) and port unreachable (4).

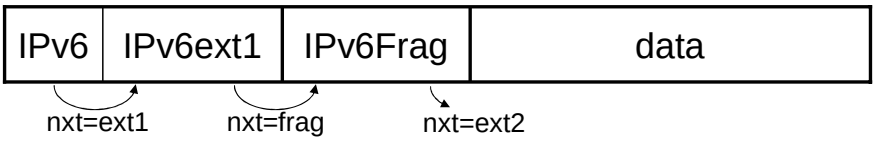
Mobility header type values include Home Test Init (1), Home Test (3), and Binding Update (5).

Background for opaque

First IPv6 fragment:



Subsequent IPv6 fragment:



Every IPv4 packet contains the upper-layer protocol (such as TCP) directly within the IP header. IPv6 works differently; the IPv6 IP header only identifies the next header in sequence, and the extension headers form a chain up until the upper-layer protocol header. The last IPv6 extension header in sequence identifies what the upper-layer protocol is (such as TCP).

Normally this is no problem, but it poses a difficulty when IPv6 packets are fragmented. If there are any IPv6 extension headers between the IPv6 fragment header and the upper-layer header, then it is impossible to determine the upper-layer protocol for some of the packet fragments.

Consider the example on this page. The original IPv6 packet was broken into two fragments. The first fragment represents a TCP packet; if you walk down the chain of extension headers you will find that the last extension header points to the TCP header. However, you cannot determine what the upper-layer protocol is for the second fragment. Because the *IPv6ext2* header is not present in this fragment, the TCP protocol value is not known for this packet.

This is a very unusual case, but it poses a problem for IP filtering. How do you filter a packet when you don't know its protocol? The solution to this problem is addressed on the next slide.

New IpService protocol: opaque

IP packet	Filter rule's IpService protocol				
	TCP	UDP	ICMP	**Opaque**	All
TCP	✓	×	×	×	✓
UDP	×	✓	×	×	✓
ICMP	×	×	✓	×	✓
Unknown	×	×	×	✓	✓

You saw on the previous slide that for some IPv6 routed packets you might not know the upper-layer protocol value. RFC 4301 requires a way to select such packets for filtering. The Opaque keyword is added to the Protocol parameter on the IpService statement to support matching routed IPv6 packets with unknown protocol value. Note in this chart that packets with unknown protocol will match both *Opaque* and *All* protocol specifications on the IpService statement.

The Opaque option is also made available in the z/OS Configuration Assistant GUI.

Routed port restrictions

- Filter rules for routed traffic must specify ALL ports
  - no specific port numbers allowed if RFC-compliant

Before:

Filter type	From	To	Protocol	Source port	Dest port	Action
Routed	1.1.1.1	2.2.2.2	TCP	ALL	80	Permit
Routed	1.1.1.1	2.2.2.2	TCP	ALL	23	ESP AES
Routed	1.1.1.1	2.2.2.2	TCP	ALL	25	ESP DES

After:

Filter type	From	To	Protocol	Source port	Dest port	Action
Routed	1.1.1.1	2.2.2.2	TCP	ALL	**ALL**	**ESP AES**

z/OS Communications Server V1R10 introduces new restrictions on filter rules to comply with RFC 4301. These restrictions only apply to routed or forwarded traffic. If all of your filter rules apply to local packets then you are not affected by this restriction.

RFC 4301 identifies potential security risks in routed traffic where packets might be fragmented. If a forwarding host has filter rules that apply to specific TCP or UDP ports, some of the fragmented packets will have ambiguous filtering decisions because their ports are not known. This ambiguity introduces several security risks.

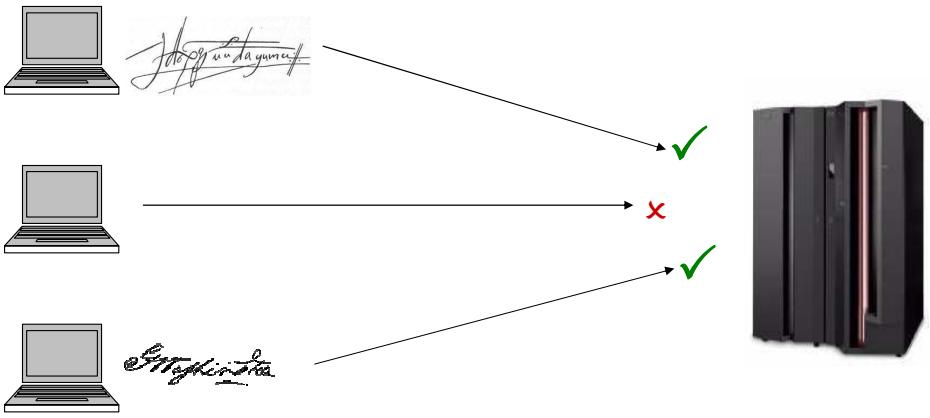
RFC 4301 allows two possible solutions. One is to implement stateful fragment checking, which temporarily stores the port values from the first fragment so that they can be used later to properly filter subsequent fragments. z/OS Communications Server does not support stateful fragment checking. The second solution allowed by RFC 4301 is to prevent routed filter rules from specifying specific ports.

Beginning in V1R10, z/OS Communications Server Policy Agent and Configuration Assistant will disallow filter definitions that apply to specific ports for routed traffic. All filter rules for routed traffic must apply to all ports. You are affected by this restriction if you have any port-specific filter rules that apply to routed traffic, or that apply to both routed and local traffic. The *z/OS Migration* manual provides instructions for updating your policy to comply with this restriction. In particular, the z/OS Configuration Assistant has the ability to import your policy and recommend updates to comply with the restriction. This slide shows an example of such an update; three separate routed filter rules for specific ports have been combined into a single filter rule. Notice that the different actions for the original filter rules had to be merged into a single action (in this case, the most secure action). Any such changes will need to be coordinated with all IPSec peers that are affected by the change.

This restriction can be temporarily suspended until you update your policy to comply with the restriction. As an interim measure, the RFC4301Compliance parameter is added to the IpFilterPolicy statement to control whether this restriction is enforced. You can choose to relax the restriction until you have updated your filter policy. If you choose to relax the restriction, you should be aware that the vulnerabilities cited in RFC 4301 concerning routed traffic and fragmented packets will apply to you. The z/OS Configuration Assistant GUI provides a similar option to relax enforcement of the routed port restrictions.

RemoteIdentity for mobile users

- New filter selector matches packets on user's IPSec identity
  - Mobile user has many IP addresses over time

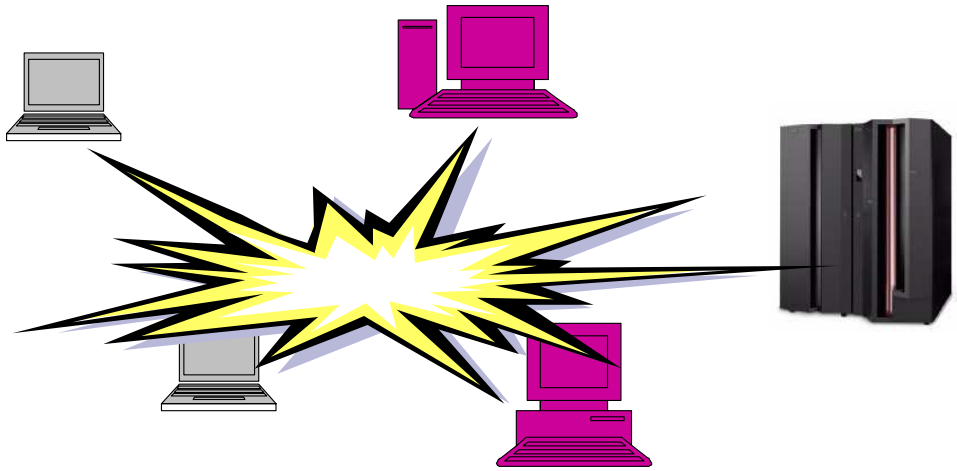


Mobile users present a challenge for IP filter configuration since their IP address is not fixed and might be unpredictable. RFC 4301 addresses this challenge by defining a new filter selector that can be used to match packets: *remote identity*. Instead of selecting on remote IP address, the mobile user's IKE identity is used to select traffic, and IPSec protection is required (in order that the IKE identity is known). This security model is configured similarly to other security models that require IPSec protection. The difference is that the peer's remote identity is indicated on the new RemoteIdentity statement in the IpFilterRule, and the IpDestAddr is typically wildcarded to all addresses.

The z/OS Configuration Assistant GUI allows for the configuration of remote identity using a new type of connectivity rule.

Multicast security associations

- Manual security associations supporting wildcarded IP addresses
  - For sending and receiving multicast traffic

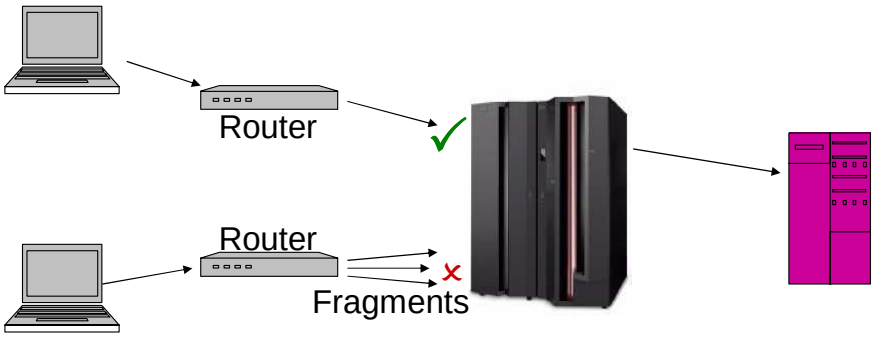


RFC 4301 requires support for a hierarchy of manual security associations – manual tunnels – sharing the same Security Parameter Index (SPI) value. These associations must have ability to support manual security associations that are not specific to local or remote IP address. The primary use case for this is to support protecting multicast traffic using manual IPsec Security Associations (SAs). Multicast SAs must support wildcarded IP addresses because the local and remote endpoints for the traffic are variable. Support was added to z/OS Communications Server to allow the wildcarding of security endpoint IP addresses for manual tunnels, and to support finding the manual tunnel with the most specific address match for a given SPI. The z/OS Configuration Assistant also supports creating multicast SAs.

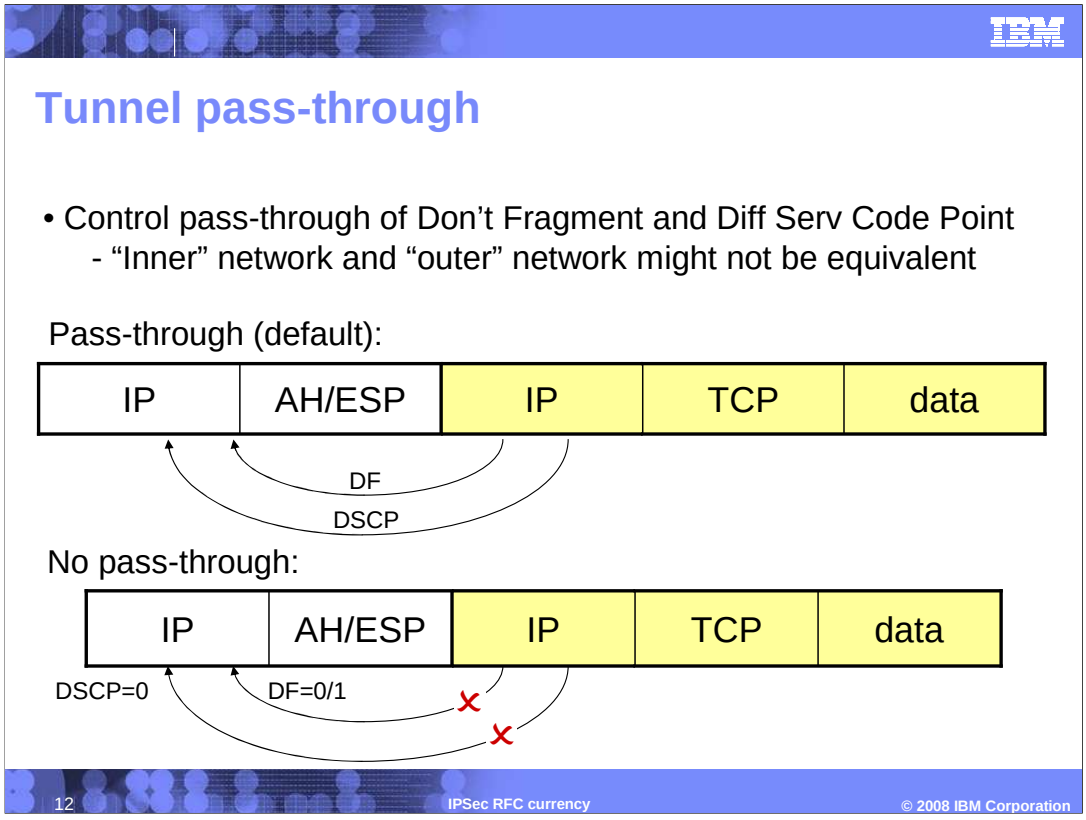
For more detail, see section 4.1 of RFC 4301; the IpManVpnAction statement for the Policy Agent; and the “Additional Topologies” section of the IP Security chapter in the *z/OS Communications Server IP Configuration Guide.*

New FragmentsOnly keyword of IpService

- Allows a filter rule that discards all received fragmented packets
  - If you know fragmentation will not occur, it is probably an attack



RFC 4301 requires support for discarding fragmented packets. This is useful for environments where it is known that fragmentation will not occur, and for which all fragmented packets are possible fragment attacks are therefore regarded with suspicion. The FragmentsOnly keyword is added to the IpService statement so that a filter rule can be defined that matches only fragmented packets. This is permitted only in combination with a permit or deny action, not an ipsec action. This support is also available in the z/OS Configuration Assistant GUI.

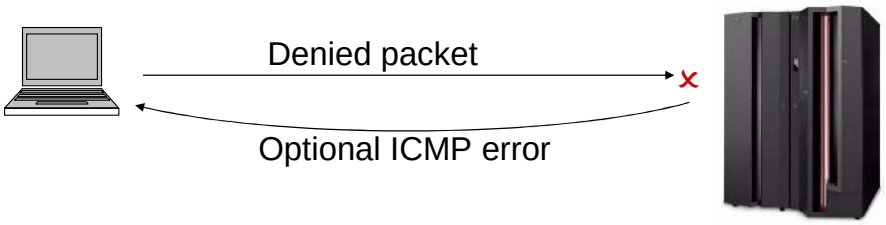


RFC 4301 requires that IPsec implementations allow for the pass-through of the don't-fragment bit and the differentiated services code point (DSCP) value from the inner to the outer IP headers when operating in tunnel mode. For don't-fragment, this allows for control on a per-tunnel basis over the fragmentability of the tunnel traffic. For DSCP, this allows for control on a per-tunnel basis of the DSCP value used for the tunnel. Pass-through for DSCP should only be enabled if the DSCP values are equivalent for the networks represented by the inner and outer IP headers.

The new parameters PassthroughDF and PassthroughDSCP are introduced to the IpManVpnAction and IpDynVpnAction statements to control this behavior. These options are also made available in the z/OS Configuration Assistant GUI. The default is to perform pass-through, which is consistent with the behavior of z/OS Communications Server in previous releases.

DiscardAction

- Silent (no ICMP on discard)
  - Makes your system invisible to attackers
- ICMP
  - Provides helpful diagnostic info to remote systems



RFC 4301 encourages providing support for sending ICMP administratively-prohibited errors when packets are denied by the filter policy. This support is optional; if you choose to disable it (the default), attackers will not receive ICMP errors for their attack packets, which can render your system effectively invisible to attackers. However, if you choose to enable it, it can provide helpful diagnostic information to remote systems, indicating the cause of the packet's denial.

The DiscardAction can be independently controlled on individual deny filter rules and on the implicit deny filter rule created by the Policy Agent. The DiscardAction parameter has choices of Silent or ICMP, and is available on the IpFilterPolicy statement (as ImplicitDiscardAction) and on the IpGenericFilterAction statement. This option is also made available in the z/OS Configuration Assistant GUI.

Discourage use of DES

RFC 4835 discourages the use of the DES encryption algorithm. This is because of its relative weakness compared to other generally available algorithms such as triple DES and AES. The Configuration Assistant and the Policy Agent will issue warnings if DES is configured in the IPsec policy, but they will continue to allow the use of DES without failure.

VPN-A interoperability

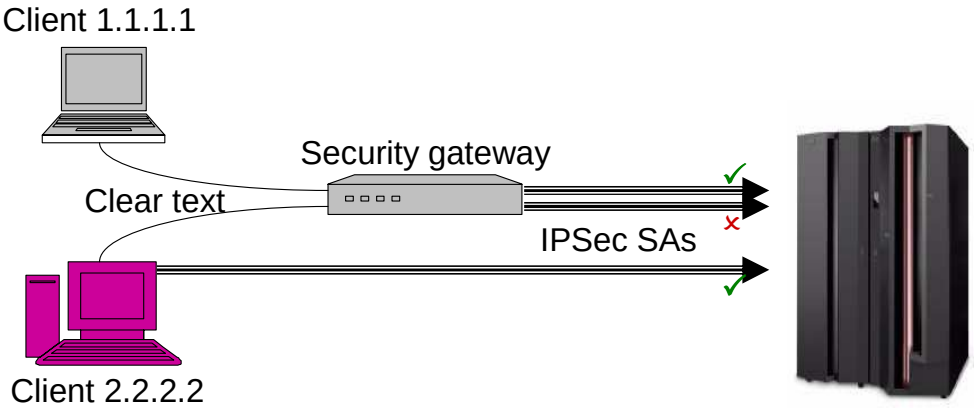
"VPN-A" suite =

	Parameter	Value
✓	Protocol	ESP
✓	Encryption	Triple DES
✓	Integrity	SHA1
✓	Diffie-Hellman	Group 2
✓	P1 Lifetime	24 hours
✓	P2 Lifetime	8 hours

IETF RFCs provide protocol standards for IPsec and IKE interoperability. Configuring IPsec and IKE can still be difficult because there are many configuration choices to be made for both phase 1 and phase 2 of the IKE negotiation. To address this problem, RFC 4308 defines standard naming conventions and meanings for "suites" of security association parameters. These can be used to ensure interoperability between different platforms. The Configuration Assistant defines the Security Level VPN-A to correspond with RFC 4308's VPN-A suite. This chart shows the security level options that correspond to the "VPN-A" suite. The encryption and integrity algorithms indicated apply to both phase 1 and phase 2 of the IKE negotiation.

Address constraints

- Limit the data endpoints for which a gateway can negotiate an SA
  - Other downstream clients can negotiate their own SAs



IPsec key exchange rules are only loosely coupled to IP filter rules. Once an IKE peer has been authorized to negotiate an IKE SA, that IKE peer can negotiate a dynamic tunnel to cover any traffic that the IP filter policy allows to be protected by IPsec. This might then permit the IKE peer to send and possibly receive spoofed traffic for another IKE peer over its own security association. In the picture above, the security gateway is permitted to negotiate an IPSec SA on behalf of client 1.1.1.1 but not on behalf of client 2.2.2.2. Client 2.2.2.2, however, is permitted to negotiate an IPSec SA on behalf of itself.

RFC 4301 requires support for tighter coupling between key exchange rules and dynamic tunnels to prevent this vulnerability. At your option, you can use the new ConstrainSourceAddr and ConstrainDestAddr parameters on the KeyExchangeAction statement to restrict the data endpoints for which an IKE peer can negotiate dynamic tunnels. For example, you might require all tunnels for a gateway *X* to fall only within IP address range *Y*. The z/OS Configuration Assistant GUI also automatically creates address constraints based on the connectivity rules being created.

z/OS Configuration Assistant for Communications Server

tem	Where you configure
Type/code ranges	Traffic descriptor
MIPv6	Traffic descriptor
Opaque	Traffic descriptor
Mobile User (remote identity)	New Connectivity rule type
Manual tunnel SPI changes, wildcard security endpoints	Connectivity rule
Address constraints	Done automatically
Discard action	Stack level for implicit deny Connectivity rule for rule deny
Fragments only	Connectivity rule
Pass-through DSCP, don't fragment	Security level
Discourage use of DES	Security level
VPN-A	IBM supplied security level
Routed traffic restrictions	Throughout (see Configuration Assistant presentation)

All the items discussed here are items are supported by the Configuration Assistant. This slide indicates where in the Configuration Assistant you configure each option.

Trademarks, copyrights, and disclaimers

The following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both:

IBM z/OS

A current list of other IBM trademarks is available on the Web at http://www.ibm.com/legal/copytrade.shtml

Product data has been reviewed for accuracy as of the date of initial publication. Product data is subject to change without notice. This document could include technical inaccuracies or typographical errors. IBM may make improvements or changes in the products or programs described herein at any time without notice. Any statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only. References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business. Any reference to an IBM Program Product in this document is not intended to state or imply that only that program product may be used. Any functionally equivalent program, that does not infringe IBM's intellectual property rights, may be used instead.

Information is provided "AS IS" without warranty of any kind. THE INFORMATION PROVIDED IN THIS DOCUMENT IS DISTRIBUTED "AS IS" WITHOUT ANY WARRANTY, EITHER EXPRESS OR IMPLIED. IBM EXPRESSLY DISCLAIMS ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NONINFRINGEMENT. IBM shall have no responsibility to update this information. IBM products are warranted, if at all, according to the terms and conditions of the agreements (for example, IBM Customer Agreement, Statement of Limited Warranty, International Program License Agreement, etc.) under which they are provided. Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products in connection with this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products.

IBM makes no representations or warranties, express or implied, regarding non-IBM products and services.

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents or copyrights. Inquiries regarding patent or copyright licenses should be made, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. The actual throughput or performance that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput or performance improvements equivalent to the ratios stated here.