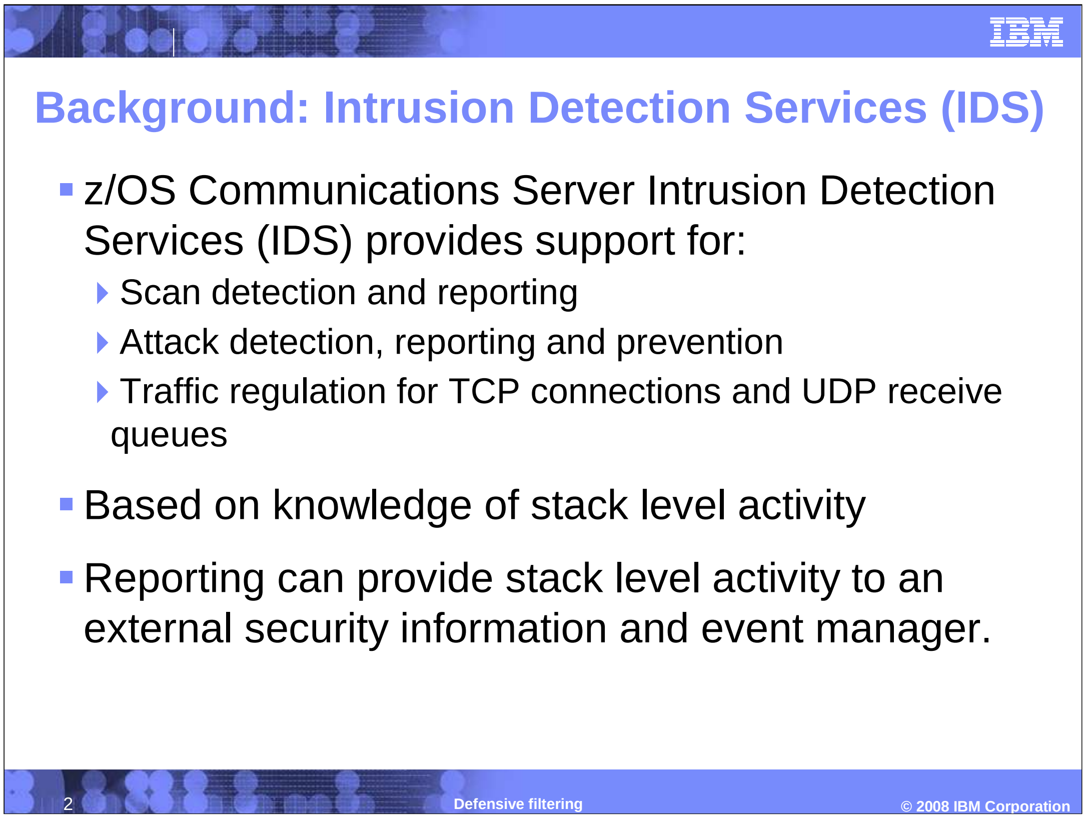
z/OS® V1R10 Communications Server

Defensive filtering

@business on demand software

© 2008 IBM Corporation

This presentation covers defensive filtering.

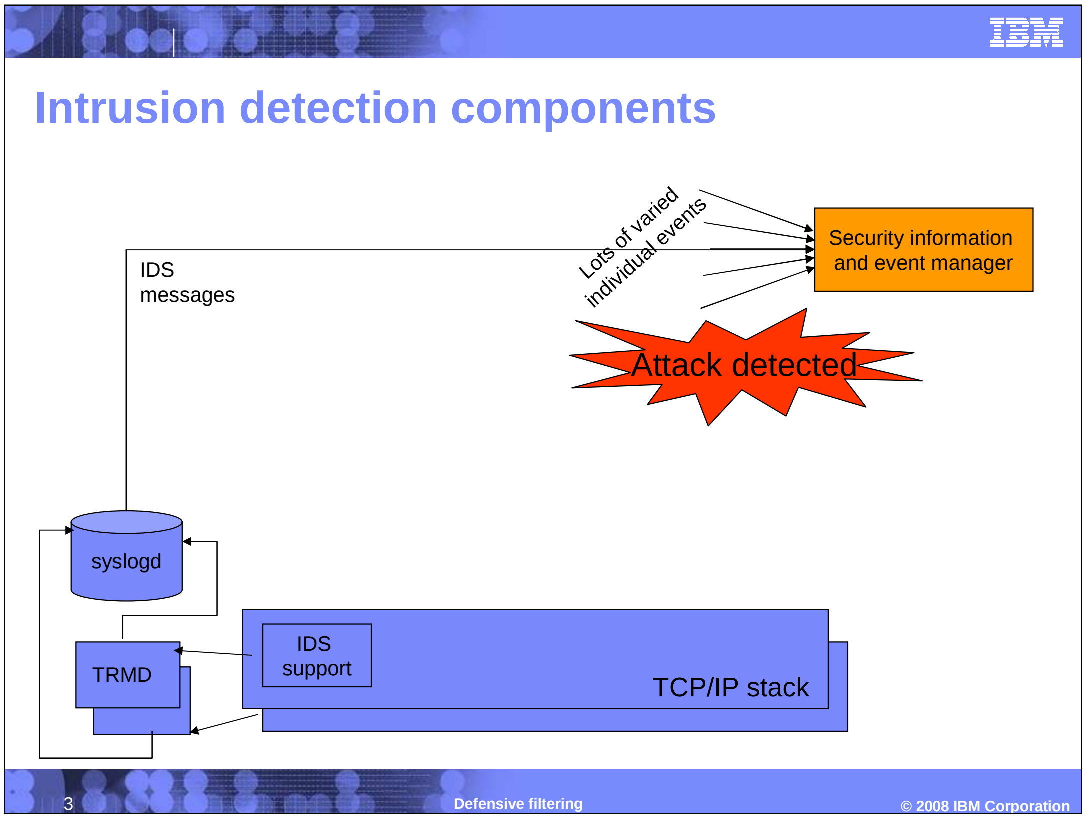


z/OS Communications Server has a wide array of security features.

Intrusion Detection Services (IDS), first introduced in V1R2, can be enabled to detect attacks and scans . It also includes traffic regulation support for TCP connections and UDP receive queues.

IDS prevention measures include the ability to drop attack packets and to limit TCP and UDP traffic by dropping packets that would exceed the configured connection limit or receive queue length.

Reporting is also a key piece of the TCP/IP stack's IDS support. z/OS Communication Server's IDS support provides protection based on stack level knowledge, in other words activity in **that** stack. However, by generating messages that can be retrieved by an external security information and event manager, a TCP/IP's stack messages can be correlated with messages and events from other parts of the network. The external manager can analyze information from across the network and take the appropriate action.



In this picture you can see that an external security information and event manager is receiving varied inputs from several sources including z/OS Communications Server IDS messages. The external security information and event manager correlates the inputs and analyzes the information. If an attack is detected, what happens next?

One possibility is that an operator is alerted and the course of action to take is to block traffic on one or more stacks.

Note that in this picture IDS messages are included as an input. However, an action to block traffic on a stack does not need to be directly related to IDS messages generated by the stack. In fact, IDS may not be running on that stack at all.

Note that this picture shows an external security information and event manager. However, you might have any number of attack detection methods deployed, from manual inspection of messages to various levels of automation.

IP security filtering

- An administrator can update IP security policy to deny or block a traffic pattern associated with the attack.
  - GUI update or
  - direct edit of the policy file

 © 2008 IBM Corporation

IP filtering, a part of the IP security function, can be used to block traffic on a z/OS Communications Server stack. An IP security filter rule can be configured that denies or blocks traffic.

Based on the information from the external security information and event management application, an administrator can update IP security policy to deny a traffic pattern associated with the attack. The IP security policy can be updated through the GUI or by a direct edit of the policy file. With a GUI update, the administrator makes the changes to their existing IP security policy, uploads the file or files to z/OS, and Policy Agent reads and installs the new policy.

Limitations of IP security to block attacks

- Mismatch between attack detection and blocking mechanism
  - Attack detection can be **automated**
  - **Manual** action required to configure deny filter
- Mismatch between attack characteristics and blocking mechanism
  - **Short** term event
  - Updating **comprehensive**, **long-term** IP security policy

© 2008 IBM Corporation

There is a mismatch between the attack detection and the blocking mechanism.

Attack detection can be an automated process. An external security information and event management application can detect an attack based on input from a variety of sources in the network. Or a CLIST can be implemented that detects an attack based on messages that it monitors.

Configuring an IP security filter to block or deny a traffic pattern associated with the attack requires a manual action by the administrator of the IP security policy file. The administrator must update the IP security policy either through the GUI or by a direct edit of the policy file.

Also, an attack typically is a short term event lasting several minutes or hours.

The IP security policy is a comprehensive policy that covers all traffic. Filter rules define which traffic to permit, which traffic to deny, and which traffic to protect with authentication and encryption. All traffic is covered.

An IP security filter to block or deny the attack can be configured to have a short duration by using the IpTimeCondition statement. However, if the filter's lifetime needs to be extended, the policy file must be updated again. After the time condition expires the filter is no longer installed in the stack but it remains in the policy file until it is manually removed.

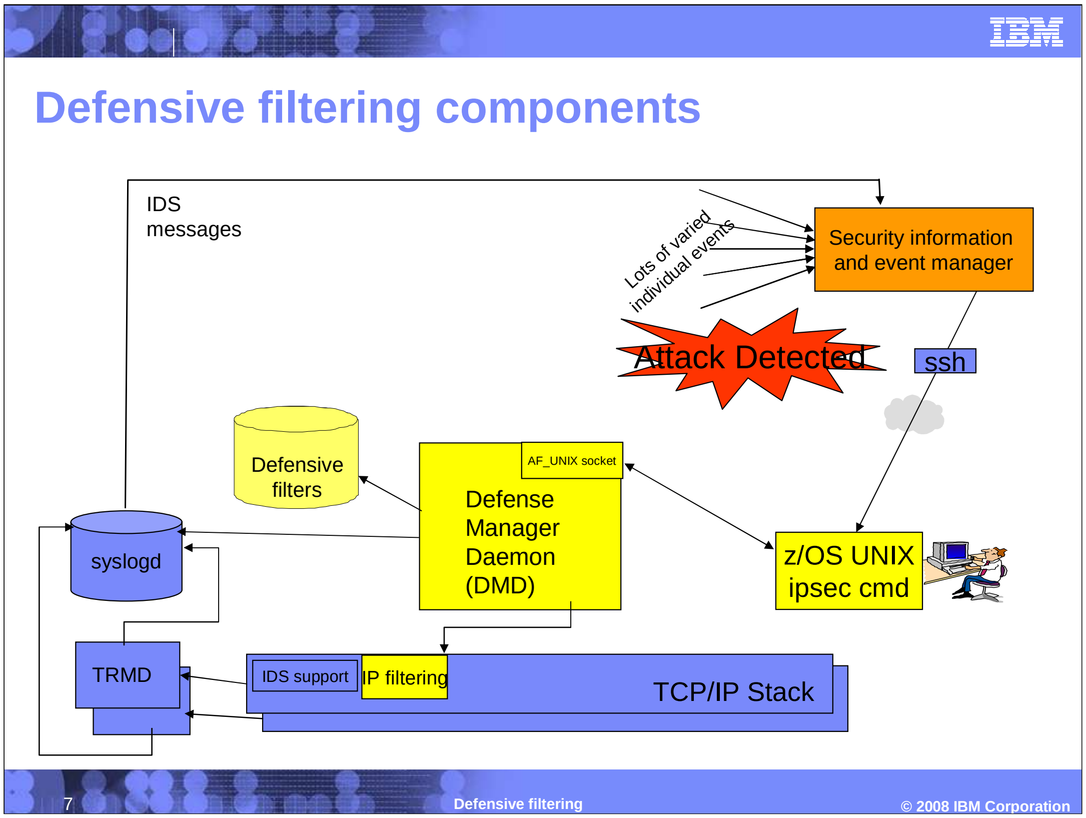
So while IP security filters provide a mechanism to block traffic in a stack, the mechanism is cumbersome to implement in response to a detected attack.

New for V1R10: Defensive Filtering

- Defensive filtering provides a command based approach to install short-term filters. These filters will block a traffic pattern identified as an attack.
  - Command can be issued by a security information and event manager with a wider view than a single TCP/IP stack or z/OS image.
  - Action can be part of a coordinated effort to protect the network.

6 Defensive filtering © 2008 IBM Corporation

Defensive filtering resolves the mismatch by providing a command based approach for installing short-term filters to block traffic. This approach allows an external security information and event manager with a wider view than a single TCP/IP stack or z/OS image (LPAR) to issue a command to this stack. This is done as part of a coordinated effort to protect the network.



This picture shows the z/OS V1R10 Communications Server components involved in defensive filtering. The three main components are the ipsec command, a new daemon - the Defense Manager daemon, and the TCP/IP stack's IP filtering. This presentation will cover these components in greater detail.

Note that an operator or administrator can issue the ipsec command from a shell environment, in addition to the command being issued by an external security information and event manager.

Defensive filtering: Main components

- z/OS UNIX® ipsec command
- Defense Manager daemon (DMD)
- TCP/IP stack filtering

 © 2008 IBM Corporation

The **z/OS UNIX ipsec command** is an existing command that is used to display and modify IP security information. For example, the ipsec command can be used to display IP security filters and tunnels.

Defensive filtering provides a new option (-F) on the ipsec command that you can use to add, update, delete and display defensive filters.

The ipsec command can be issued manually or by automation. A user on the z/OS image (LPAR) can issue the ipsec command or an external user can establish a secure login environment (for example, through Secure Shell (ssh)) and issue the ipsec command.

The ipsec command can only be issued by an authorized user. The user must be logged in under a SAF user ID that has READ access to the appropriate SERVAUTH profiles. You will see more information about the SERVAUTH profiles shortly.

The **Defense Manager daemon or DMD** is a new daemon that is the control point for defensive filters on a z/OS image (LPAR). It interacts with the TCP/IP stack to manage defensive filters. For example, an ipsec command request to add a defensive filter to stack TCPCS is processed by the DMD which will install the defensive filter in the stack. The DMD also reinstalls active defensive filters in the stack when a stack goes down and is restarted.

The DMD maintains a file for each stack with a copy of the stack's active defensive filters. These files are persistent. If the DMD goes down and is restarted, the DMD reads the files on start so that it is in sync with defensive filters installed in the stacks.

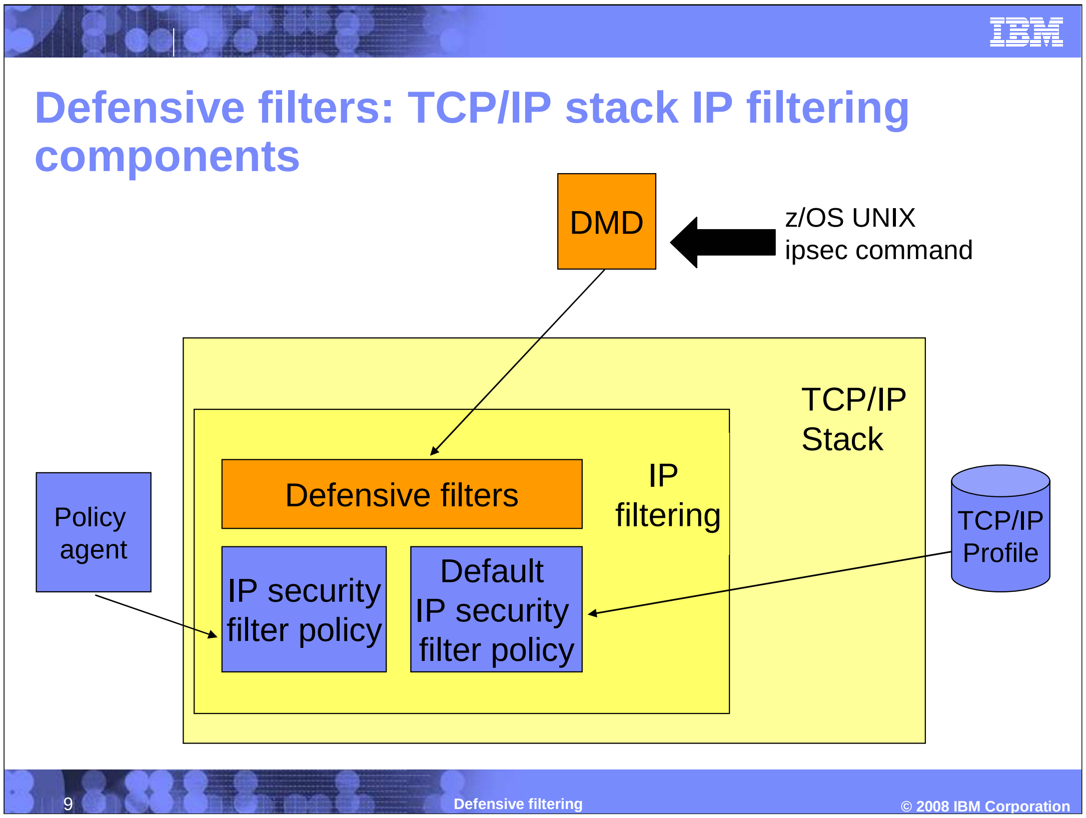
An instance of the DMD is required for each z/OS image where you want to implement defensive filtering. If you use multiple TCP/IP stacks in a CINET environment, one instance of the DMD will manage defensive filtering for all stacks on the z/OS image (LPAR).

Also modified is **TCP/IP stack filtering**. Traffic that matches a defensive filter is denied, in a manner similar to an IP security deny filter.

If IP security is enabled for a stack, IP filtering will now check traffic against defensive filters in addition to IP security filters. If there are defensive filters installed in the stack, they are searched before IP security filters. There is no additional TCP/IP stack configuration to enable defensive filtering.

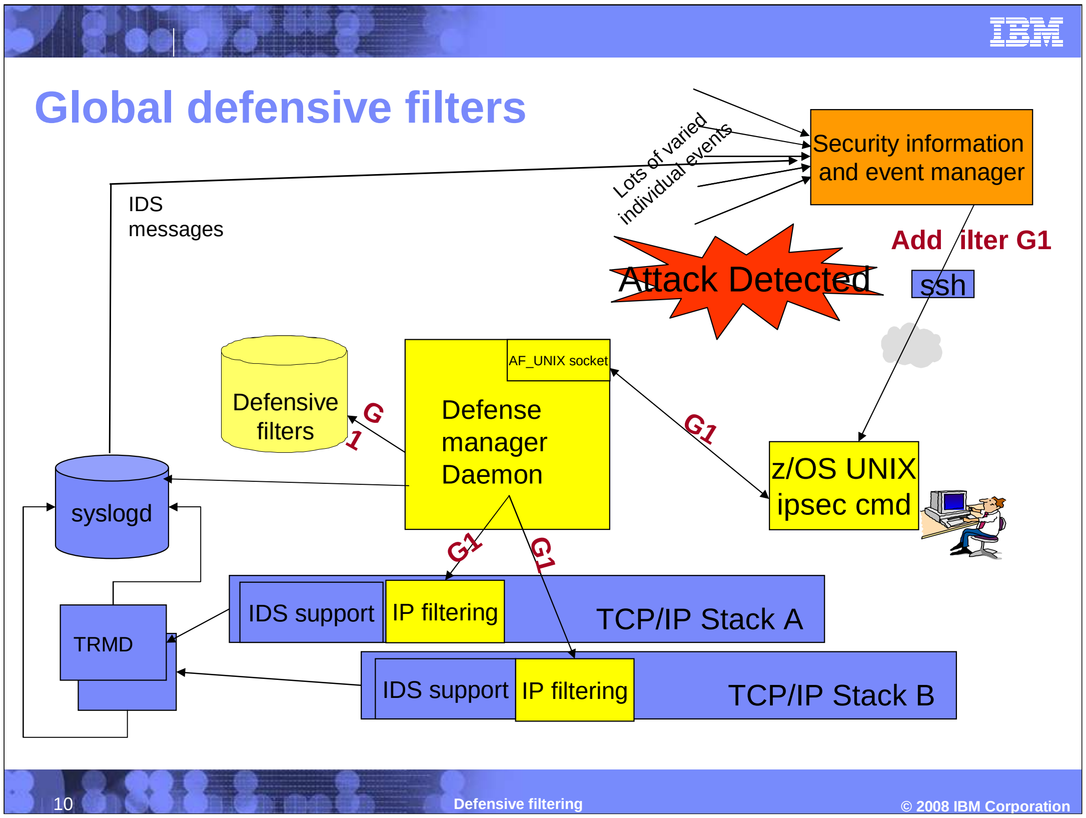
The search order for IP security filters is determined by their order in the configuration file or the GUI rule panel. When a defensive filter is installed in the stack, it is added to the top of the search order.

As with IP security filters, when a packet matches a defensive filter a log message can be generated.



In this picture you can see that defensive filters are processed by the stack's existing IP filtering component. IP filters can be installed in three ways. Policy Agent installs IP security filters that are configured in the Policy Agent flat file, or by way of the Configuration Assistant. Default IP security filters are configured in the TCP/IP profile and installed when the stack is started or when a VARY OBEY command is processed. When IP security is enabled by coding IPSECURITY on the IPCONFIG statement, one of these policies is in effect.

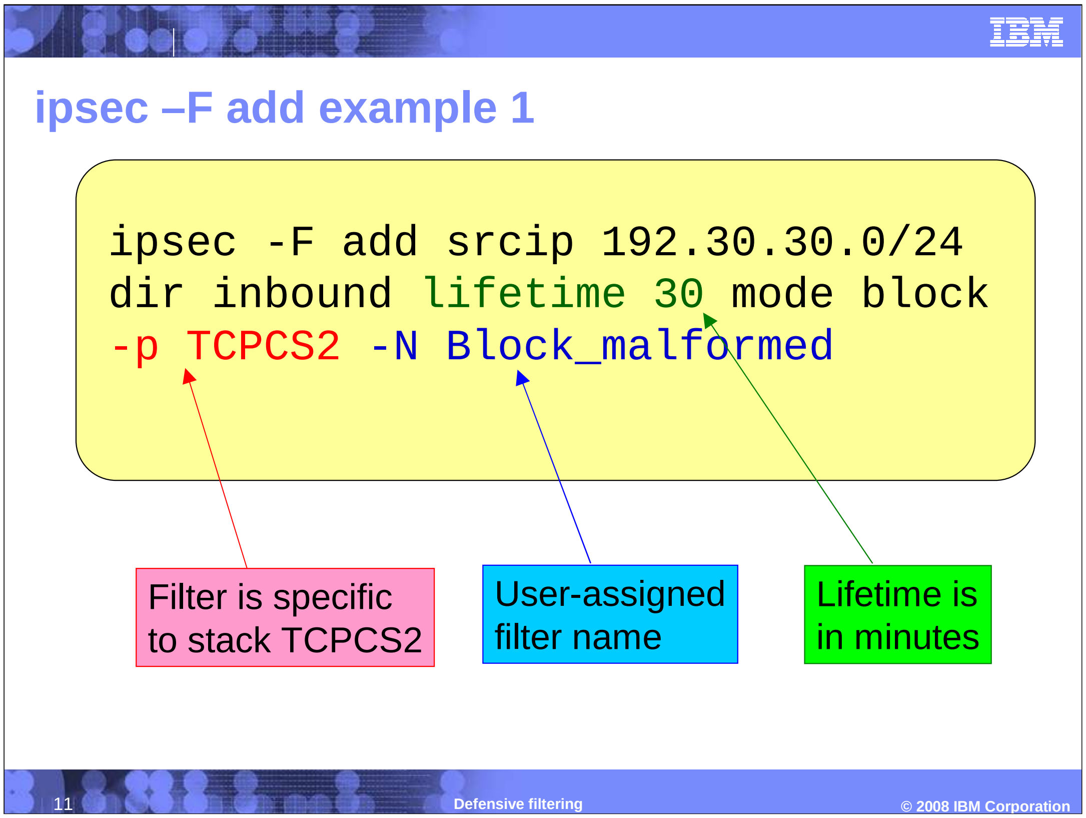
The third, new way that IP filters can be installed is through the DMD based on a z/OS UNIX ipsec command request. When there are defensive filters installed, the defensive filters are checked first before the IP security filters. This is true without regard to which IP security policy is in effect.



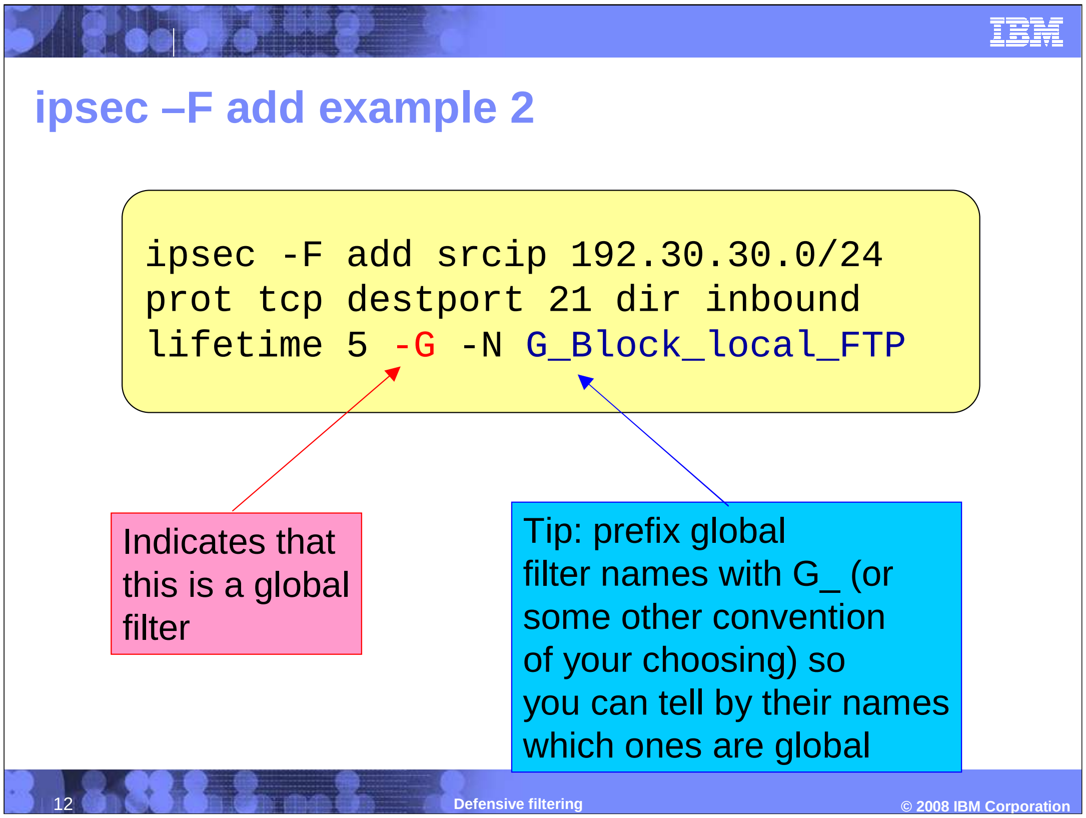
In a Common INET environment with multiple TCP/IP stacks the ipsec command can be used to add a global filter. A global filter is added to all active stacks on the z/OS image that are enabled for defensive filtering. It will also be added to a stack that comes up later if it is enabled for defensive filtering.

This means that one command can be issued to protect all stacks on the z/OS image!

In this picture a global filter, G1, is added by a security information and event manager by issuing one z/OS UNIX ipsec command. The request is transmitted to the DMD which installs filter “G1” in both TCP/IP stack A and B. The DMD also writes a copy of the filter to its persistent file storage. If a third TCP/IP stack was brought up before filter “G1” expired, the DMD would add the filter to the new stack as well.



Example 1 adds a stack specific defensive filter to TCPCS2 with a name of Block_malformed. The filter will block inbound traffic from subnet 192.30.30.0/24. It will remain installed in the stack for 30 minutes.



Example 2 adds a global defensive filter with a name of G_Block_local_FTP that will block inbound TCP traffic to port 21 from subnet 192.30.30.0/24 for five minutes. The Defense Manager daemon (DMD) maintains a copy of the global filter and generates a copy that is installed in each local TCPIP stack for which defensive filtering is enabled.

Defensive filter names

- Global filters and stack-specific filters share the same defensive filter namespace.
- **Tip:** If you are manually creating defensive filters choose a distinct naming convention for global filters. For example, start all global filter names with a “G”.

13 Defensive filtering © 2008 IBM Corporation

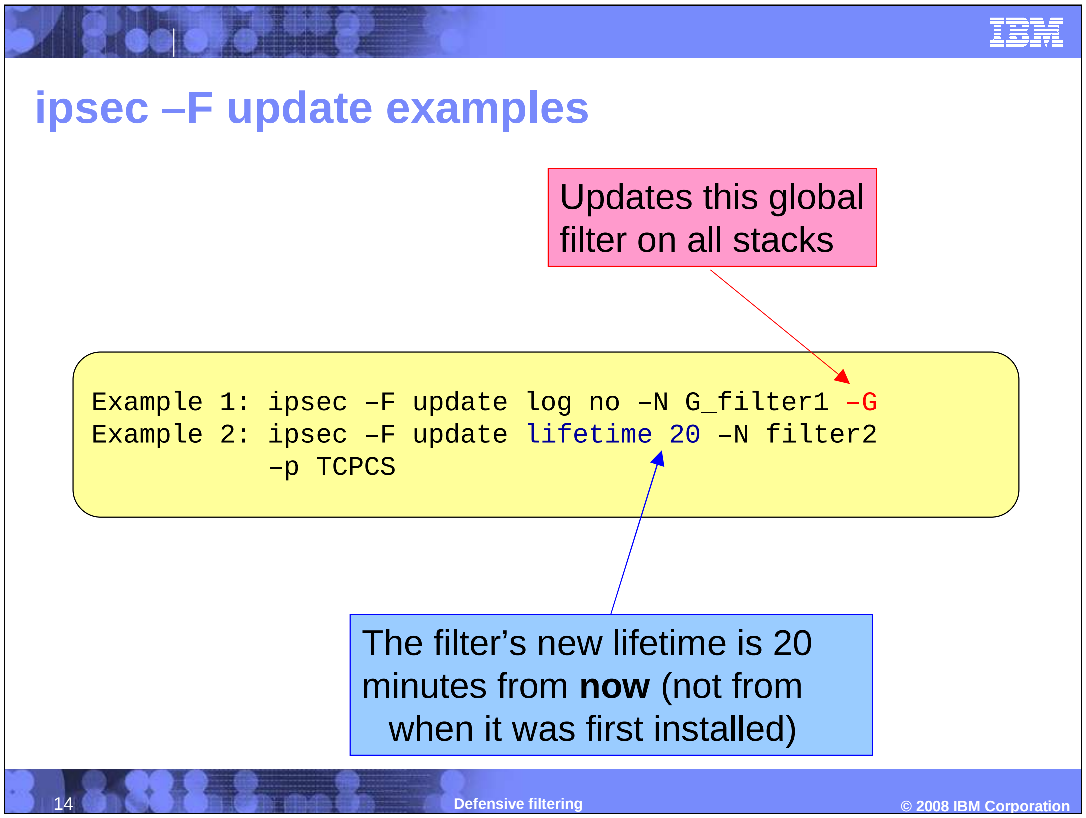
Global filters and stack-specific filters share the same defensive filter namespace. A filter name cannot be used for both a global filter and a stack-specific filter.

A stack-specific filter add will fail if the filter name is already used for a global filter or for a stack-specific filter in the target stack.

A global filter add will fail if there is already a global filter with that name or if there is already a stack-specific filter with that name in any stack.

The shared namespace allows a global filter to be updated or deleted for a specific stack, and on a global basis.

Consider using a unique naming convention for global filters.



A defensive filter can be updated with the ipsec –F update command. The characteristics that describe the traffic pattern, such as srcip, cannot be changed. The mode of the filter (block or simulate) can be changed, as can the log setting (yes or no).

Also, the lifetime of the filter can be changed, either extending or shortening the lifetime. For example, if a filter is added with the default lifetime of 30 minutes, after 20 minutes pass, the remaining lifetime is 10 minutes. If you issue **ipsec –F update lifetime 5**, the filter's remaining lifetime is shortened to five minutes. If you issue **ipsec –F update lifetime 15** instead, the filter's remaining lifetime is lengthened to 15 minutes. The filter's remaining lifetime is set to the number of minutes specified on the update.

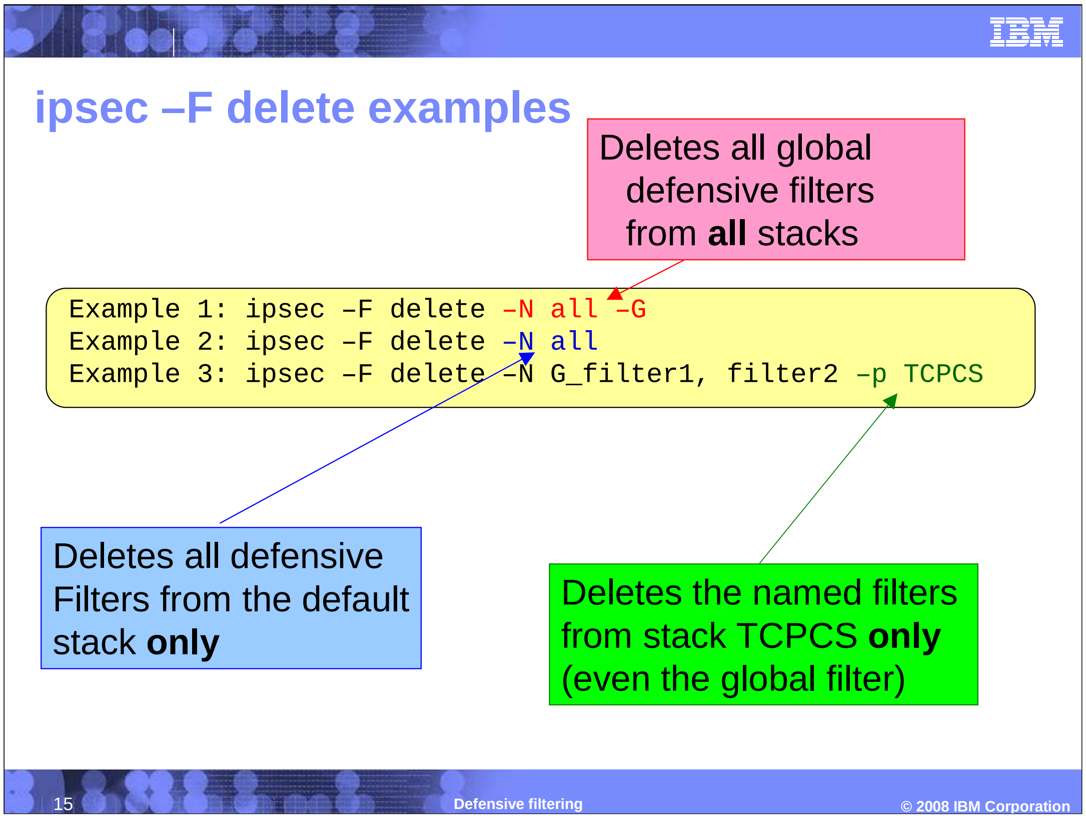
The –G or –p option determines which copies of a filter are updated. If neither –G or –p is specified, the update request is directed to the default stack.

The -G option indicates that all copies of a global defensive filter should be updated.

The –p option indicates that the copy of the defensive filter in the specified stack should be updated. Note that this filter might have been created as a stack-specific filter and this is the only copy. Or this filter might have been created as a global filter and only this copy is being updated.

In the first example, global defensive filter "G_filter1" is being updated to turn logging off. All copies of "G_filter1" are being updated.

In the second example, the lifetime of defensive filter "filter2" is being set to 20 minutes. The copy of "filter2" installed in TCPCS is updated.



One or more defensive filters can be deleted with the ipsec –F delete command.

The names specified on the –N option identify the filters to be deleted. One or more defensive filter names can be specified on the –N option. -N all can also be specified to indicate that all defensive filters should be deleted.

The –G or –p option determines which copies of a filter are deleted. If neither –G or –p is specified, the delete request is directed to the default stack.

The -G option indicates that all copies of the identified global defensive filters should be deleted. "ipsec –F delete –N all –G" will result in all copies of all global filters being deleted from the z/OS image (LPAR).

The –p option indicates that the copy of the identified defensive filters in the specified stack should be deleted. Note that this filter might have been created as a stack-specific filter and this is the only copy. Or this filter might have been created as a global filter and only the copy in this stack is being deleted. Note that if a copy of a global filter is deleted from a stack that it is not added back to that stack even if the stack goes down and is brought back up.

Example 1 will result in all global defensive filters being deleted from all stacks on the z/OS image.

Example 2 will result in all defensive filters being deleted from the default stack. This includes both global and stack-specific filters.

Example 3 will result in G_filter1 and filter2 being deleted from stack TCPCS. Even though G_filter1 is global, it will only be deleted from TCPCS and not modified on any other stacks.

ipsec –F display example (part 1)

- **Display of defensive filter G_Block_local_FTP from TCPCS2**

```
USER1@MVS118:/u/user1:> ipsec -F display -p TCPCS2 -N G_Block_local_FTP

CS V1R10 ipsec  Stack Name: TCPCS2  Sun Jan 27 17:57:36 2008
Primary:  Defensive Filt  Function: Display          Format:   Detail
Source:   Stack           Scope:    n/a              TotAvail: 11
Logging:  n/a             Predecap: n/a              DVIPSec:  n/a
NatKeepAlive:  0
Defensive Mode: Active
Exclusion Address: 9.42.0.0/16

FilterName:                  G_Block_local_FTP
FilterNameExtension:         1
GroupName:                   n/a
LocalStartActionName:        n/a
VpnActionName:               n/a
TunnelID:                    n/a
Type:                        Defensive
DefensiveType:               Global
State:                       Active
Action:                      Defensive Block
...
```

16 Defensive filtering © 2008 IBM Corporation

In this example you see a display of the defensive filter G_Block_local_FTP that was added in the earlier ipsec –F add example. This is a global filter that was added to both active stacks, TCPCS2 and TCPCS3. This display is directed to stack TCPCS2 with the –p option. The display shows the filter as it is installed in TCPCS2.

Some fields to note are:

Type: Defensive indicates the filter type is defensive.

DefensiveType: Global indicates that the filter was created as a global filter with the –G option. A value of **stack** indicates that the filter was created as a stack-specific filter

Action: Defensive Block indicates that the filter was created with a mode of block. A value of **Defensive Simulate** indicates that the filter was created with a mode of simulate.

ipsec –F display example (part 2)

```
Scope:                          Local
Direction:                      Inbound
OnDemand:                       n/a
SecurityClass:                  0
Logging:                        All
Protocol:                       TCP(6)
ICMPType:                       n/a
ICMPCode:                       n/a
OSPFType:                       n/a
TCPQualifier:                   None
ProtocolGranularity:            Packet
SourceAddress:                  192.30.30.0
SourceAddressPrefix:            24
SourceAddressRange:             n/a
SourceAddressGranularity:       Packet
SourcePort:                     All
SourcePortRange:                n/a
SourcePortGranularity:          Packet
DestAddress:                    0.0.0.0
DestAddressPrefix:              0
DestAddressRange:               n/a
```


© 2008 IBM Corporation

This is a continuation of the display output for **ipsec -F display -p TCPCS2 -N G_Block_local_FTP**.

You can see the characteristics of the filter that was defined: inbound, TCP, source IP address 192.30.30.0/24

ipsec –F display example (part 3)

```
DestAddressGranularity:       Packet
DestPort:                     21
DestPortRange:                n/a
DestPortGranularity:          Packet
OrigRmtConnPort:              n/a
RmtIDPayload:                 n/a
RmtUdpEncapPort:              n/a
CreateTime:                   2008/01/27 17:56:53
UpdateTime:                   2008/01/27 17:56:53
DiscardAction:                Silent
MIPv6Type:                    n/a
TypeRange:                    n/a
CodeRange:                    n/a
RemoteIdentityType:           n/a
RemoteIdentity:               n/a
FragmentsOnly:                No
FilterMatches:                5
LifetimeExpires:              2008/01/27 18:01:53
AssociatedStackCount:         n/a
*************************************************************
```

18 Defensive filtering © 2008 IBM Corporation

This is a continuation of the display output for **ipsec -F display -p TCPCS2 -N G_Block_local_FTP**.

You can see the characteristics of the filter that was defined: destination port 21.

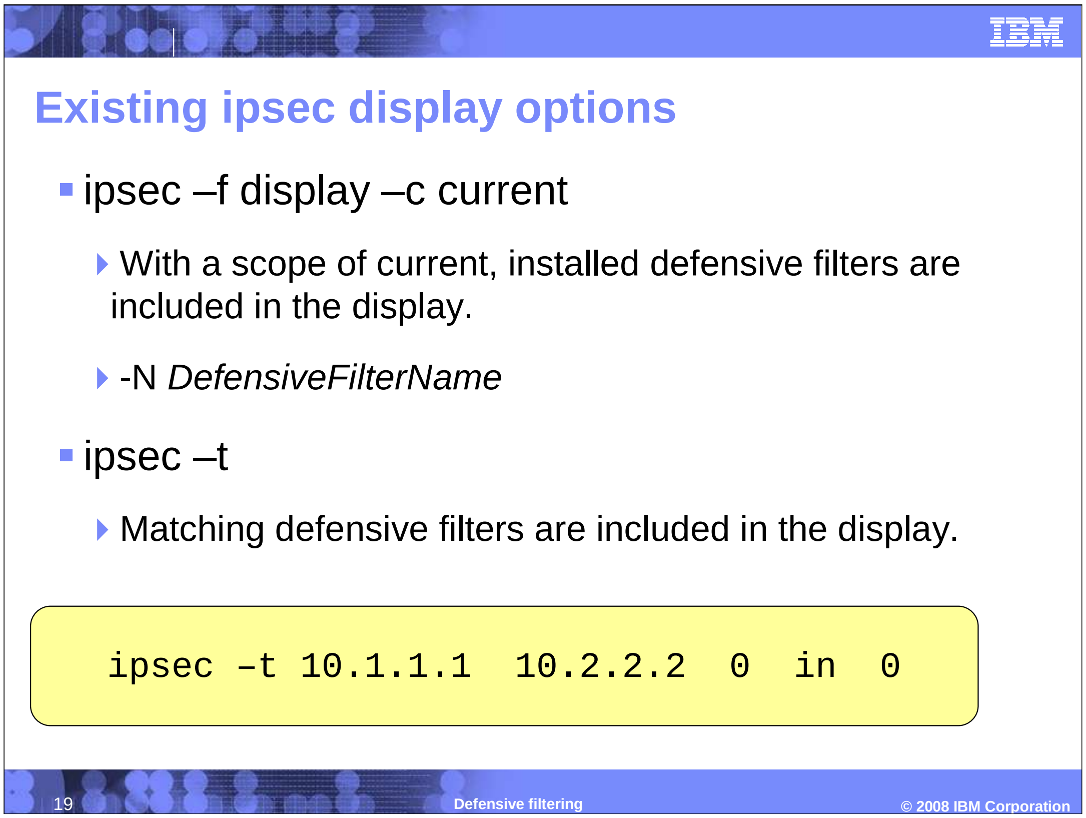
Some fields to note:

CreateTime is the time that the filter was created.

UpdateTime is the time that the filter was last updated. The update time is initially equal to the CreateTime.

FilterMatches indicates the number of packets that have matched this filter.

LifetimeExpires is the time that this filter will expire and be deleted.



The existing filter display command (ipsec –f display) has been updated. When the scope is –c current, both defensive filters and IP security filters are displayed.

The –N option can be used to limit the number of filters displayed to the defensive filters identified on –N.

The –n option continues to be used with IP security filter names.

The existing traffic test command (ipsec –t) has been updated. The traffic test command takes an input traffic pattern and returns the IP filters that match the traffic pattern. Both matching defensive filters and IP security filters are displayed.

In the example, a traffic test is being run to find out what filter rules an inbound packet with a source IP address of 10.1.1.1 and destination IP address of 10.2.2.2 matches. The output can contain defensive filters, in addition to IP security filters. Note that the first 0 indicates any protocol and the second 0 indicates any security class.

ipsec command access control

command access controlled by profiles in the SERVAUTH class

Resource name	ipsec options allowed
EZB.IPSECCMD.sysname.stackname.DISPLAY (existing profile)	ipsec –F display
EZB.IPSECCMD.sysname.stackname.CONTROL (existing profile)	ipsec –F add ipsec –F update ipsec –F delete
EZB.IPSECCMD.sysname.DMD_GLOBAL.DISPLAY	ipsec –F display –G
EZB.IPSECCMD.sysname.DMD_GLOBAL.CONTROL	ipsec –F add –G ipsec –F update –G ipsec –F delete -G

© 2008 IBM Corporation

Access to the z/OS UNIX ipsec command is controlled by profiles in the SERVAUTH class.

To display defensive filters the user ID issuing the command must have READ access to the existing EZB.IPSECCMD.sysname.stackname.DISPLAY profile.

To add, update, and delete defensive filters the user must have READ access to the existing EZB.IPSECCMD.sysname.stackname.CONTROL profile.

New profiles must be defined if you want to allow a user to create and manage global defensive filters. For global filters a special form of the stack name is used in the profile: DMD_GLOBAL.

To display global defensive filters the user ID must have READ access to the new EZB.IPSECCMD.sysname.DMD_GLOBAL.DISPLAY profile.

To add, update, and delete global filters with the –G option the user ID must have READ access to the new EZB.IPSECCMD.sysname.DMD_GLOBAL.CONTROL profile.

if you have a profile defined that uses generics to give access to the ipsec command for all stacks , no additional profiles are needed to allow the user ID to issue the ipsec –F command with the –G option. This profile would be (EZB.IPSECCMD.sysname.*.*)

ipsec –z option

- -z *clientname* directs the NSS server to obtain information from the NSS client
- Not supported for new –F option
- Existing ipsec –f display –c current –z *clientname* supported

21 Defensive filtering © 2008 IBM Corporation

The existing –z *clientname* option directs the NSS server to send the ipsec request to the specified NSS client. The –z option is not supported with the new –F option. So for example, ipsec –F display –z client4 is rejected, as will any other ipsec –F request.

Note: The NssStackConfig statement in the IKE daemon configuration file defines the *clientname* that corresponds to a TCP/IP stack.

The –z option is supported for the existing ipsec –f requests. So an ipsec –f display –c current –z client4 is sent to the NSS client. The NSS client will return any installed defensive filters in addition to IP security filters. If the ipsec command is being issued from a V1R10 system, the remote display will contain the same information as the local display. If, however, the ipsec command is being issued from a V1R9 system, the remote display will not include new filter display fields. Also, the filter type for a defensive filter is displayed as Unknown on the remote display.

IBM

Defense manager daemon (DMD)

- DMD configuration file
- (optional) DMD environment variables
- Defining the DMD user ID to the external security manager
- Starting and stopping the DMD
- MODIFY command

22 Defensive filtering © 2008 IBM Corporation

The defense manager daemon (DMD) is responsible for installing and maintaining defensive filters. This presentation covers at a high level setting up and configuring this daemon. This presentation will cover the configuration file, environment variables, external security manager requirements, starting and stopping the daemon, and controlling it with the MODIFY command.

DMD configuration file example

```
DmConfig
  {
    SyslogLevel 8
    DefensiveFilterDirectory /var/dm/filters
  }
  DmStackConfig TCPCS
  {
    Mode Simulate
    MaxLifetime 2000
    Exclude 10.1.1.0/24
  }
```

 © 2008 IBM Corporation

There are two types of statements in the DMD configuration file. The DmConfig statement controls the overall operation of the DM daemon The DmStackConfig statement provides parameters and information about the stacks to be served by the daemon.

See *z/OS Communications Server IP Configuration Reference* for more details on any configuration statement

On the **DmConfig** statement the **SyslogLevel** parameter defines the level of DMD logging in the system log. The **DefensiveFilterDirectory** parameter specifies where the DMD will keep its database of filters (more on this in the next slides)

There is a **DmStackConfig** statement for each stack that will receive defensive filters. In this example, stack TCPCS is being configured.

Mode can have several values. **Active** indicates that defensive filters will block traffic on this stack. **Inactive** indicates that defensive filters will not be installed to this stack. **Simulate** indicates that defensive filters are installed but will not block traffic but will issue messages indicating that traffic would have been blocked.

MaxLifetime limits the lifetime of defensive filters on this stack, in minutes. If defensive filters are installed with a longer lifetime, their lifetime is truncated to this value.

As part of creating your IP security policy, you typically define a permit rule near the top of your filter policy to allow administrative access to the stack. However, defensive filters are checked before IP security filters. To ensure that an administrator is not inadvertently blocked by a defensive filter, you can exclude the administrator's IP address from defensive filter processing. You do this by specifying the administrator's address on an **Exclude** parameter on the DmStackConfig statement. Up to 10 IP addresses or subnets can be excluded.

Inbound packets originating from an IP address in the exclusion list are excluded from defensive filter processing. Outbound packets destined to an IP address in the exclusion list are excluded from defensive filter processing.

DMD details: DefensiveFilterDirectory

DefensiveFilterDirectory parameter

- Specifies the directory where DMD should create a file per stack with a copy of the stack's active defensive filters.
- **Important**: Defensive filter files are binary files managed by DMD.
  - Should **not** be manually opened or modified.
  - Files are persistent: if DMD is stopped and restarted, the files are read by DMD to establish active defensive filters.

© 2008 IBM Corporation

The DefensiveFilterDirectory parameter specifies the directory where the DMD should create a file for each stack with a copy of the stack's active defensive filters. The directory must exist when the DMD is started and the DMD must have the authority to create, delete, read and write files in the directory. The default directory is /var/dm/filters .

The DefensiveFilterDirectory parameter cannot be changed with the MODIFY REFRESH command. If you need to change the directory, DMD must be stopped and restarted with the new value.

Each stack with active defensive filters has a file in the directory with a file name in the form active.*stackname.* If there are active global defensive filters there is also a file with the file name active._globals_ .

Important: Defensive filter files are binary files managed by DMD. They should **not** be manually opened or modified.

Mode on ipsec command versus mode on DmStackConfig

	DmStackConfig Mode=Active	DmStackConfig Mode=Simulate	DmStackConfig Mode=Inactive
Filter setting = block	Block the packet	Simulate blocking the packet	No defensive filters
Filter setting = simulate	Simulate blocking the packet	Simulate blocking the packet	No defensive filters

Simulate block = "EZD1722I Packet would have been denied by defensive filter" logged and IP filtering continues

© 2008 IBM Corporation

You have seen mode as a parameter on the ipsec command and as a parameter on the DmStackConfig in the DMD configuration file. With this chart you can look at how the two mode settings interact. A defensive filter's mode is set when the filter is created or updated by the ipsec command. The DMD configuration file also provides a mode setting of Active, Simulate, or Inactive on the DmStackConfig statement.

A mode of **Active** enables defensive filtering for a stack and honors the mode setting of the individual filters. On the chart you see that when Mode Active is set on the DmStackConfig statement that a defensive filter with mode block will block a matching packet. A defensive filter with mode simulate will only simulate a block.

A mode of **Simulate** on the DmStackConfig statement enables defensive filtering and overrides the mode setting of the individual filters. On the chart you see that when Mode Simulate is set on the DmStackConfig statement that all defensive filters will simulate a block.   This is regardless of the filter's individual mode setting.

A mode of **Inactive** on the DmStackConfig statement disables defensive filtering. There are no defensive filters in the stack when Mode is Inactive.

Simulate block means that this message is logged: "EZD1722I Packet would have been denied by defensive filter" and then IP filtering continues. Simulate mode is typically used in a test environment.

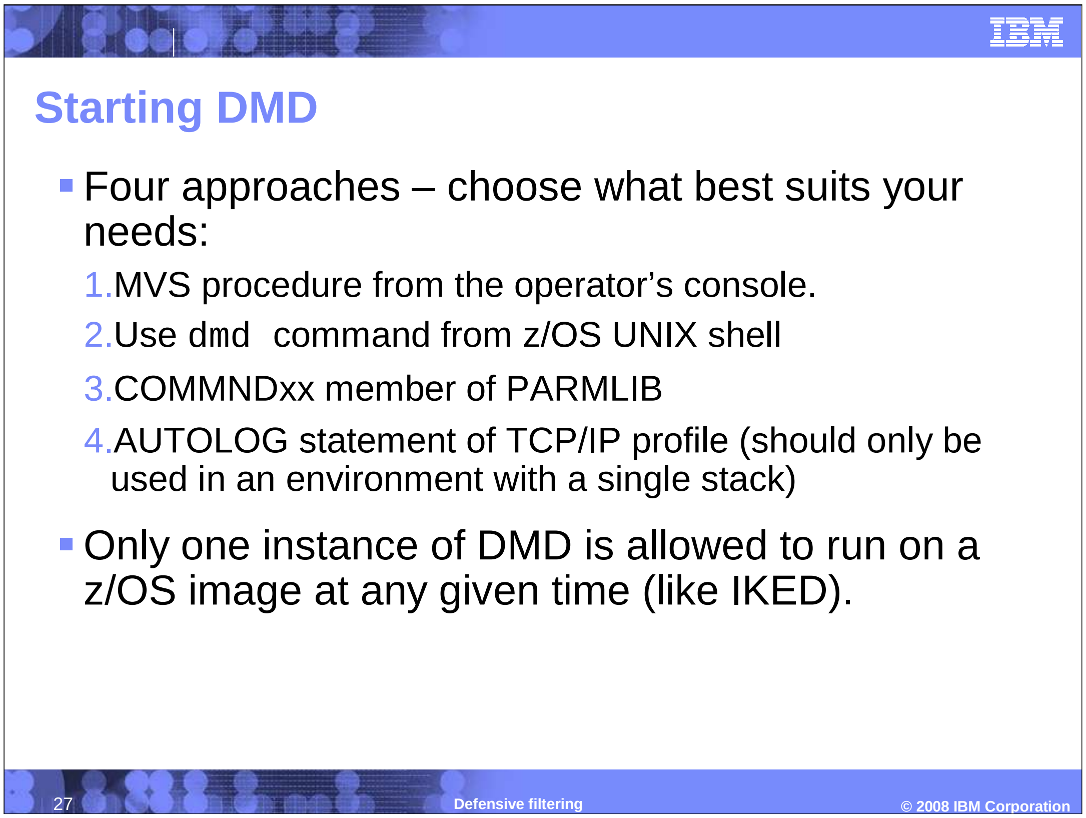
DMD authorization

- DMD is authorized just like any other z/OS UNIX daemon:
  - Define DMD user ID to external security manager
  - Authorize DMD user ID to the STARTED class (if started from MVS started procedure)
  - Permit DMD user ID to SYS1.PARMLIB
    - Default DMD component trace parmlib member is stored in SYS1.PARMLIB

26 Defensive filtering © 2008 IBM Corporation

The DMD is a z/OS UNIX application that you can start from the z/OS UNIX shell or from an MVS started procedure. Before starting the DMD, you must define the DMD user ID to the external security manager. If you start the DMD from an MVS started procedure, the DMD user ID must also be authorized to the STARTED class.

Permit the DMD user ID to SYS1.PARMLIB. The DMD uses the TCP/IP component trace (CTRACE) to perform service-level tracing. The default DMD component trace parmlib member is stored in SYS1.PARMLIB. The DMD user ID must be permitted to access SYS1.PARMLIB.



The DMD can be started in four ways.

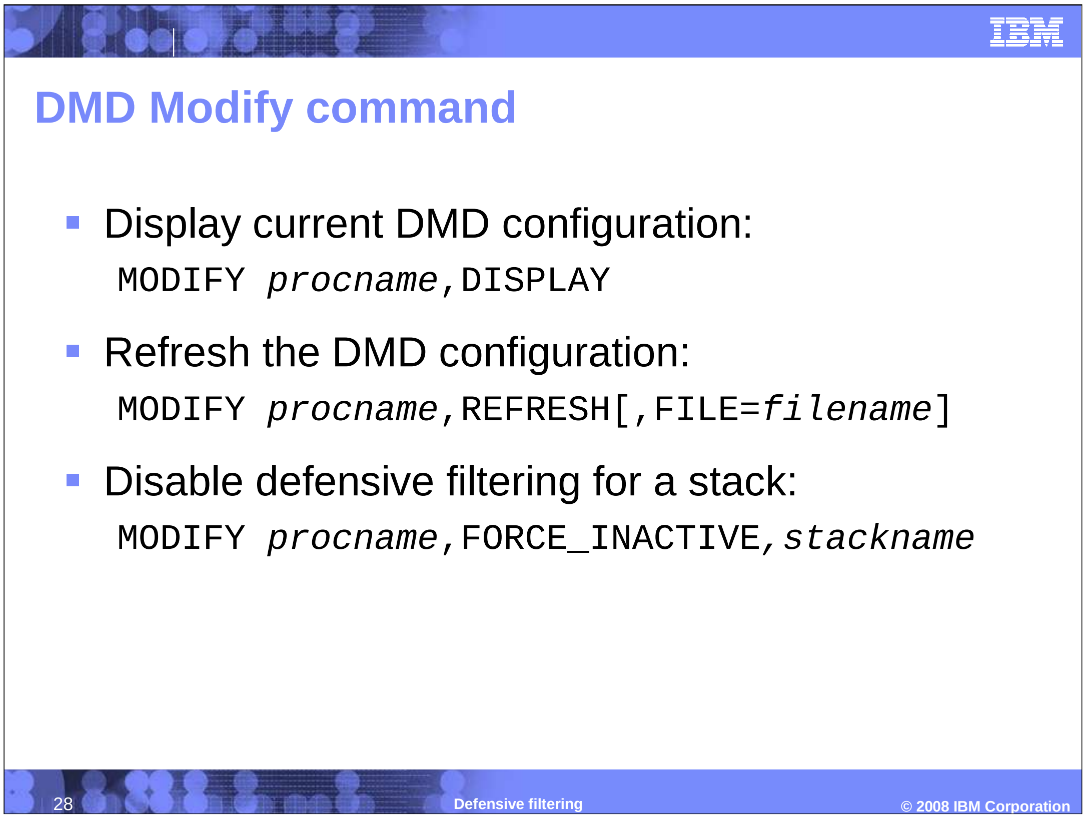
You can use an MVS procedure from the MVS operator console. A sample start procedure is provided in SEZAINST(DMD). Set the environment variables using the STDENV DD statement in the DMD procedure.

You can issue the dmd command from the z/OS UNIX shell.

You can use the COMMNDxx member of PARMLIB. This allows the DMD to be automatically started when the system is IPLed. For information about the use and configuration of the COMMNDxx member of PARMLIB, see *z/OS MVS Initialization and Tuning Reference*.

You can use the AUTOLOG statement in the TCP/IP profile. You should not start the DMD using the AUTOLOG statement in a stack's profile if you are running in a CINET environment with more than one stack configured. In that environment it is better to consider using the COMMNDxx member of PARMLIB to automatically start the DMD.

Only one instance of the DMD can run on a z/OS image (LPAR). If you attempt to start a second instance, that instance will fail and the original instance continues to run.



The DMD provides support for a MODIFY command.

You can use the MODIFY DISPLAY command to display the configuration values currently in use by the DMD.

You can use the MODIFY REFRESH command to cause the DMD to reread its configuration. Most DMD configuration parameters can be updated using the MODIFY REFRESH command but not all. For information about whether a parameter can be updated using MODIFY REFRESH refer to the parameter's description in the *z/OS Communications Server IP Configuration Reference*.

You can use the MODIFY FORCE_INACTIVE command to disable defensive filtering for a stack. MODIFY FORCE_INACTIVE forces the TCP/IP stack's defensive filtering Mode to Inactive. All defensive filters for the stack are removed from the DMD and from the stack. No additional defensive filters can be added for the stack while it is in Inactive mode. The change to the stack's Mode persists until the next successful MODIFY REFRESH.

Note that the stack does not have to be configured in the DMD configuration file in order for the FORCE_INACTIVE to be processed. If the stack is active and IP security is enabled, any defensive filters are removed regardless of the stack's DMD configuration status.

For more information on the MODIFY command, see *z/OS Communications Server IP System Administrator's Commands*.

IBM

Defensive filtering and IP security

- IP security must be enabled to use defensive filtering.
- If you already have an IP security policy no change is needed to your current policy.
- If you do not have an IP security policy
  - An IP security policy must be implemented for each stack where you want to use defensive filtering.
  - A minimal policy can be implemented in the TCP/IP profile to allow all traffic.
    - You can now allow routed traffic in the TCP/IP profile policy

29 Defensive filtering © 2008 IBM Corporation

IP security must be enabled for a TCP/IP stack to be able to do defensive filtering. If you already have IP security enabled and an IP security policy in place, no change is required to your TCP/IP profile or IP security policy to use defensive filtering!

If you do not have IP security enabled, see the *z/OS Communications Server IP Configuration Guide* for information on enabling IP security and creating an IP security policy. If you do not need to have a comprehensive IP security policy you can create a minimal policy in your TCP/IP profile that allows all traffic. Defensive filters can then be installed to deny traffic as threats are detected. In z/OS V1R10 Communications Server, a ROUTING parameter has been added to the IPSECRULE and IPSEC6RULE so that routed traffic can be allowed by the default filter policy.

Defensive filtering and IPv4 and IPv6 addresses (part 1)

- Defensive filtering is supported for IPv4 and IPv6 traffic.
- srcip and destip values
  - IPv4 address, IPv6 address, or all
  - Cannot have an IPv4 address for one value and an IPv6 address for the other.
    - **ipsec –F add srcip 10.1.1.1 destip 9::8** ←rejected

30 Defensive filtering © 2008 IBM Corporation

Defensive filtering is supported for both IPv4 and IPv6 traffic. The srcip value specified on an **ipsec –F add** command can be an IPv4 address, an IPv6 address, or all. The destip value specified can also be an IPv4 address, an IPv6 address, or all.

A filter with one or both values specified as an IPv4 address will result in an IPv4 defensive filter which will block IPv4 traffic.

A filter with one or both values specified as an IPv6 address will result in an IPv6 filter which will block IPv6 traffic.

You are not allowed to specify an IPv4 address for one value and an IPv6 address for the other. So for example, **ipsec –F add srcip 10.1.1.1 destip 9::8** is rejected.

Defensive filtering and IPv4 and IPv6 addresses (part 2)

- If defensive filter added with all for both srcip and destip:
  - IPv4 defensive filter added.
  - If the stack supports IP security for IPv6, IPv6 defensive filter also added.
  - Base name is the same for both filters. Different index values are assigned.

Example:

```
ipsec –F add srcip all destip all prot tcp srcport 4567
–N Block_TCP_port_4567
```


© 2008 IBM Corporation

If a defensive filter is added with a value of all for both the srcip and the destip an IPv4 defensive filter is added. If the stack supports IP security for IPv6, an IPv6 defensive filter is also added. A stack supports IP security for IPv6 if IPSECURITY is specified on the IPCONFIG6 statement in the TCP/IP profile.

The base name is the same for both filters. Different index values are assigned to the IPv4 and IPv6 filters.

In the example the command **ipsec –F add srcip all destip all prot tcp srcport 4567 –N Block_TCP_port_4567** is issued, what do you expect to happen? Because srcip and destip are both all, both an IPv4 and IPv6 filter are added.

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

UNIX is a registered trademark of The Open Group in the United States and other countries.

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.

© Copyright International Business Machines Corporation 2008. All rights reserved.

Note to U.S. Government Users - Documentation related to restricted rights-Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract and IBM Corp.