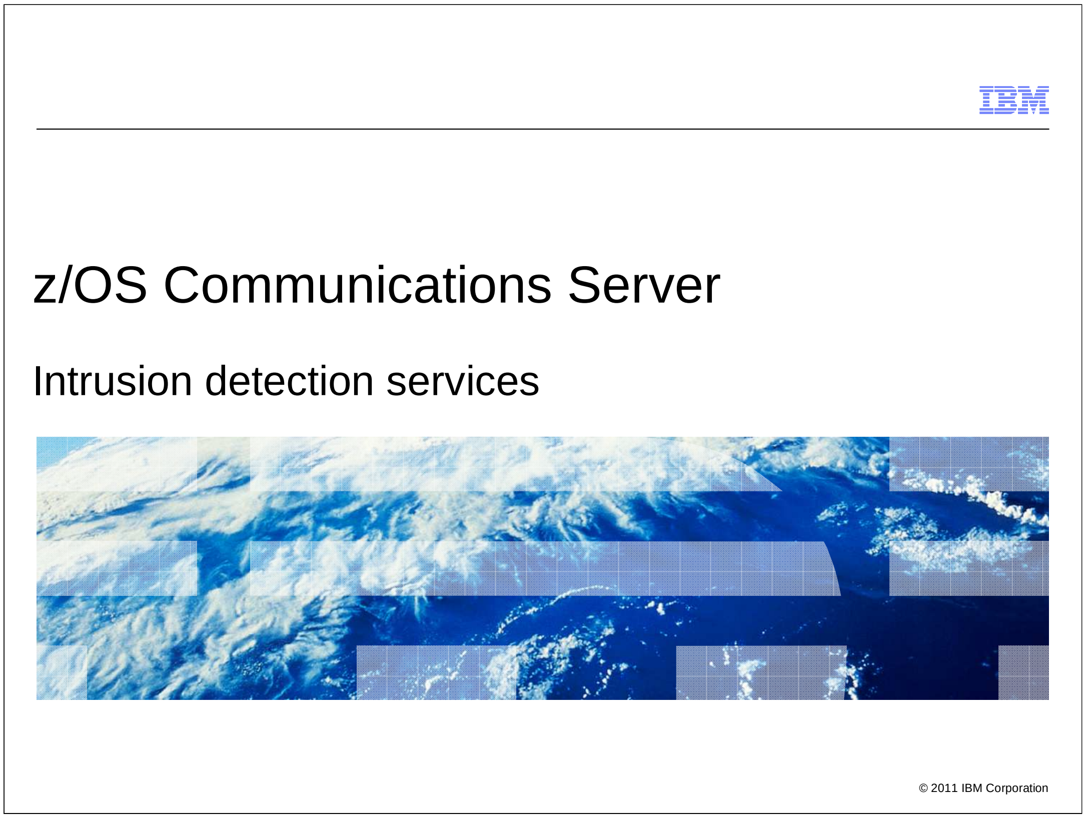
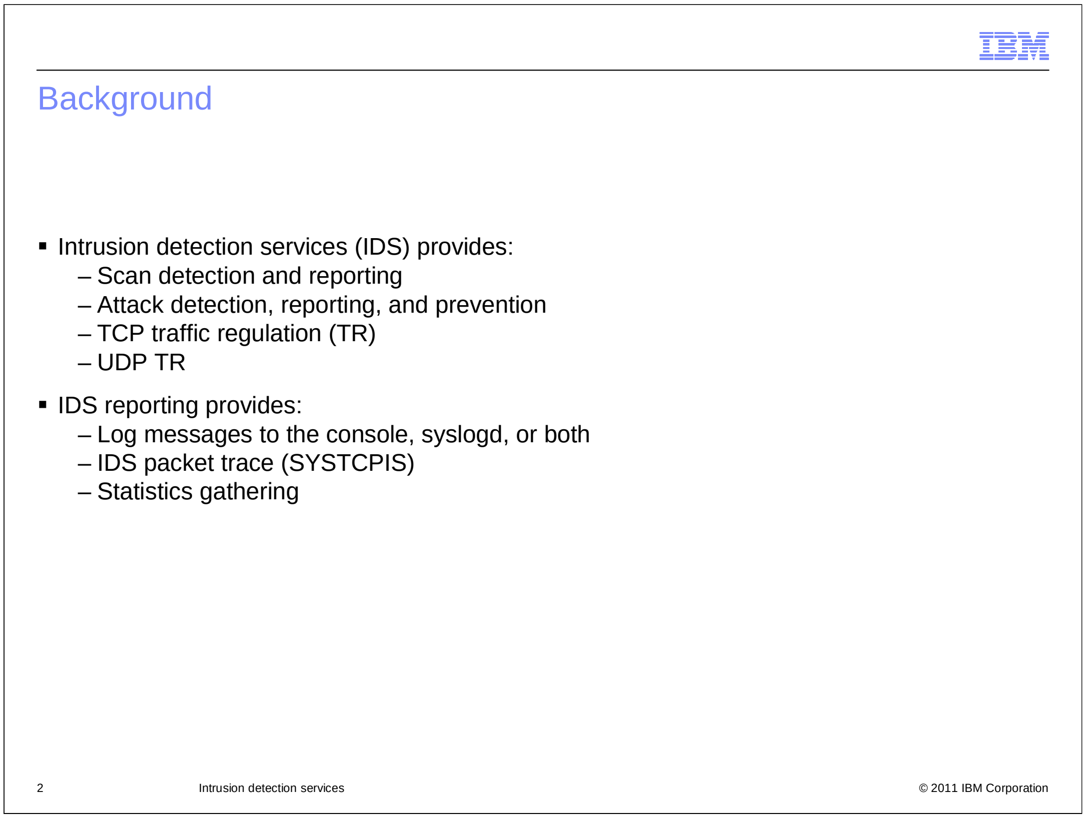


# z/OS Communications Server

## Intrusion detection services

This presentation describes the enhancements to the intrusion detection services, including support for IPv6, in z/OS® V1R13 Communications Server. This presentation includes some background information, a description of the changes, migration information, diagnosis, and other information. The function externals for IDS are described in a separate presentation.



## Background

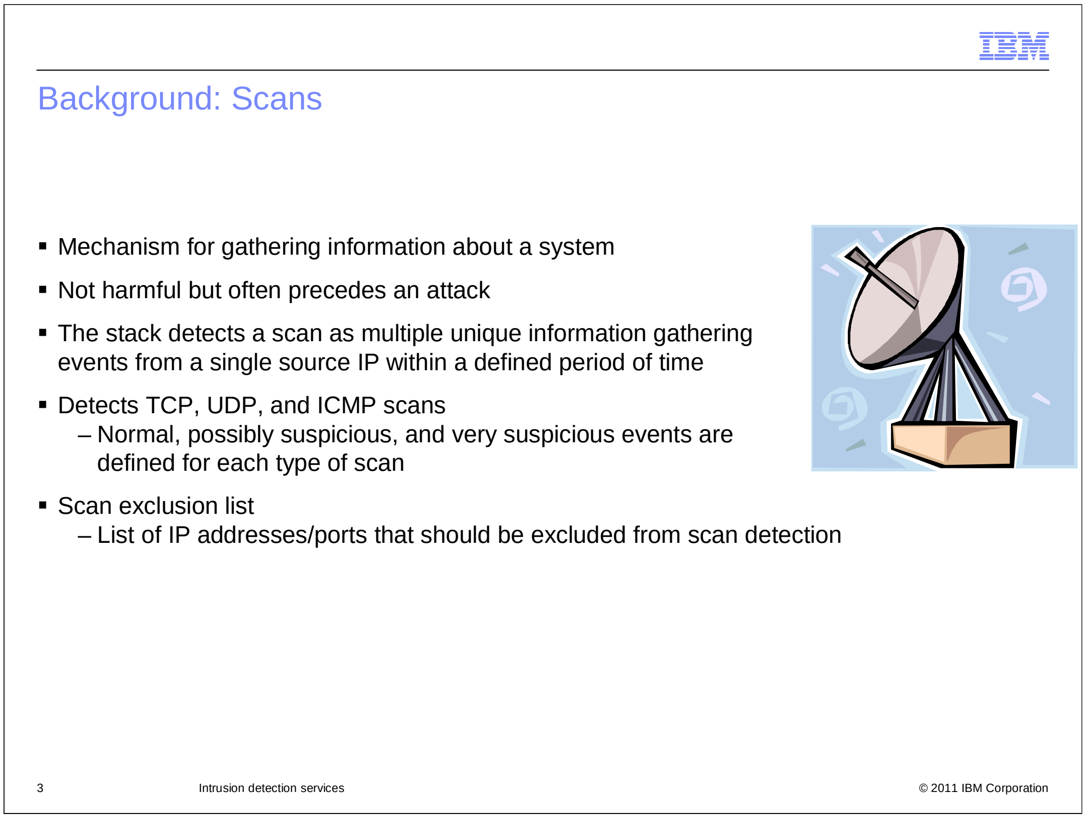
- Intrusion detection services (IDS) provides:
  - Scan detection and reporting
  - Attack detection, reporting, and prevention
  - TCP traffic regulation (TR)
  - UDP TR
- IDS reporting provides:
  - Log messages to the console, syslogd, or both
  - IDS packet trace (SYSTCPIS)
  - Statistics gathering

2 Intrusion detection services © 2011 IBM Corporation

z/OS Communications Server provides intrusion detection services that are integrated into the TCP/IP stack's processing. Checking is in context, allowing factors such as connection state to be considered when evaluating a packet. This integrated approach complements external IDS systems that monitor packets for signature-based intrusions.

Intrusion Detection Services (IDS), first introduced in V1R2, can be enabled to detect attacks and scans. It also includes traffic regulation support for TCP servers 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 exceed the configured connection limit or receive queue length.

There are several IDS reporting options. You can choose to have events logged to the system console, to syslogd, or to both. You can choose to have packets traced to the IDS packet trace. You also can request that statistics be kept for various IDS events that occur. Reporting is 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, that is, activity in that stack. However, by generating messages that can be consumed by an external security information and event manager, a TCP/IP stack's 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.



## Background: Scans

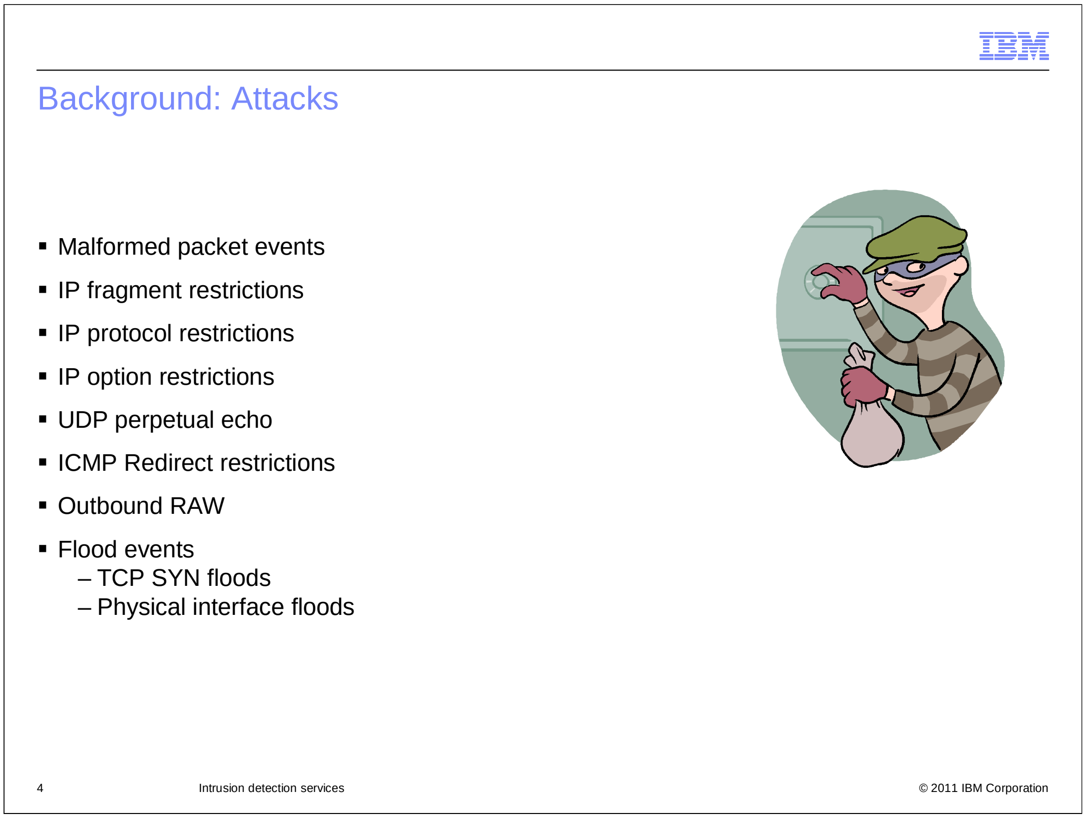
- Mechanism for gathering information about a system
- Not harmful but often precedes an attack
- The stack detects a scan as multiple unique information gathering events from a single source IP within a defined period of time
- Detects TCP, UDP, and ICMP scans
  - Normal, possibly suspicious, and very suspicious events are defined for each type of scan
- Scan exclusion list
  - List of IP addresses/ports that should be excluded from scan detection

A scan is a mechanism for gathering information about a system. For example, a port scan can determine which ports are open on a system and are potentially available for attack. The scan itself is not harmful, but it often precedes an attack. Information learned about the scanner can be useful in tracing an attack.

z/OS Communications Server detects a scan when multiple unique information-gathering events from a single source IP address occur within a defined period of time.

What is an information-gathering event? z/OS Communications Server classifies TCP, UDP, and ICMP events as normal, possibly suspicious, and very suspicious. For example, receiving a RST for a half-open TCP connection is considered a possibly suspicious event. You configure the types of events that IDS should monitor. For example, you might enable scan detection for TCP connections on the well known ports (1-1023) with a suspicion level of MEDIUM. This results in all possibly suspicious and very suspicious TCP events for ports 1-1023 being counted.

If you have network monitoring tools that perform legitimate scans, use the scan exclusion list to exclude these devices from scan detection. The scan exclusion list is a configured list of IP addresses and optionally ports to be excluded from scan detection.



## Background: Attacks

- Malformed packet events
- IP fragment restrictions
- IP protocol restrictions
- IP option restrictions
- UDP perpetual echo
- ICMP Redirect restrictions
- Outbound RAW
- Flood events
  - TCP SYN floods
  - Physical interface floods

An attack can be a single packet that is designed to stop or hang a system or it can be multiple packets designed to consume a limited resource causing a denial of service. z/OS Communications Server’s IDS support offers eight different attack types that can be enabled. For each enabled attack type, you configure the types of notification that are provided if an attack is detected. Also, you configure whether the packet should be discarded or not.

The Malformed packet attack type provides notification when a malformed packet is detected. Malformed packets have incorrect or partial header information. The TCP/IP stack ALWAYS discards these packets, regardless of whether IDS is configured or not.

The IP fragment restrictions attack type detects packets that are fragmented within the first 88 bytes. This is considered suspicious behavior, although it does not violate any RFC standards.

The IP protocol restrictions attack type detects packets using IP protocols that you define as restricted.

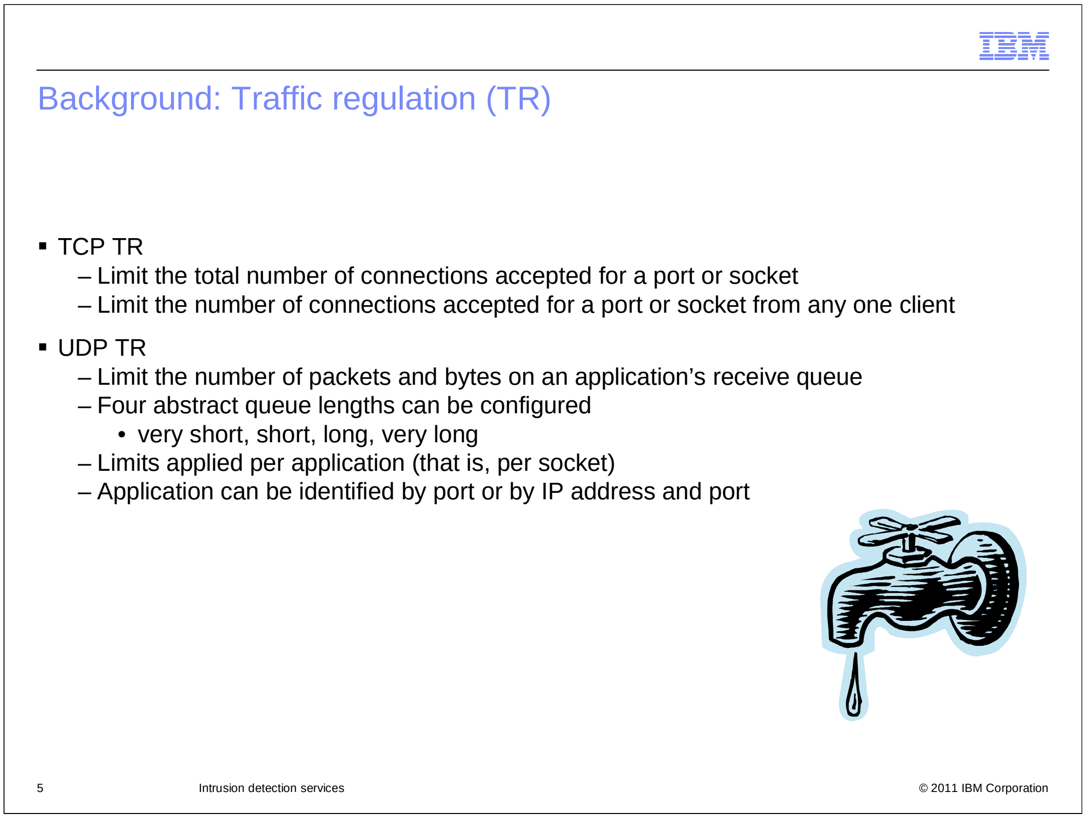
The IP option restrictions attack type detects packets that include IP options that you define as restricted.

The UDP perpetual echo attack type detects UDP packets with source and destination applications (that is, ports) that will always respond, causing a perpetual echo.

The ICMP redirect restrictions attack type restricts the use of ICMP redirects.

The Outbound RAW attack type detects the use of a RAW socket to build a malicious packet.

The Flood attack type includes detection and reporting for TCP SYN floods and physical interface floods. A TCP SYN flood is a flood of SYN packets (that is, connection requests) directed to a TCP listening application from spoofed addresses. The intent of a SYN flood is to fill up the listener’s backlog queue and block legitimate connection requests. Processing to protect the TCP listening application is ALWAYS in place, regardless of whether IDS is configured or not. Enabling the flood attack allows you to receive notification about the attack. An interface flood is detected when a large number of the incoming packets over an interface is being discarded. Enabling the flood attack allows you to receive notification about the attack.

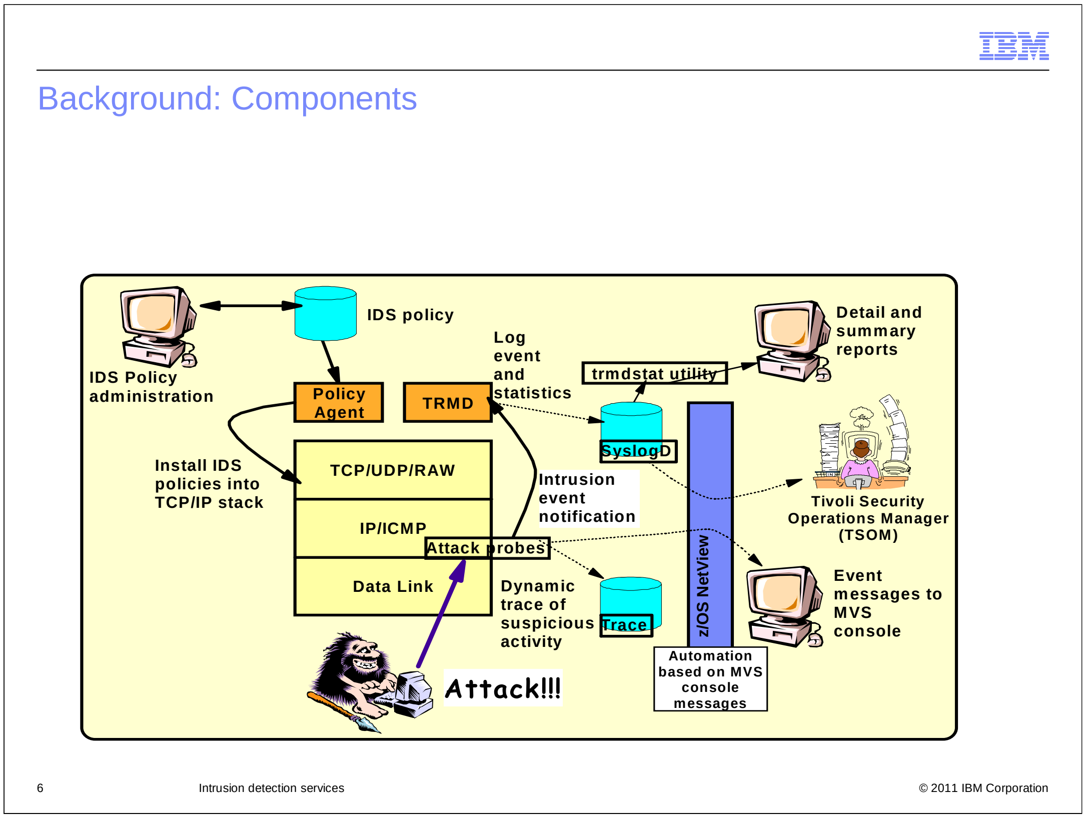


## Background: Traffic regulation (TR)

- TCP TR
  - Limit the total number of connections accepted for a port or socket
  - Limit the number of connections accepted for a port or socket from any one client
- UDP TR
  - Limit the number of packets and bytes on an application’s receive queue
  - Four abstract queue lengths can be configured
    - very short, short, long, very long
  - Limits applied per application (that is, per socket)
  - Application can be identified by port or by IP address and port

z/OS Communications Server’s TCP TR support provides a fair-share algorithm to prevent any one client from consuming all the connection resources for a server. You can limit a server’s total number of active connections. You can also limit the number of active connections that any one client can have. Limits can be configured for a specific port (for example, port 21 for FTP) or for a specific port instance. When multiple applications are bound to the same port, applying the limits based on port instance allows you to have different limits based on the characteristics of each application. For example, in some cases, telnetd and TN3270 can both bind to port 23. This is a case where you might want limits to be applied per port instance, not across both applications.

z/OS Communications Server’s UDP TR support provides a mechanism to limit the number of packets and the number of bytes on a UDP application’s receive queue. There are four abstract queue lengths that can be configured. The limit is applied per socket. The application can be identified by port or by a combination of the IP address and port.



## Background: Components

IDS policy
IDS Policy administration
Policy Agent
TRMD
Log event and statistics
trmdstat utility
Detail and summary reports
SyslogD
Tivoli Security Operations Manager (TSOM)
Install IDS policies into TCP/IP stack
TCP/UDP/RAW
IP/ICMP
Data Link
Attack probes
Intrusion event notification
Dynamic trace of suspicious activity
Trace
z/OS NetView
Event messages to MVS console
Automation based on MVS console messages
Attack!!!

6 Intrusion detection services © 2011 IBM Corporation

In this picture you can see many of the components involved in z/OS Communications Server's IDS support.

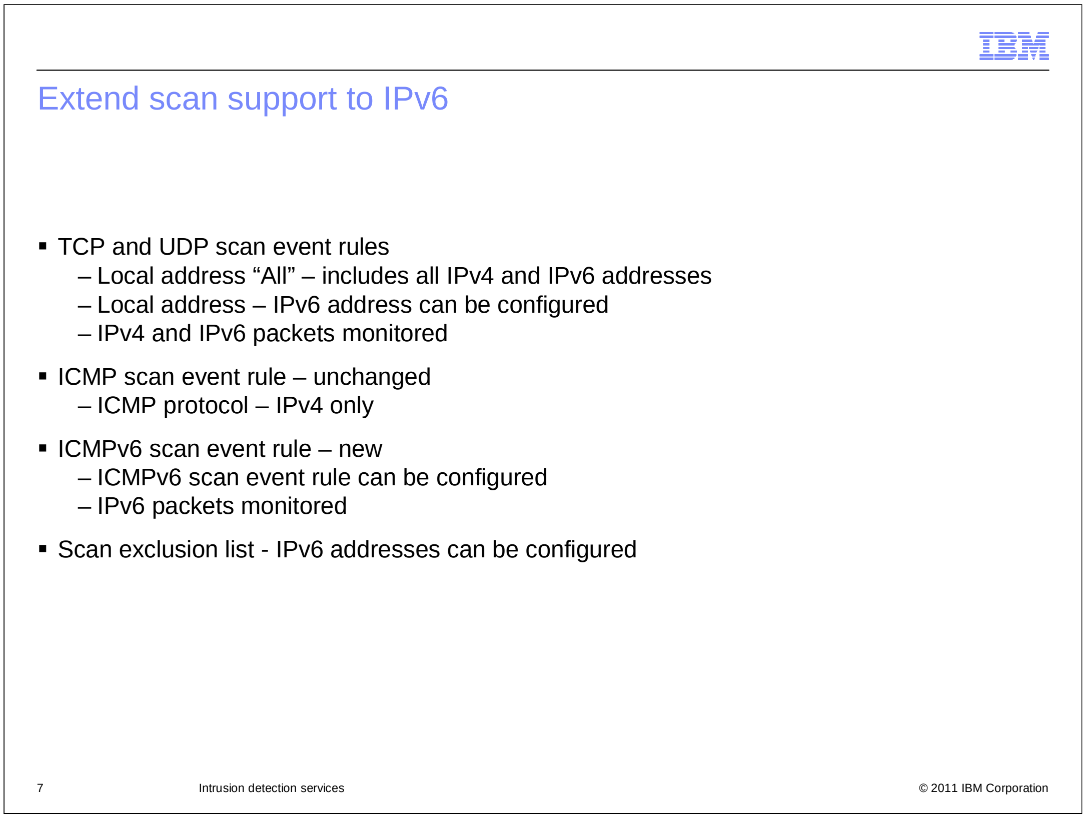
The first step is to enable IDS protection. The IDS policy administrator can use the Configuration Assistant to create an IDS policy file, or manually create an IDS policy. **Policy Agent** reads the IDS policy and installs it in the **TCP/IP stack**.

With IDS policy installed, you can follow the flow of an external attack against the TCP/IP stack. Based on the configured policy rules, the **TCP/IP stack** detects the attack, discards the packet, and takes a series of notification steps. The stack provides **TRMD** with data to generate a message to be logged to **syslogd**. The stack also generates a message for display on the **MVS console** and creates a trace entry that is written to the **IDS Trace dataset (SYSTCPIS)**. Finally, the stack counts the attack in statistics data. The statistics data is provided to **TRMD** on a configured time interval and TRMD will generate a statistics message that is logged to **syslogd**.

The z/OS UNIX® system services **trmdstat command** can be used to generate reports from the syslogd messages, providing a consolidated view of the messages.

External event managers, such as Tivoli® Security Event Information Manager (TSEIM), can monitor the **syslogd log file**. z/OS Netview automation can monitor either the **MVS console messages** or the **syslogd messages**.

The defense manager daemon (DMD) is not shown in the picture because it is not an IDS component. However, defensive filtering can be used to block packets in response to an attack.



## Extend scan support to IPv6

- TCP and UDP scan event rules
  - Local address “All” – includes all IPv4 and IPv6 addresses
  - Local address – IPv6 address can be configured
  - IPv4 and IPv6 packets monitored
- ICMP scan event rule – unchanged
  - ICMP protocol – IPv4 only
- ICMPv6 scan event rule – new
  - ICMPv6 scan event rule can be configured
  - IPv6 packets monitored
- Scan exclusion list - IPv6 addresses can be configured

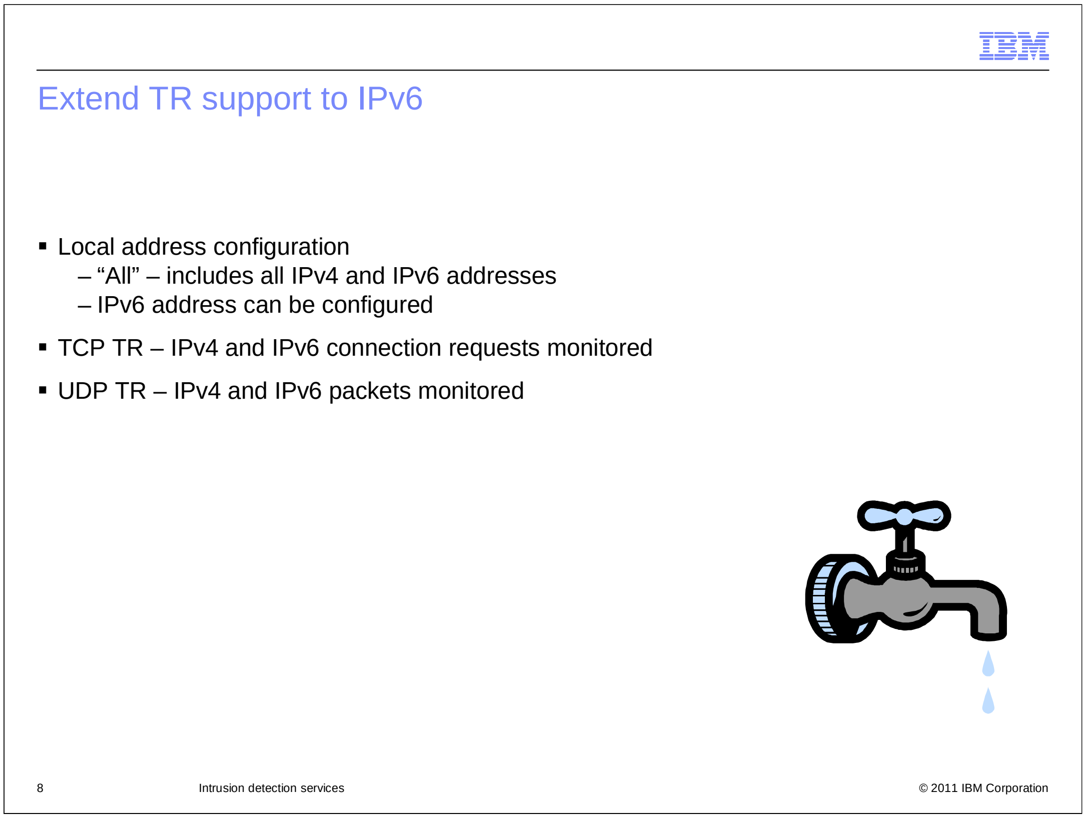
The existing IDS scan support is extended to IPv6.

For TCP and UDP scan event rules, the local address defaults to a value of All, which includes both IPv4 and IPv6 addresses. For cases where you want to configure a specific IP address, the address can now be an IPv4 or IPv6 address. Both IPv4 and IPv6 packets are monitored for TCP and UDP scan events.

The existing ICMP scan event rule is unchanged. The ICMP protocol applies to IPv4 only.

A new ICMPv6 scan event rule can be configured. IPv6 packets are monitored for ICMPv6 scan events.

Now that IPv6 packets are monitored for scan events, you might need to add additional addresses to the scan exclusion list. Both IPv4 and IPv6 addresses can be in the exclusion list.



## Extend TR support to IPv6

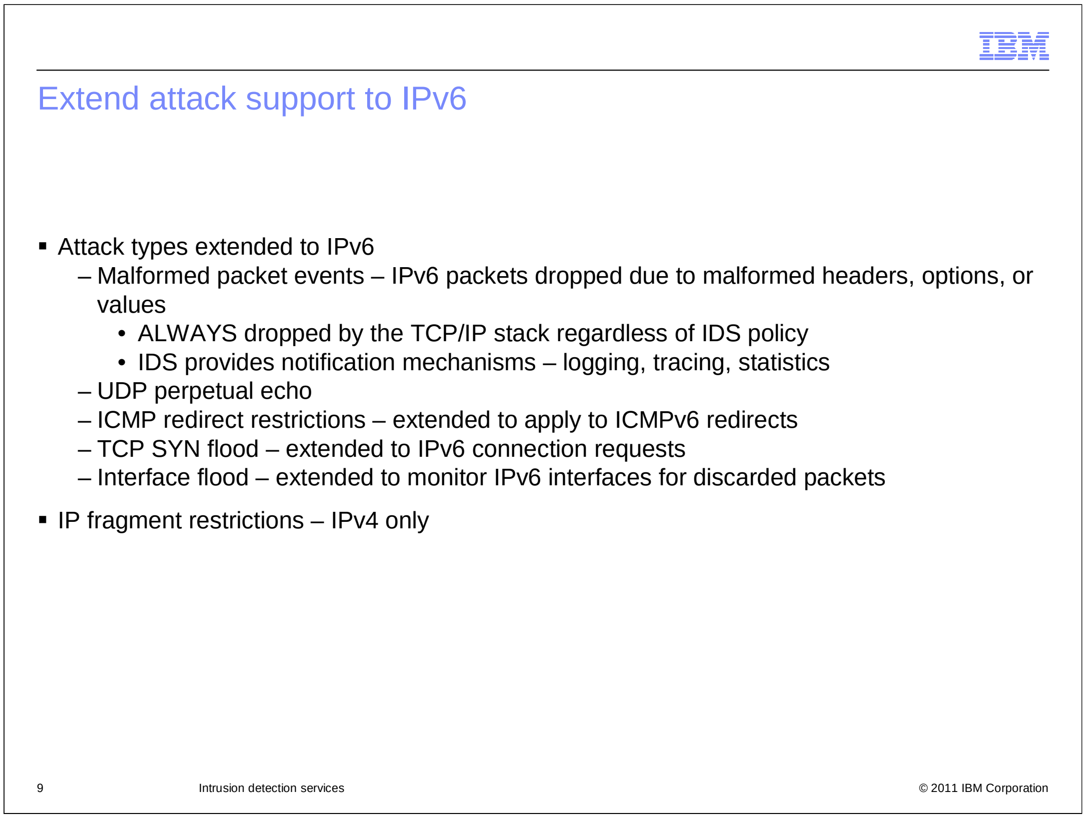
- Local address configuration
  - “All” – includes all IPv4 and IPv6 addresses
  - IPv6 address can be configured
- TCP TR – IPv4 and IPv6 connection requests monitored
- UDP TR – IPv4 and IPv6 packets monitored

The existing IDS TR support is extended to IPv6.

For TCP and UDP TR rules, the local address defaults to a value of All, which includes both IPv4 and IPv6 addresses. For cases where you want to configure a specific IP address, the address can now be an IPv4 or IPv6 address.

Both IPv4 and IPv6 connection requests are monitored for TCP traffic regulation.

Both IPv4 and IPv6 packets are monitored for UDP traffic regulation.



## Extend attack support to IPv6

- Attack types extended to IPv6
  - Malformed packet events – IPv6 packets dropped due to malformed headers, options, or values
    - ALWAYS dropped by the TCP/IP stack regardless of IDS policy
    - IDS provides notification mechanisms – logging, tracing, statistics
  - UDP perpetual echo
  - ICMP redirect restrictions – extended to apply to ICMPv6 redirects
  - TCP SYN flood – extended to IPv6 connection requests
  - Interface flood – extended to monitor IPv6 interfaces for discarded packets
- IP fragment restrictions – IPv4 only

9 Intrusion detection services © 2011 IBM Corporation

The IDS attack types listed here are extended to IPv6:

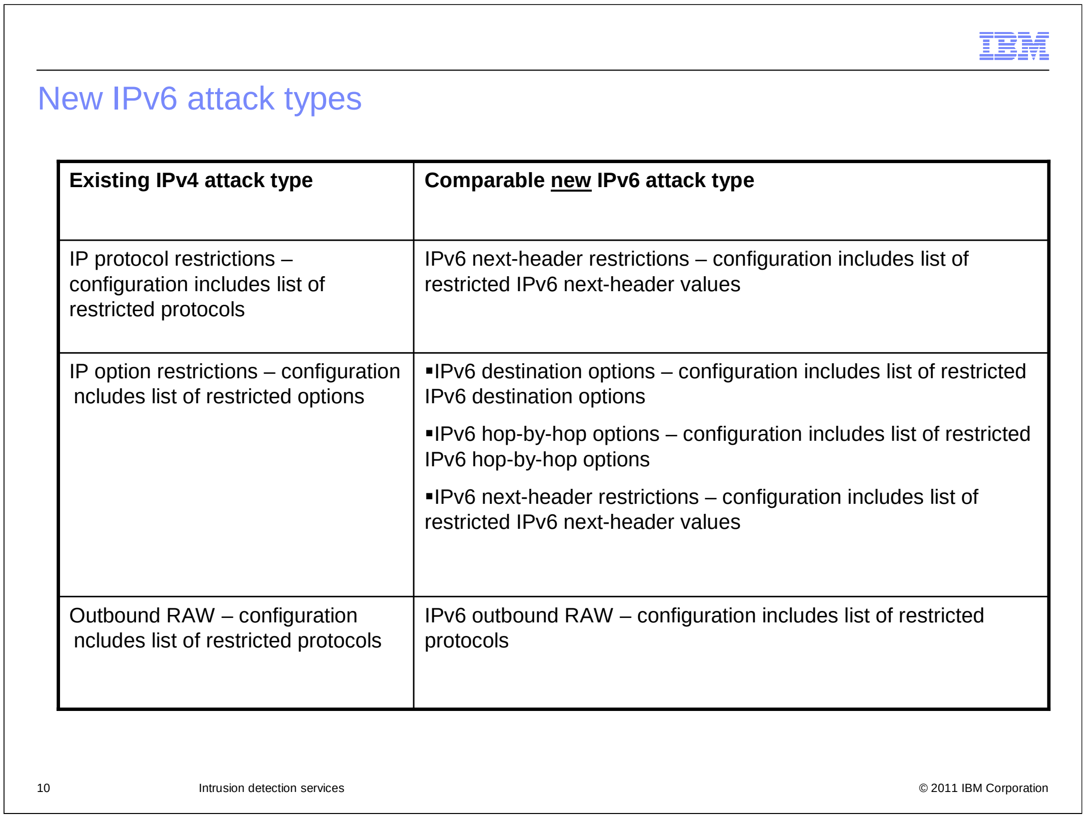
If the malformed packet attack type is enabled, IDS notification is provided for both malformed IPv4 and IPv6 packets. The TCP/IP stack always discards these malformed packets whether IDS policy is enabled or not. An IPv6 packet can be discarded due to a malformed header, option, or value.

If the UDP perpetual echo attack type is enabled, both IPv4 and IPv6 packets are checked for a perpetual echo attack. For this attack type, you can choose whether the packet should be discarded and how you want to be notified when an attack is detected.

If the ICMP redirect restrictions attack type is enabled, checking is done for both an ICMP redirect and an ICMPv6 redirect packet. For this attack type, you can choose whether the packet should be discarded and how you want to be notified when an attack is detected.

If the flood attack type is enabled, the detection and reporting of TCP SYN floods is provided for both IPv4 and IPv6 connection requests. Any IPv6 packets discarded due to this attack are monitored, which allows a physical interface flood to be detected for an IPv6 interface and IDS reporting to be performed.

However, the IP fragment restrictions attack type remains an IPv4-only attack type. If this attack type is enabled, only IPv4 packets are checked.



## New IPv6 attack types

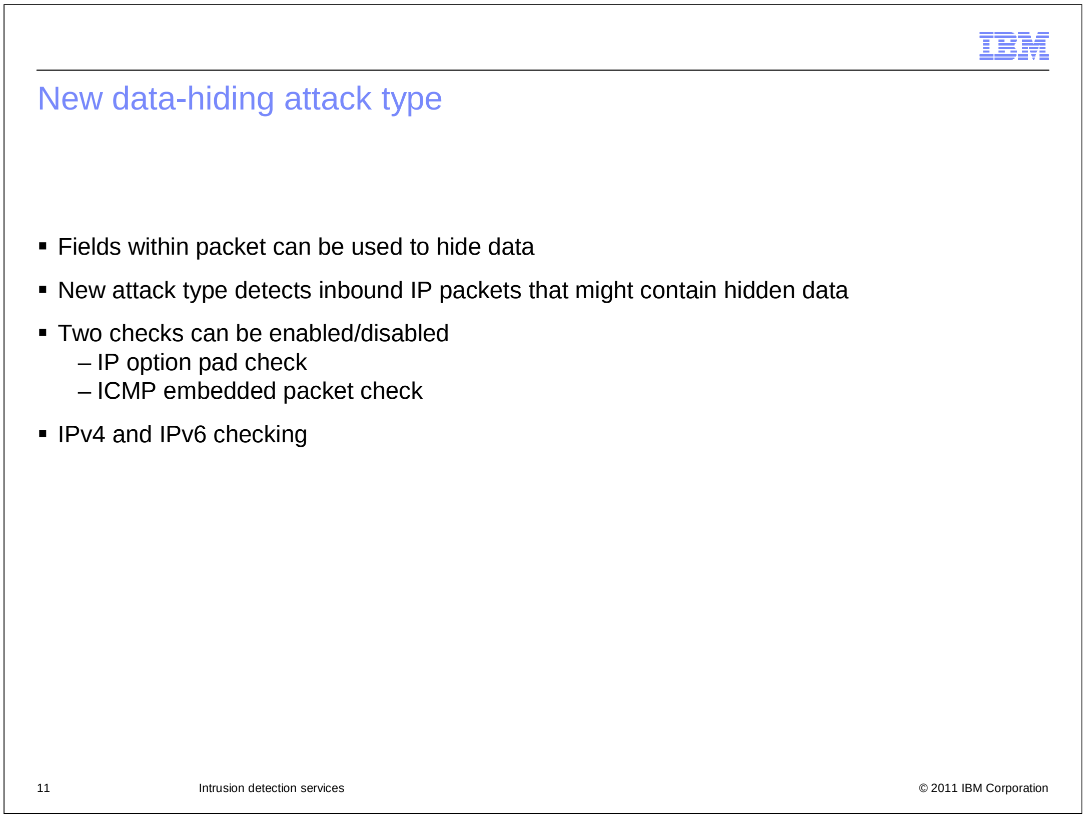
| Existing IPv4 attack type | Comparable new IPv6 attack type |
|---|---|
| IP protocol restrictions – configuration includes list of restricted protocols | IPv6 next-header restrictions – configuration includes list of restricted IPv6 next-header values |
| IP option restrictions – configuration ncludes list of restricted options | ▪IPv6 destination options – configuration includes list of restricted IPv6 destination options<br>▪IPv6 hop-by-hop options – configuration includes list of restricted IPv6 hop-by-hop options<br>▪IPv6 next-header restrictions – configuration includes list of restricted IPv6 next-header values |
| Outbound RAW – configuration ncludes list of restricted protocols | IPv6 outbound RAW – configuration includes list of restricted protocols |

This table covers the set of attacks that conceptually apply to both IPv4 and IPv6, but require very different implementations due to differences in the IPv4 and IPv6 packet formats.

The IPv4 header has a protocol field. The value in the protocol field identifies the upper-layer protocol header that follows the IP header (such as TCP or UDP). The IP protocol restrictions attack type allows you to configure a list of protocols that should not be in use on your system. The IPv6 header and any subsequent extension headers include a next-header field. The value in the next-header field identifies the next header in the packet. It is either an upper-layer protocol header (such as a TCP or UDP header) or an extension header (such as a fragmentation or routing header). The new IPv6 next-header restrictions attack type allows you to configure a list of next-header values that should not be in use on your system.

The IPv4 header can include up to 40 bytes of option data. The IP option restrictions attack type allows you to configure a list of IP options that should not be in use on your system. An IPv6 packet includes options through the use of an IPv6 destination-options extension header and an IPv6 hop-by-hop-options extension header. The new IPv6 destination-option restrictions attack type allows you to configure a list of option values that should not be received in an IPv6 destination-option extension header. The new IPv6 hop-by-hop-option restrictions attack type allows you to do the same for the IPv6 hop-by-hop-option extension header. In some cases, a function provided in IPv4 options is provided by a distinct IPv6 extension header not by one of the IPv6 option extension headers. For example, the functions provided by the IPv4 loose and strict source routing options are provided for IPv6 by the IPv6 routing extension header. So in that case, the IPv6 next-header restrictions provide the comparable function to the IPv4 option restrictions attack type.

IPv4 RAW support provides a mechanism for an application to build the entire IP packet, including the IP header. This support can be used to build an attack packet. IPv6 RAW support is more restrictive. There is no mechanism to allow an application to build the IPv6 header. The existing outbound RAW attack type detects attempts to build a packet which can be used to initiate an attack. One check restricts the protocol value in the packet based on the configured restricted protocol list. The new IPv6 outbound RAW attack type also detects attempts to build an attack packet. A restricted protocol list is part of this attack type.



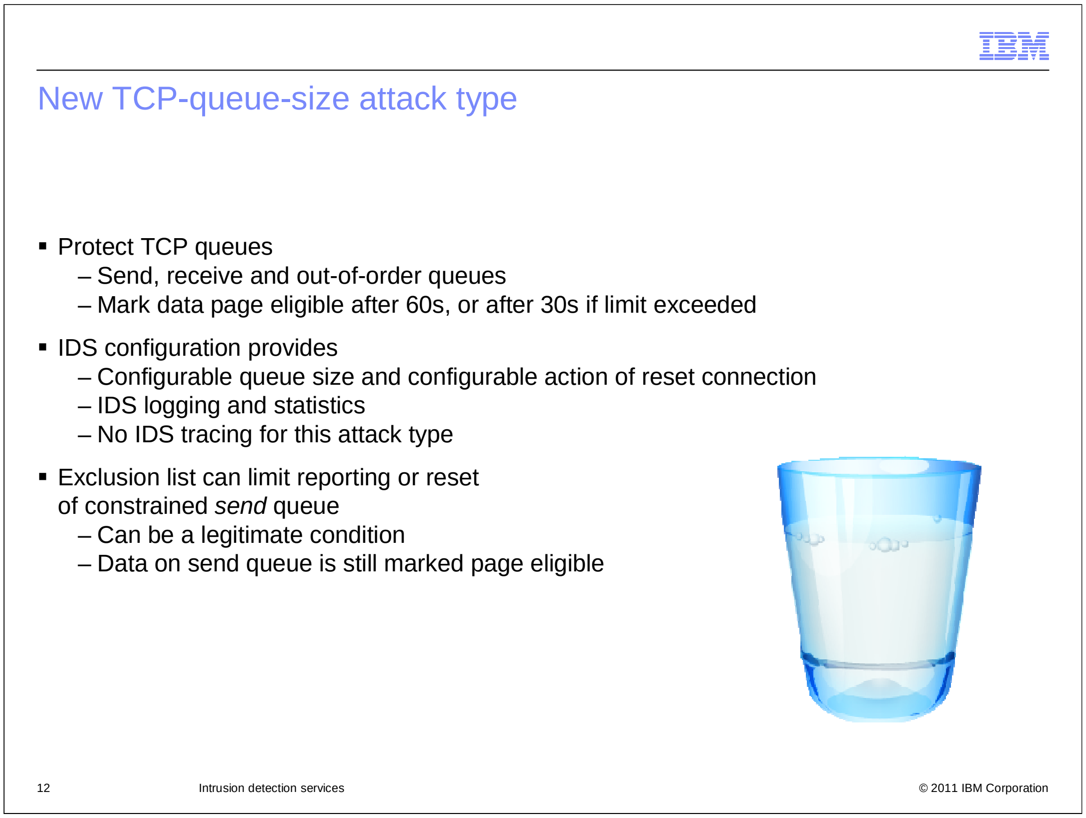
## New data-hiding attack type

- Fields within packet can be used to hide data
- New attack type detects inbound IP packets that might contain hidden data
- Two checks can be enabled/disabled
  - IP option pad check
  - ICMP embedded packet check
- IPv4 and IPv6 checking

Certain fields within a packet can be used to hide data. The new data-hiding attack type detects inbound IP packets that might contain hidden data. Checking is done for both IPv4 and IPv6 packets. The attack type includes two checks that can be enabled or disabled separately.

The first check is for non-zero IP option pad fields. For IPv4 packets, the options field is in the IP header and can contain padding for alignment purposes. For IPv6 packets, a hop-by-hop options extension header or a destination options extension header can include one or more PadN options for alignment purposes.

The second check is for hidden data in embedded packets within ICMP and ICMPv6 error messages. It verifies that the source address for the embedded packet in error corresponds to the destination address on the ICMP or ICMPv6 packet.



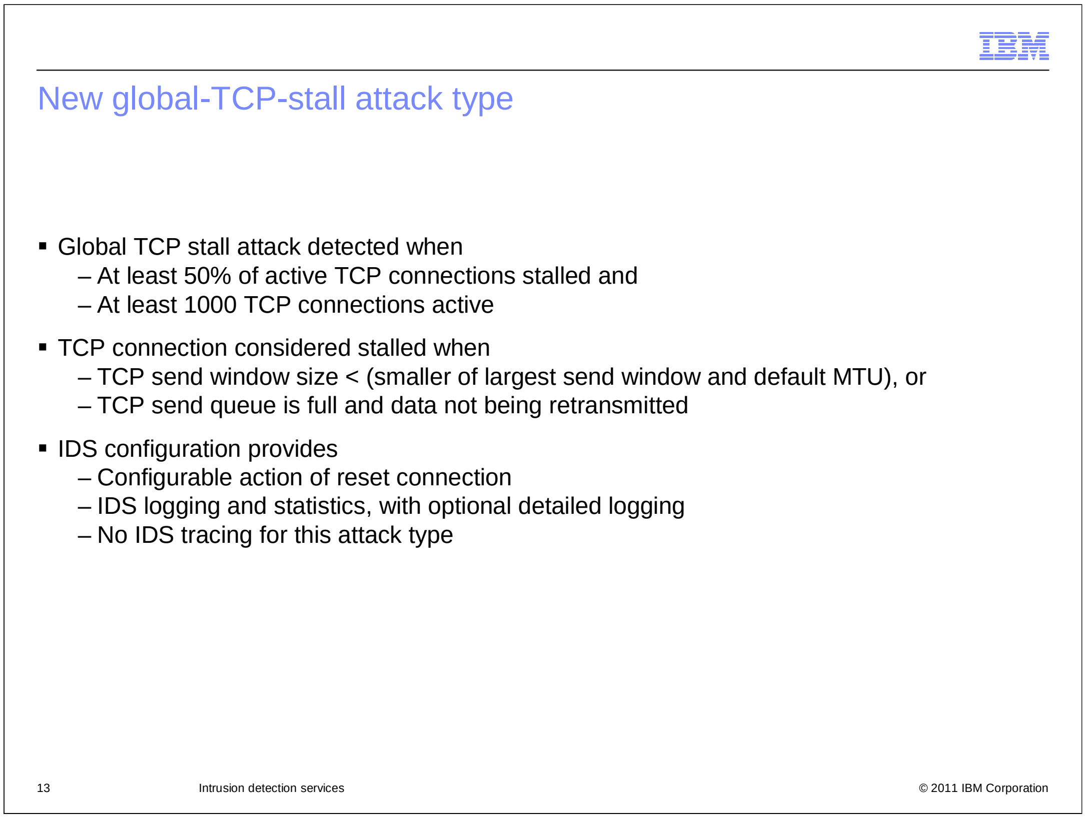
## New TCP-queue-size attack type

- Protect TCP queues
  - Send, receive and out-of-order queues
  - Mark data page eligible after 60s, or after 30s if limit exceeded
- IDS configuration provides
  - Configurable queue size and configurable action of reset connection
  - IDS logging and statistics
  - No IDS tracing for this attack type
- Exclusion list can limit reporting or reset of constrained *send* queue
  - Can be a legitimate condition
  - Data on send queue is still marked page eligible

A new IDS attack type, TCP queue size, is provided to protect connections' send, receive, and out-of-order queues. This attack type works in conjunction with existing code to mark queued data page eligible when it exceeds certain time and size limits. When you enable the TCP-queue-size attack type, you can configure a queue size value Very Short, Short, Long, or Very Long. These are abstract values that represent a percentage of the total size of each queue and are subject to change. A queue becomes constrained if the corresponding amount of data remains on the queue for at least thirty seconds. In the absence of IDS configuration, a default value of Short is used to determine when queued data is marked page eligible.

You can configure an optional action of reset connection for the TCP-queue-size attack type. When the TCP/IP stack detects a constrained queue, the connection is automatically reset if an action of reset connection has been configured. Additionally, most of the IDS notification options can be enabled for the TCP-queue-size attack type. Logging can be done to the system console, to syslogd, or to both. Statistics can be gathered. However, there is no IDS packet tracing for this attack type.

There are legitimate scenarios where a TCP connection might have data queued on the send queue for a long period of time. For example, a TCP connection associated with a printer that has run out of paper can persist in a state waiting for paper to be loaded. To exclude a device, such as a printer, from TCP send queue size checking, you can configure an exclusion specifying the IP address, and optionally the port, of the device. Excluding a remote peer from the TCP queue size attack type, keeps constrained / unconstrained messages from being generated for the send queue. It also keeps a connection from being reset when the send queue becomes constrained. It does not prevent data on a constrained send queue from being marked page eligible.



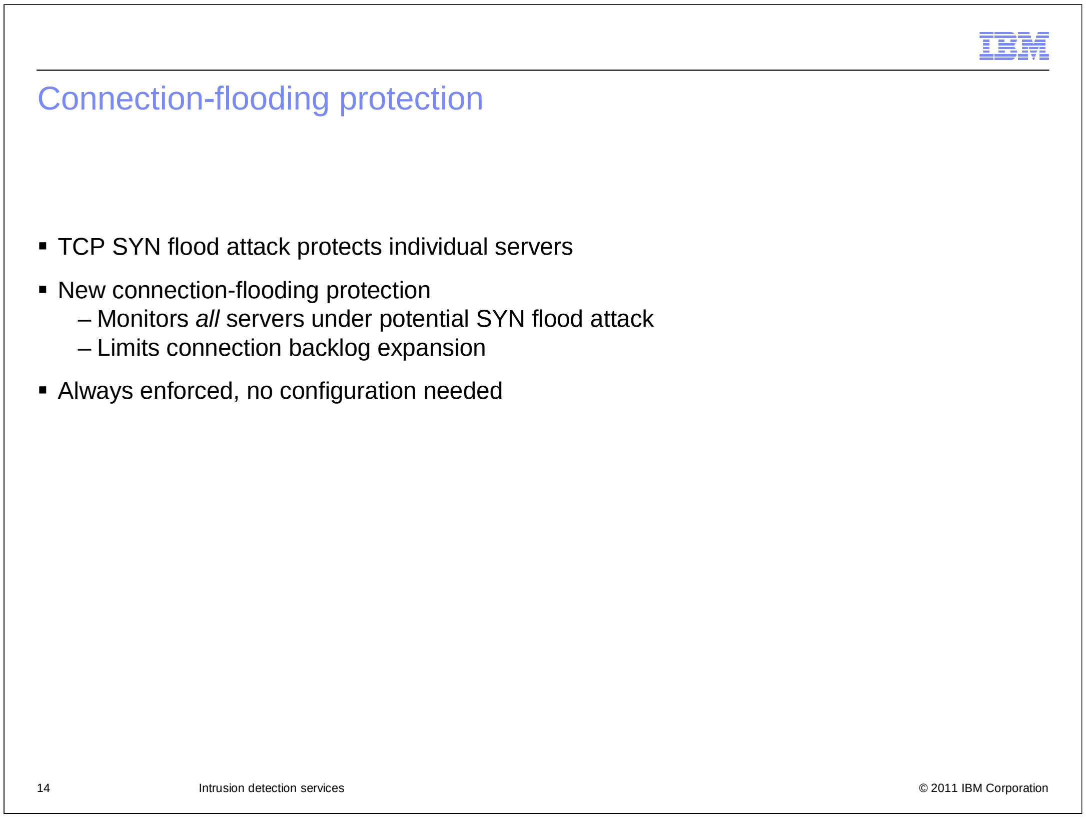
## New global-TCP-stall attack type

- Global TCP stall attack detected when
  - At least 50% of active TCP connections stalled and
  - At least 1000 TCP connections active
- TCP connection considered stalled when
  - TCP send window size < (smaller of largest send window and default MTU), or
  - TCP send queue is full and data not being retransmitted
- IDS configuration provides
  - Configurable action of reset connection
  - IDS logging and statistics, with optional detailed logging
  - No IDS tracing for this attack type

A new IDS attack type is provided to detect a global TCP stall condition. This is a case when a large number of connections are stalled sending data. Such attacks are denial-of-service attacks, and are designed to consume system resources as data is queued and unable to be sent. Since it is normal for individual connections to be stalled from time to time, this attack is only detected when a significant number and percentage of connections are stalled. If this attack type is enabled, the TCP/IP stack monitors stalled connections. A global-TCP-stall condition is detected when at least 50% of the active TCP connections are stalled and at least 1000 TCP connections are active. A global stall is exited when the stalled connection count drops below 25% of all active TCP connections or below 450.

A TCP connection, either IPv4 or IPv6, is considered stalled if one or more of two conditions are true. First, if the TCP send window size is less than the smaller of the largest send window seen for the connection and the default MTU. Second, if the TCP send queue is full and data is not being retransmitted. (Connections in retransmission are excluded from consideration because this likely represents a network outage rather than an attack.)

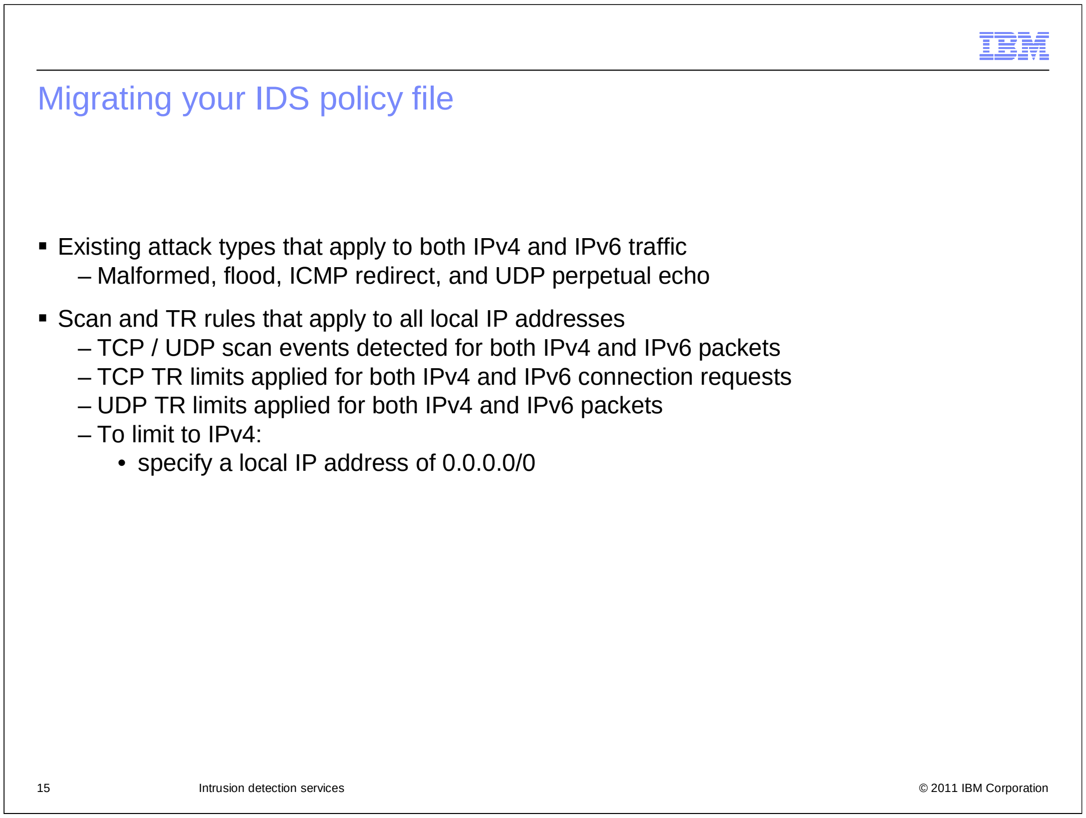
You can configure an action of reset connection for the global-TCP-stall attack type. If this action is specified, then when a global TCP stall condition is detected, the TCP/IP stack resets all stalled connections. You can also configure one or more notification options. Logging can be done to the system console, to syslogd, or to both. Statistics can be gathered. There is no IDS packet tracing for this attack type. Additionally, there are two levels of syslogd logging available for this attack type. If you have configured the base level of logging to syslogd, messages are generated when a global TCP stall is entered and exited. If you have configured detailed logging, messages are generated for each stalled connection, in addition to the general entered and exited messages. Care should be taken when using detailed logging because each time a global TCP stall is detected over 500 individual syslogd messages are generated.



## Connection-flooding protection

- TCP SYN flood attack protects individual servers
- New connection-flooding protection
  - Monitors *all* servers under potential SYN flood attack
  - Limits connection backlog expansion
- Always enforced, no configuration needed

IDS SYN-flood protection has always protected individual listening sockets from being flooded with connection requests. This protection involves expanding server backlog queues up to 768 entries to accommodate pending half-open connection requests. Beginning in V1R13, z/OS Communications Server now monitors for SYN-flood conditions across all servers. Server backlogs are not expanded if more than 25 servers are under a potential connection flood attack, in order to prevent excessive storage growth. This protection is always on; no configuration is needed.



## Migrating your IDS policy file

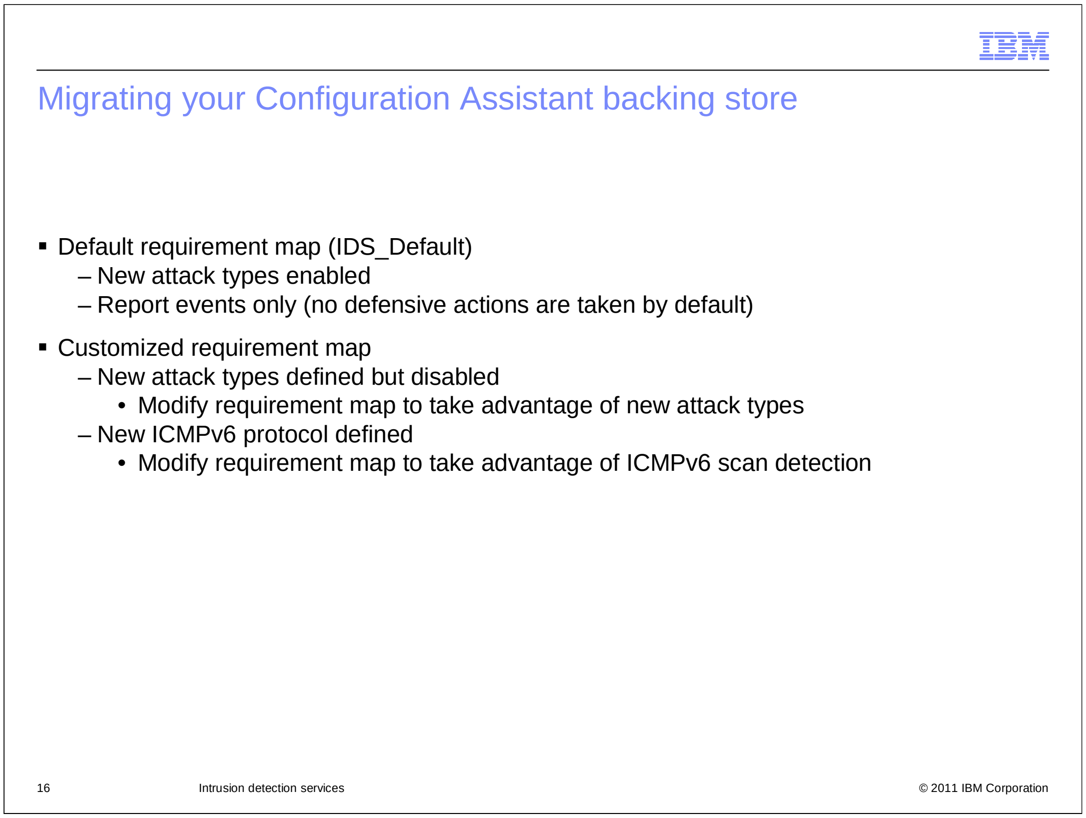
- Existing attack types that apply to both IPv4 and IPv6 traffic
  - Malformed, flood, ICMP redirect, and UDP perpetual echo
- Scan and TR rules that apply to all local IP addresses
  - TCP / UDP scan events detected for both IPv4 and IPv6 packets
  - TCP TR limits applied for both IPv4 and IPv6 connection requests
  - UDP TR limits applied for both IPv4 and IPv6 packets
  - To limit to IPv4:
    - specify a local IP address of 0.0.0.0/0

15 Intrusion detection services

If you are using IDS on a stack that is being run as a dual-mode stack (IPv4 and IPv6), your existing IDS policy is applied to IPv6 traffic in some cases. Note that this is true regardless of whether your policy is defined in a policy agent configuration file or in LDAP.

Four of the existing attack types apply to both IPv4 and IPv6 traffic: malformed, flood, ICMP redirect, and UDP perpetual echo. If you have any of these attack types enabled, be aware that when you migrate to V1R13 that both IPv4 and IPv6 traffic are monitored for attacks of these types.

Scan and traffic regulation rules can now apply to IPv6 addresses. If you have a scan or TR rule that applies to all local IP addresses, then scan events are now detected and TR limits are now enforced for both IPv4 and IPv6 activity. If you want the scan or TR rule to only apply to IPv4 activity, modify your rule to specify a local IP address of 0.0.0.0/0.



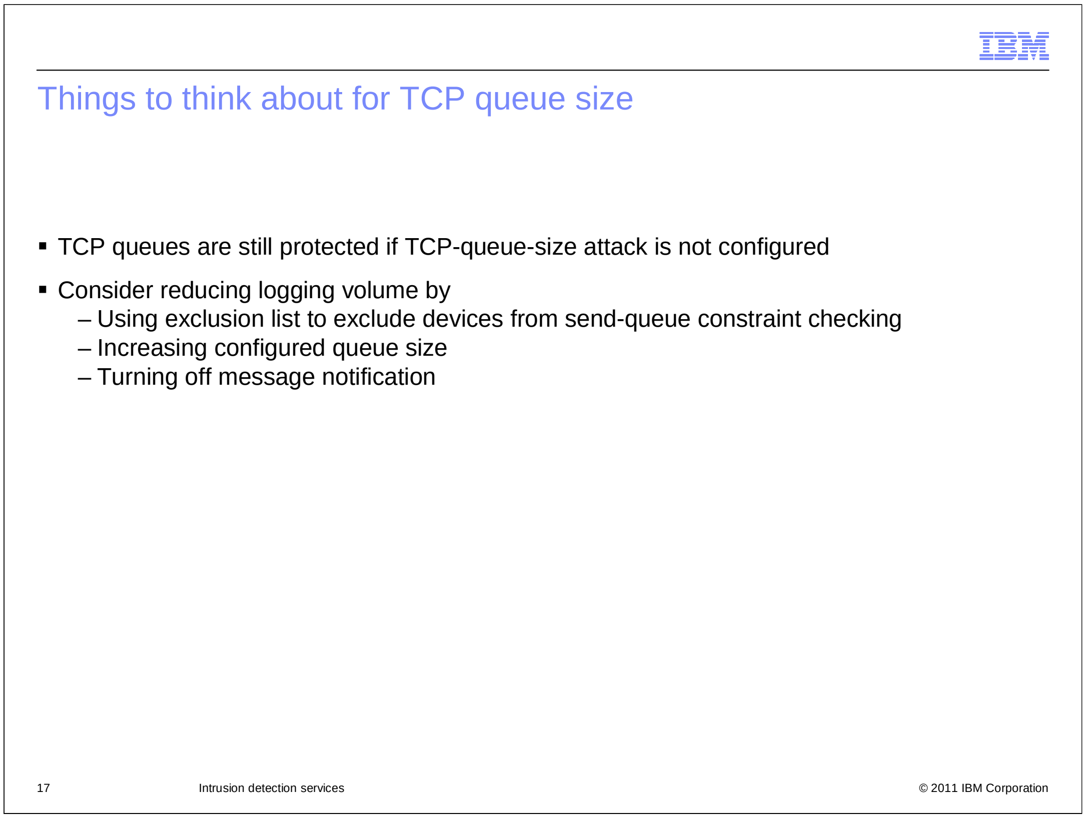
## Migrating your Configuration Assistant backing store

- Default requirement map (IDS_Default)
  - New attack types enabled
  - Report events only (no defensive actions are taken by default)
- Customized requirement map
  - New attack types defined but disabled
    - Modify requirement map to take advantage of new attack types
  - New ICMPv6 protocol defined
    - Modify requirement map to take advantage of ICMPv6 scan detection

If you use the Configuration Assistant to generate your IDS policy, there are several things to consider when you migrate your backing store to V1R13.

If you are using the IDS default requirement map (IDS_Default), be aware that this IBM-supplied requirement map has been updated to enable all of the new attack types in a notification mode.

If you are using a customized requirement map, all new attack types are disabled. You must modify your requirement map to take advantage of the new attack type support. You must also modify your requirement map to take advantage of ICMPv6 scan detection.

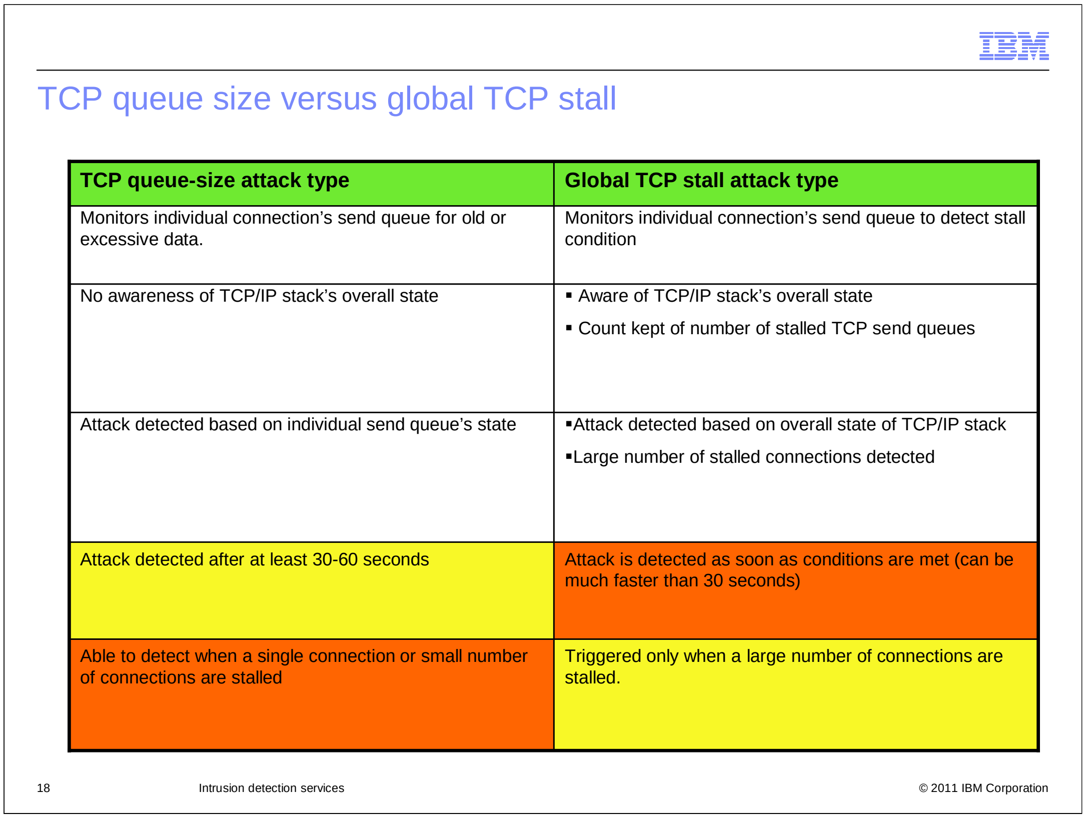


## Things to think about for TCP queue size

- TCP queues are still protected if TCP-queue-size attack is not configured
- Consider reducing logging volume by
  - Using exclusion list to exclude devices from send-queue constraint checking
  - Increasing configured queue size
  - Turning off message notification

If you do not have the TCP-queue-size attack type configured, the TCP queues are still protected. In this case, when a constraint is detected, the queued data is marked page eligible and a syslogd message is issued to indicate the connection’s queue is constrained.

If you find that the TCP-queue-size protection generates an excessive number of log messages on your system, and you have determined that there is no problem, you can consider one of these actions. First, you can enable the new TCP-queue-size attack type and use the exclusion list to eliminate trusted IP addresses from causing constrained and unconstrained messages for the send queue. Second, you can enable the new TCP-queue-size attack type and increase the queue size used to determine constraints. The default queue size is Short but you can increase it if needed. Finally, if the existing protection of the TCP queues meets your installation’s needs but you do not want queue constrained messages to be logged, you can enable this new attack type and disable notification.



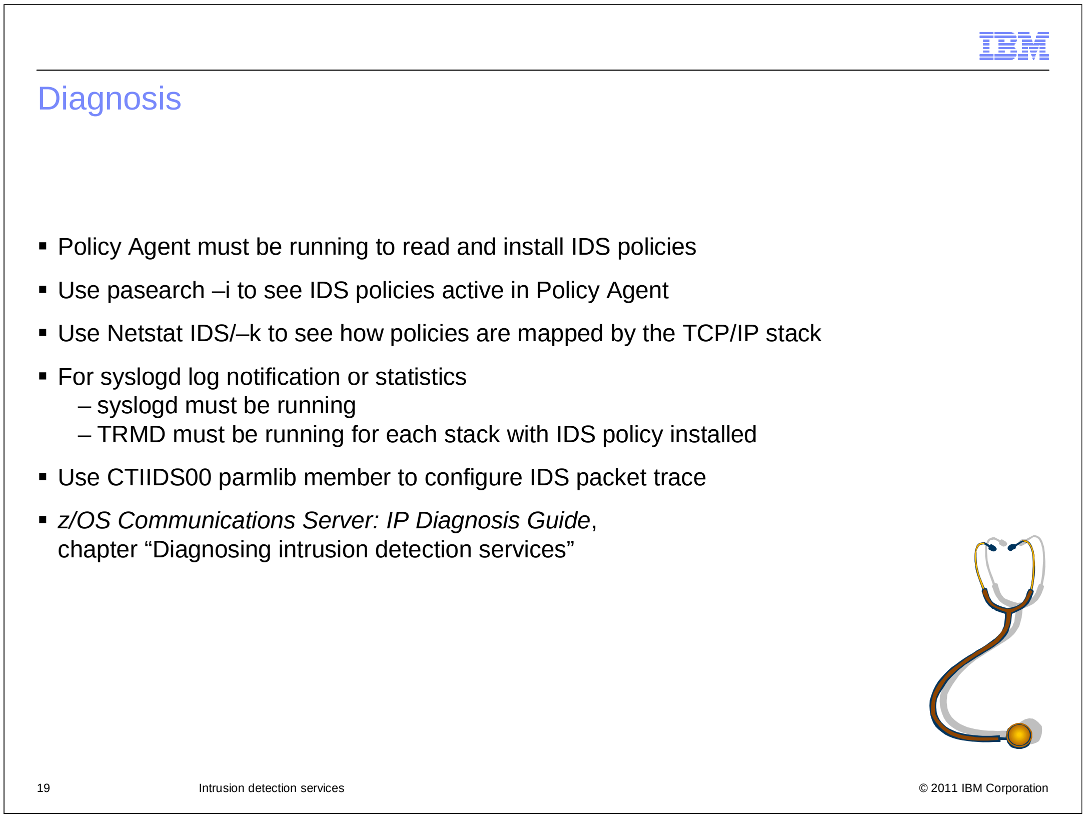
## TCP queue size versus global TCP stall

| TCP queue-size attack type | Global TCP stall attack type |
|---|---|
| Monitors individual connection’s send queue for old or excessive data. | Monitors individual connection’s send queue to detect stall condition |
| No awareness of TCP/IP stack’s overall state | ▪ Aware of TCP/IP stack’s overall state<br>▪ Count kept of number of stalled TCP send queues |
| Attack detected based on individual send queue’s state | ▪Attack detected based on overall state of TCP/IP stack<br>▪Large number of stalled connections detected |
| Attack detected after at least 30-60 seconds | Attack is detected as soon as conditions are met (can be much faster than 30 seconds) |
| Able to detect when a single connection or small number of connections are stalled | Triggered only when a large number of connections are stalled. |

Both the TCP-queue-size attack type and the global-TCP-stall attack type are sensitive to a stalled TCP send queue. The stack-wide count of stalled TCP connections is incremented as soon as a connection cannot send data. If the send window quickly opens, the data is sent, the count is decremented and the TCP-queue-size attack type is never triggered. If, however, the connection remains stalled for at least 30-60 seconds, the TCP-queue-size attack is triggered.

The global-TCP-stall attack type is able to detect an attack where a large number of connections are stalled. This can be detected before the 30-60 seconds needed for the TCP-queue-size attack type to detect that each individual connection’s queue has old or excessive data.

The TCP-queue-size attack type is able to detect when a single connection (or a small number of connections) is stalled. The TCP-queue-size attack type reacts to old and excessive data on the send, receive and out-of-order queues.

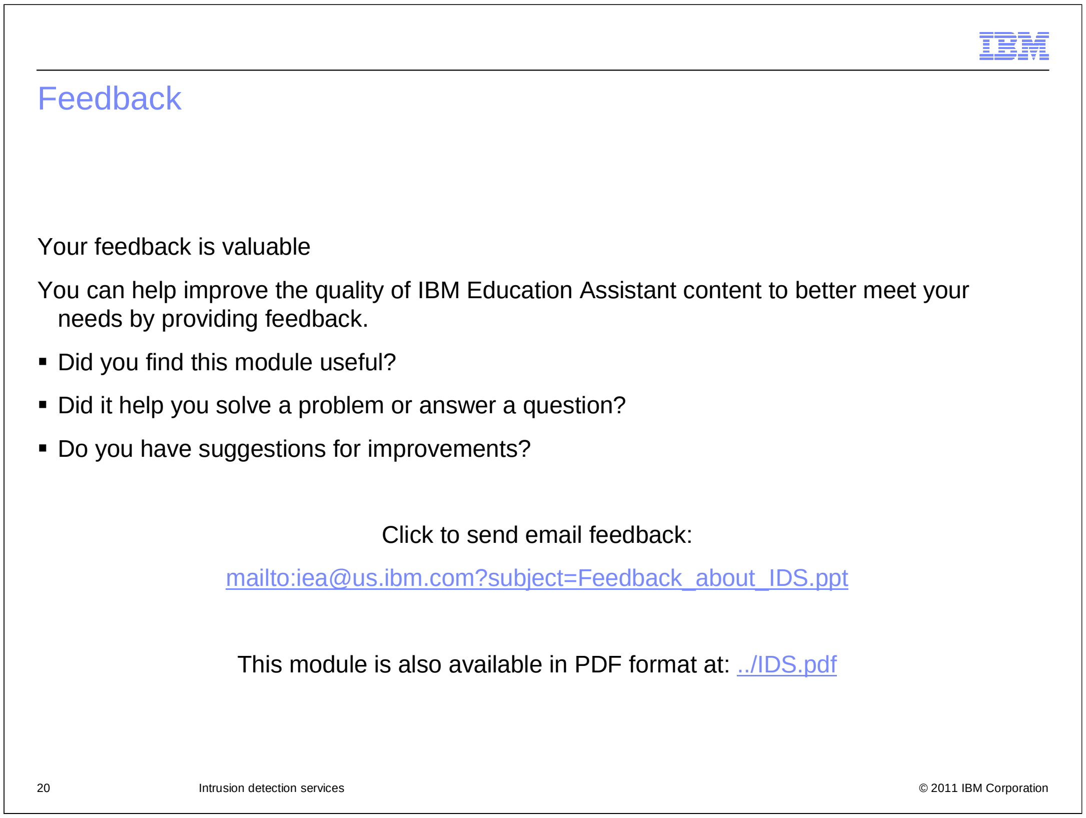


## Diagnosis

- Policy Agent must be running to read and install IDS policies
- Use pasearch –i to see IDS policies active in Policy Agent
- Use Netstat IDS/–k to see how policies are mapped by the TCP/IP stack
- For syslogd log notification or statistics
  - syslogd must be running
  - TRMD must be running for each stack with IDS policy installed
- Use CTIIDS00 parmlib member to configure IDS packet trace
- *z/OS Communications Server: IP Diagnosis Guide*, chapter “Diagnosing intrusion detection services”

The infrastructure for IDS is the same as in previous releases. The same basic components must be active: Policy Agent, syslogd, TRMD, TCP/IP stack. You can use pasearch and Netstat to display policy data.

The “Diagnosing intrusion detection services” chapter in the *z/OS Communications Server: IP Diagnosis Guide* provides information on diagnosing several problems.



## Feedback

Your feedback is valuable

You can help improve the quality of IBM Education Assistant content to better meet your needs by providing feedback.

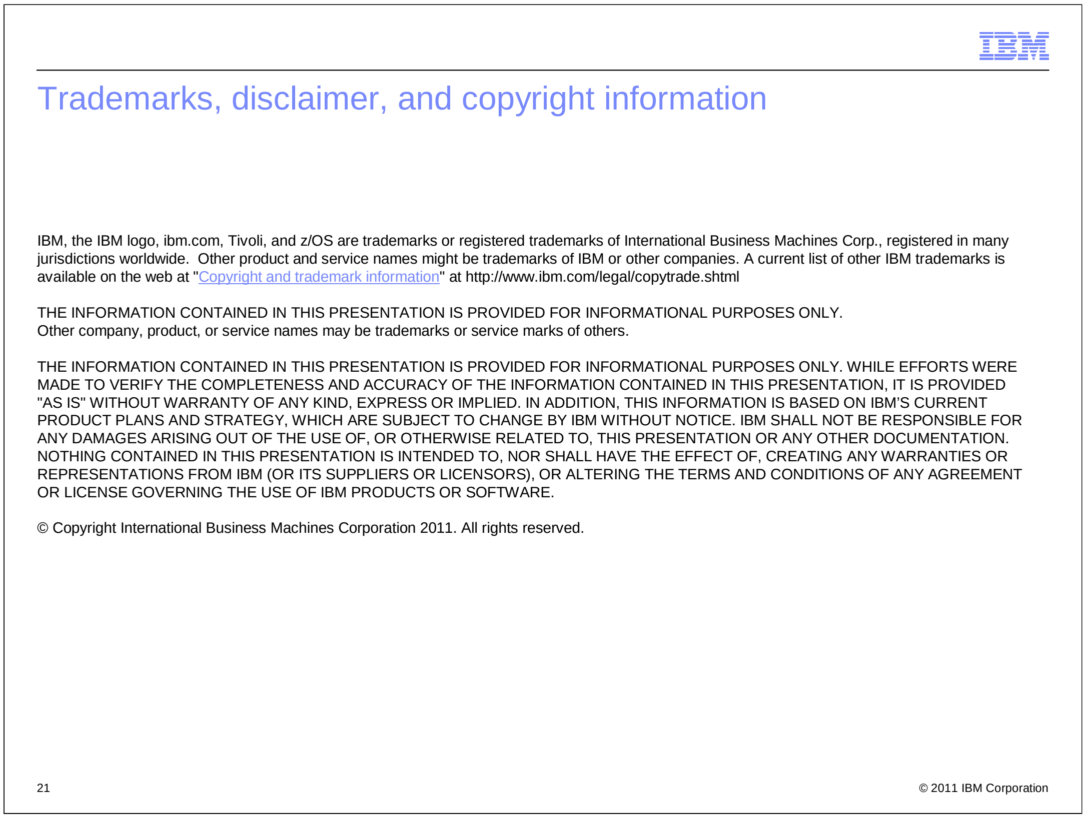
- Did you find this module useful?
- Did it help you solve a problem or answer a question?
- Do you have suggestions for improvements?

Click to send email feedback:

mailto:iea@us.ibm.com?subject=Feedback_about_IDS.ppt

This module is also available in PDF format at: ../IDS.pdf

You can help improve the quality of IBM Education Assistant content by providing feedback.



## Trademarks, disclaimer, and copyright information

IBM, the IBM logo, ibm.com, Tivoli, and z/OS are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of other IBM trademarks is available on the web at "Copyright and trademark information" at http://www.ibm.com/legal/copytrade.shtml

THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY.
Other company, product, or service names may be trademarks or service marks of others.

THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY. WHILE EFFORTS WERE MADE TO VERIFY THE COMPLETENESS AND ACCURACY OF THE INFORMATION CONTAINED IN THIS PRESENTATION, IT IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. IN ADDITION, THIS INFORMATION IS BASED ON IBM'S CURRENT PRODUCT PLANS AND STRATEGY, WHICH ARE SUBJECT TO CHANGE BY IBM WITHOUT NOTICE. IBM SHALL NOT BE RESPONSIBLE FOR ANY DAMAGES ARISING OUT OF THE USE OF, OR OTHERWISE RELATED TO, THIS PRESENTATION OR ANY OTHER DOCUMENTATION. NOTHING CONTAINED IN THIS PRESENTATION IS INTENDED TO, NOR SHALL HAVE THE EFFECT OF, CREATING ANY WARRANTIES OR REPRESENTATIONS FROM IBM (OR ITS SUPPLIERS OR LICENSORS), OR ALTERING THE TERMS AND CONDITIONS OF ANY AGREEMENT OR LICENSE GOVERNING THE USE OF IBM PRODUCTS OR SOFTWARE.