IBM WW Z Security Conference

October 6-9, 2020

z/OS TCP/IP Intrusion Detection Services

Chris Meyer

z/OS Network Security Architect

meyerchr@us.ibm.com

Joshua Bennetone

z/OS Communications Server Developer

jbenneto@us.ibm.com

Agenda

- Function overview
- Events detected
- IDS actions and reports
- Steps for validating IDS policy
- For more information, Q&A

Agenda

- **Function overview**
- Events detected
- IDS actions and reports
- Steps for validating IDS policy
- For more information, Q&A

The intrusion threat

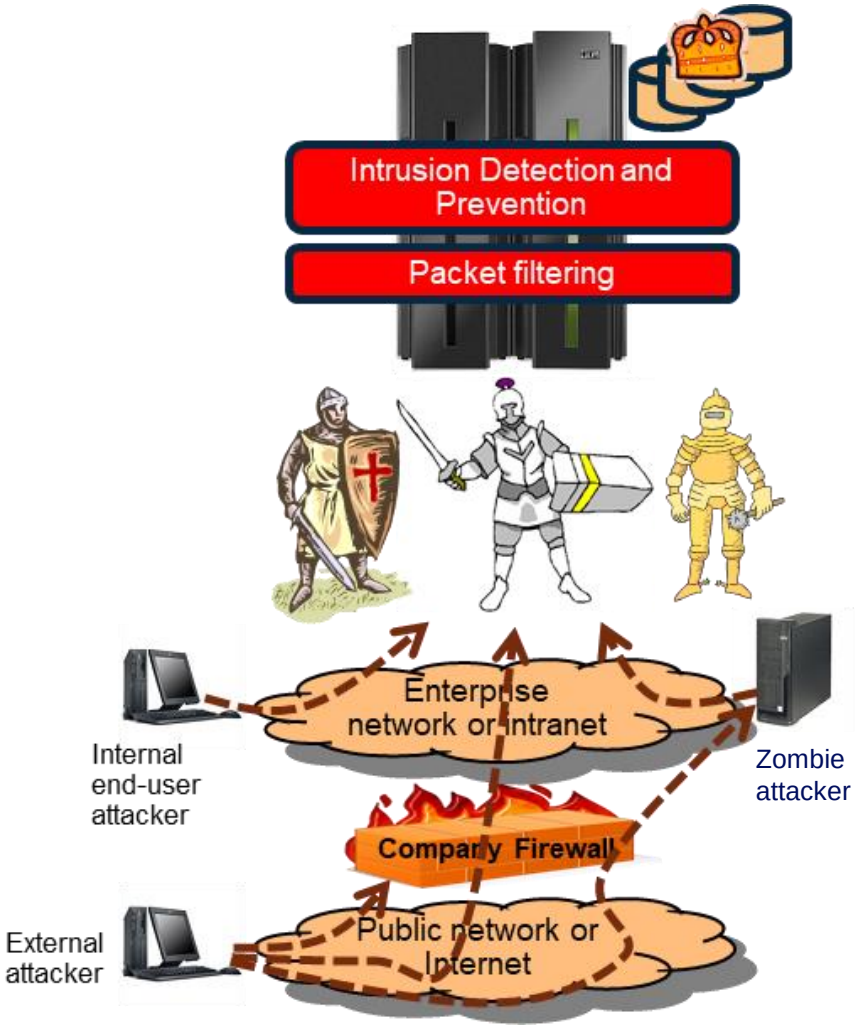
What is an intrusion?

- Information Gathering
  - Network and system topology
  - Data location and contents
- Eavesdropping/Impersonation/Theft
  - On the network/on the host
  - Base for further attacks on others through Amplifiers, Robots, or Zombies
- Denial of Service - Attack on availability
  - Single packet attacks - exploits system or application vulnerability
  - Multi-packet attacks - floods systems to exclude useful work

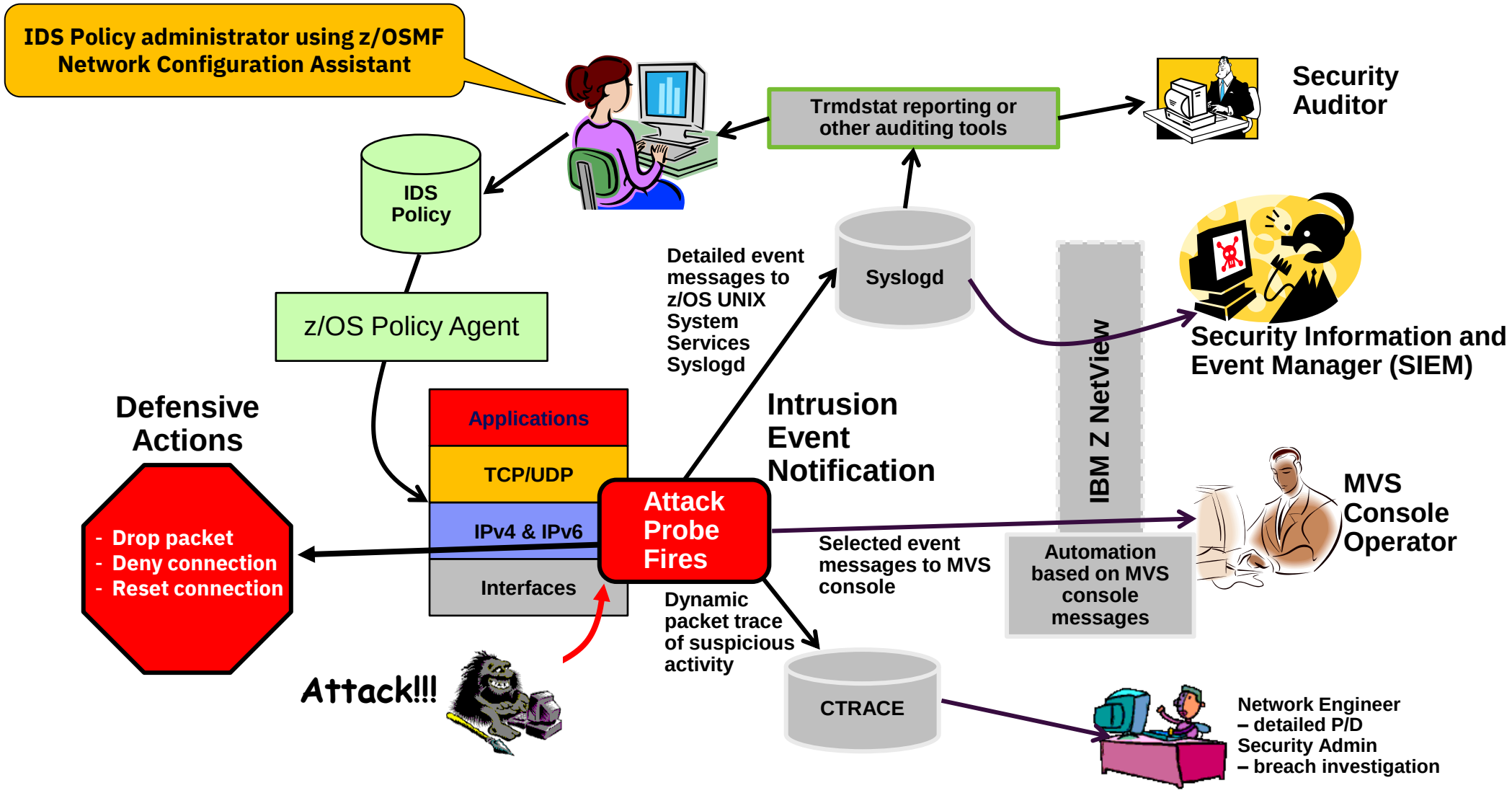
Attacks can occur from Internet or intranet

- Company firewalls and intrusion prevention appliances can provide some level of protection from Internet
- Perimeter security strategy alone may not be sufficient.
  - Some access is permitted from Internet – typically into a Demilitarized Zone (DMZ)
  - Trust of intranet

Attacks can be intentional (malicious) but often occur as a result of errors on nodes in the network (config, application, etc.)



z/OS TCP/IP IDS overview



z/OS TCP/IP IDS features

IDS Events

- **Scans – attempts by remote nodes to discover information about the z/OS system**

- **Attacks – numerous types**

  - **Malformed packets**
  - **IP option and IP protocol restrictions**
  - **Specific usage ICMP**
  - **Interface and TCP SYN floods**
  - **and so forth…**

- **Traffic Regulation**
  - **TCP - limits the number of connections any given client can establish**
  - **UDP – limits the length of data on UDP queues by port**

Defensive actions

- Packet discard
- Limit connections
- Drop connections

Reporting

- Logging
- Console messages
- IDS packet trace
- Notifications to external event managers (like IBM Z NetView and SIEMs)

z/OS in-context IDS complements network-based intrusion detection/prevention:

- *Does not replace network-based IDS/IPS!*
- In-context means – z/OS IDS operates as the communications endpoint, not as an intermediary
- Applies to networking protocols only – does not interrogate message payloads (application data) like network IDS devices
- Can evaluate some inbound encrypted data - IDS applied after decryption on z/OS – especially good for IPsec ESP protection
- IDS attack probes are part of protocol processing logic – very efficient – not per-packet evaluation against table of known attacks
- Detects statistical anomalies realtime - has stateful data / internal thresholds that are generally unavailable to external IDSs

Agenda

- Function overview
- **Events detected**
- IDS actions and reports
- Steps for validating IDS policy
- For more information, Q&A

z/OS TCP/IP IDS event types

Scans
- Identify potential attack vectors on the target system
- Search for things like open ports, addresses, subnet structure, software versions, etc.

Attacks
- Attempt to impact availability of an application or the system
- Could be single or multiple packets

Traffic regulation
- Protect against over-consumption of TCP connections and/or UDP queues
- From either malicious activities or unexpected peak loads

Scans: Prelude to an attack

- z/OS definition of a scanner
  - Source host that accesses multiple unique resources (ports or interfaces) over a specified period of time
  - Number of unique events (threshold) and time period (interval) are configurable in IDS policy

- Scan categories
  - Fast scan: many resources accessed in a short period of time (less than 5 minutes). Program driven.
  - Slow scan: different resources access intermittently over longer period of time (many hours). Used to avoid detection.

- Scan event types
  - ICMP, ICMPv6 scans
  - TCP port scans
  - UDP port scans

Scans: Scan policy allows you to…

- Obtain notification and documentation of scan activity
  - Console and syslogd messages
  - Trace potential scan packets
- Define scan event parameters
  - Fast and slow scan intervals
  - Threshold for triggering IDS scan events
- Reduce number of false positives
  - Exclusion lists for “known scanners”
  - Sensitivity levels for different event types

Scans: Event counting and sensitivity

- Each event is internally classified as normal, suspicious or very suspicious (see z/OS Communications Server IP Configuration Guide for details)

- Sensitivity determines whether an event is “countable”

Sensitivity from policy	Normal event	Suspicious event	Very suspicious event
Low			Count
Medium		Count	Count
High	Count	Count	Count

- Scan events are counted against the source IP address. If the number of counted events reaches threshold value, a scan event is triggered and policy determines actions

```
Feb 20 16:47:39 EVILMF TRMD.TCPIP[50397191]: EZZ8643I TRMD SCAN threshold exceeded:02/20/2020 16:47:22.56,sipaddr=172.
30.0.234,scantype=F,pthreshold=5,pinterval=1,vs=0,ps=10,norm=0,correlator=5,probeid=0300FFF1,sensorhostname=EVILMF.EVI
LMAINFRAME.COM
```

z/OS TCP/IP IDS event types

Scans
- Identify potential attack vectors on the target system
- Search for things like open ports, addresses, subnet structure, software versions, etc.

Attacks
- Attempt to impact availability of an application or the system
- Could be single or multiple packets

Traffic regulation
- Protect against over-consumption of TCP connections and/or UDP queues
- From either malicious activities or unexpected peak loads

Attacks: TCP/IP stack defenses vs. IDS

- The TCP/IP stack **always silently defends itself** against many attacks
- IDS allows you to…
  - control recording of intrusion events and to provide supporting documentation
  - detect and disable uncommon or unused features which could be used as an attack vector
  - in some cases, specify additional defensive actions

Attacks: TCP/IP attack categories

- Malformed packets (incorrect or partial IPv4 or IPv6 packet headers)
- Inbound fragment restrictions (attempts to create invalid IP packets by manipulating IP fragmentation)
- IPv4 and IPv6 protocol restrictions (detect use of unexpected IP protocols)
- IPv4 and IPv6 option restrictions (detect use of unexpected IP options)
- ICMP, ICMPv6 redirect restrictions (detect attempts to modify routing tables)
- UDP perpetual echo (detect attempts to exploit known UDP applications that unconditionally respond to every inbound datagram)
- Outbound RAW socket restrictions (detect application-crafted invalid outbound packets)
- Flood events
  - Detect SYN floods from spoofed remote addresses
  - Detect high volume of discarded packets on physical IPv4 and IPv6 interfaces
- Data hiding (detect attempt to “leak” data inside of IP packet header and extension fields)
- TCP queue size (detect queue size constraints for individual connections)
- TCP global stall (detect cases where large number and percentage of TCP connections are stalled)
- Enterprise Extender specific attacks:
  - Malformed packets
  - LDLC check
  - Port check
  - EE XID flood

Attacks: Attack policy allows you to…

- Control attack detection for one or more attack categories independently
- Generate notification and documentation of attacks
  - Console and syslogd messages
  - Trace potential attack packets
- Generate attack statistics on time interval basis (normal or exception)
- In some cases, control defensive action in case of attack

Attacks: Example: Interface flood detection (1 of 3)

- A high percentage of discarded packets may indicate that a physical interface or the host it belongs to is under attack.
  - A packet discarded *by the TCP/IP stack -- for any reason* -- will count against the flood threshold
  - The ability of the interface to keep up with traffic is not a factor

- Notification and (optionally) traces are generated in this case

- Information provided:
  - Interface under attack (so you can take defensive action)
  - Source MAC of the prior hop for OSA QDIO and LCS devices
  - Source IP address from outer IPsec header if packet was protected by tunnel-mode IPsec (could help narrow source closer than prior hop if source address is a gateway or firewall)

Attacks: Example: Interface flood detection (2 of 3)

- Flood attack policy specifies two attributes:
  - Minimum number of discarded packets (default 1000)
  - Discard percentage (default 10%)

- On a per-physical interface basis:
  - IF
    - the minimum number of discards is reached within a one minute interval AND
    - The discard rate (`discards_during_interval / inbound_packets_during_interval`) meets or exceeds configured discard percentage

    THEN an interface flood condition is raised
  - Once a flood condition is raised, flood statistics are computed and reported at one minute intervals
  - Flood condition ends when:
    - Discards for a subsequent interval fall below minimum OR
    - Discard rate for a subsequent interval is less than or equal to ½ of configured percentage

Attacks: Example: Interface flood detection (3 of 3)

Example: Assume that interface flood policy specifies
Minimum Discards = 2000 and Interface Flood Percentage = 10%

time

Time Interval	Inbound Count	Discard Count	Discard Rate	Notes
> 1 minute	13,000	2,000	N/A	Took longer than 1 minute to reach minimum discard count, so not a flood
< 1 minute	30,000	2,000	6.6%	Discard rate is < configured percentage (10%), so not a flood
< 1 minute	20,000	2,000	10%	**Interface flood condition raised**. Begin collecting flood statistics at 1 minute intervals until flood condition ends.
+1 minute	40,000	3,000	7.5%	Flood condition continues, reset 1 minute interval timer and continue collecting statistics.
+1 minute	50,000	2,500	5%	**Interface flood condition ends**. Discard rate is ½ of that specified in policy.

z/OS TCP/IP IDS event types

Scans
- Identify potential attack vectors on the target system
- Search for things like open ports, addresses, subnet structure, software versions, etc.

Attacks
- Attempt to impact availability of an application or the system
- Could be single or multiple packets

Traffic regulation
- Protect against over-consumption of TCP connections and/or UDP queues
- From either malicious activities or unexpected peak loads

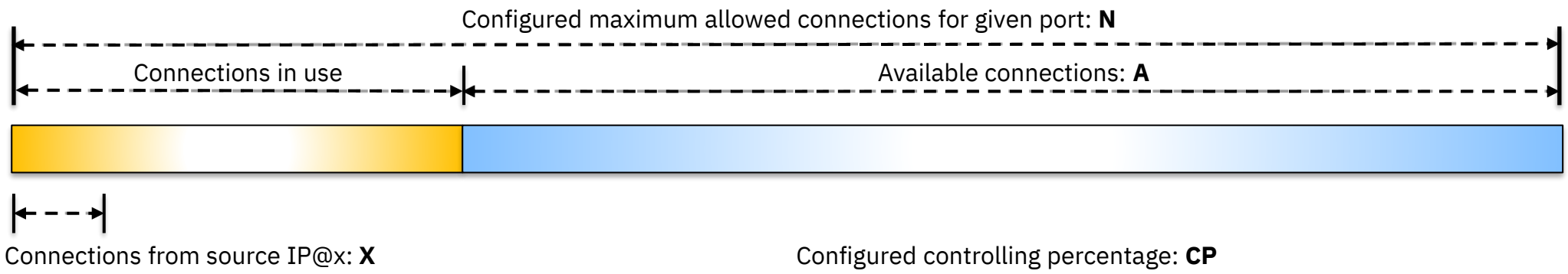
TCP Traffic Regulation (1 of 3)

- Controls number of inbound connections from a single host
  - Can be specified on a per-application (port) basis
  - Allows independent policies for applications sharing a port (like telnetd and TN3270)

- Connection limit expressed as
  - Port limit for all connecting hosts AND
  - Individual limit for a single connecting host

- “Fair share algorithm”
  - Based on percentage of available connections
  - Each host allowed at least one connection as long as port limit not reached
  - QoS connection limit overrides TCP TR – useful for concentrator sources like web proxy servers

TCP Traffic Regulation (2 of 3)



If a new connection request is received and:

- A=0, the request is rejected
- A>0 and the request is from a source that does NOT have an existing connection to the port, allow the connection
- A>0 and the request is from a source that already has connections with this port (IP@x in this example), then:

If X+1 < CP * A then
  Allow the new connection
Else
  Deny the new connection

Goal: The number of connections allowed to any given source IP address shrinks as the connection limit is approached.

TCP Traffic Regulation (3 of 3)

Example: Source address IP@x has four connections. It is now attempting its fifth connection

Scenario	Total Allowed	Existing Connections	Available Connections	Connections allowed by CP=20%	Allowed?
A	100	60	40	8	
B	100	80	20	4	

A If we currently have 40 connections available (A=40) and a controlling percentage of 20% (CP=20%), when IP@x tries to establish its fifth connection, it will be allowed: 40 * 20% = 8, so 5 connections is less than the regulated limit.

B If we have 20 connections available (A=20) and CP is still at 20%, when IP@x tries to establish its fifth connection, it will be rejected: 20 * 20% = 4, so 5 connections would exceed the regulated limit.

UDP Traffic Regulation

- Controls allowable length of inbound UDP receive queues on a per-application (port) basis
- UDP TR policy supersedes UDPQueueLimit in TCP/IP profile (global limit for all UDP queues)
- If neither UDP TR or UDPQueueLimit are used, a stalled application or a flood on a single UDP port could consume all available buffer storage
- Queue limits expressed in abstract terms:
  - SHORT or VERY SHORT for applications that tend to receive data faster than they can process it
  - LONG or VERY LONG for fast or high priority applications with bursty arrival rates

Agenda

- Function overview
- Events detected
- **IDS actions and reports**
- Steps for validating IDS policy
- For more information, Q&A

Actions: Recording actions

- Controlled by IDS policy action specification. Each of the following options independently selectable
  - Event logging
    - syslogd – number of events recorded in 5 minute interval can be limited per attack type (for most attack types)
    - Console – recording suppressed if number of console messages reaches thresholds specified in policy
  - Statistics – written to syslogd for normal or exception conditions (configurable)
  - IDS packet trace
    - Activated *after* an attack is detected
    - Standard packet trace format, but for suspected attack packets
    - Number of packets traced is limited
    - Amount of data traced is configurable (header, full, byte count)

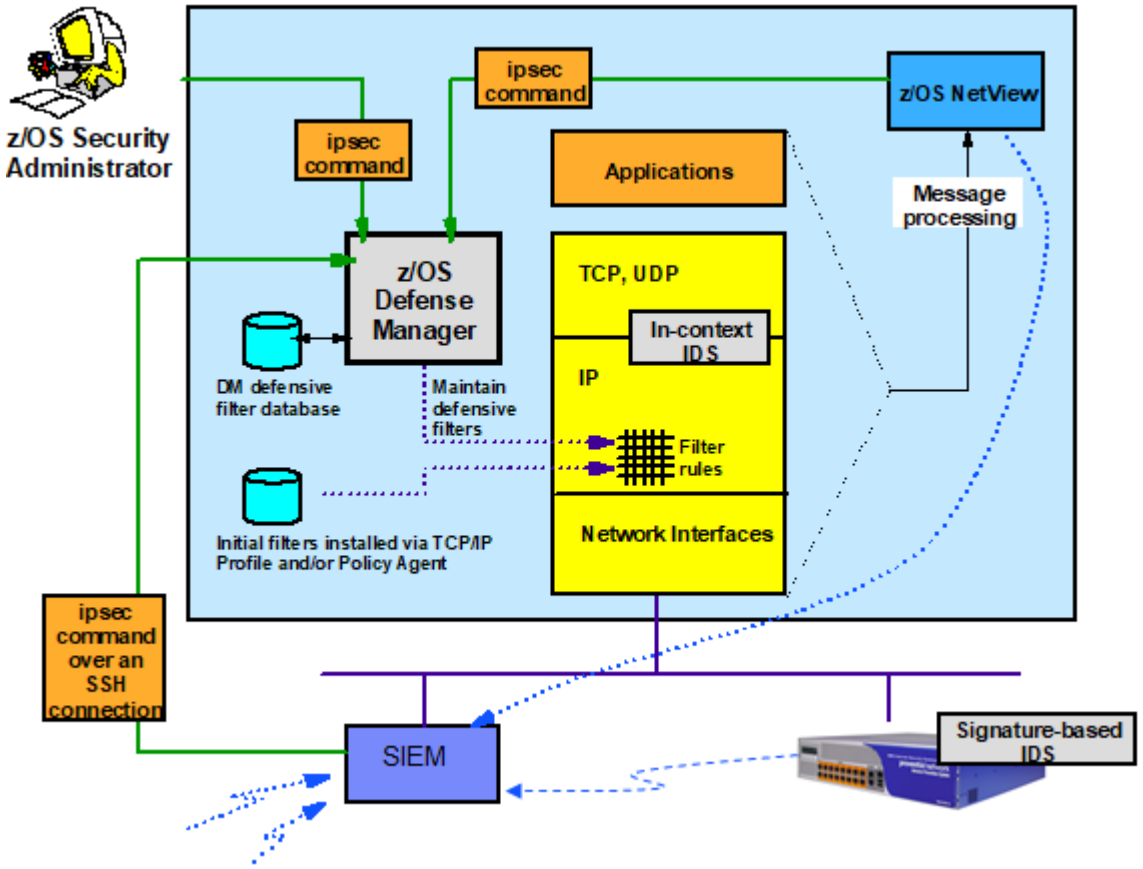
- All recorded IDS events contain a *probeid* and *correlator*
  - *probeid* indicates point at which the event was detected
  - *correlator* allows association of related syslog, console and packet trace records

Actions: Configurable defensive actions by event type

- Attack events
  - Packet discard
    - These events ALWAYS result in packet discard, regardless of IDS policy action
      - Malformed packets
      - TCP SYN floods
    - Discard for most attack events controlled by IDS policy action
      - ICMP redirect restrictions
      - IPv4, IPv6 option restrictions
      - IPv4, Ipv6 protocol restrictions
      - IP fragmentation
      - Outbound raw restrictions
      - UDP perpetual echo
      - Data hiding
      - EE malformed, LDLC and port checks
  - Reset connection
    - TCP queue size
    - TCP global stall
  - No defensive action defined for interface floods beyond what TCP/IP always does to protect itself
- Scan events
  - No defensive actions defined
- Traffic Regulation events
  - TCP – connection limiting
  - UDP – packet discard

Defense Manager for dynamic defensive filtering

- Defensive filters enable dynamic defensive actions in case of attack
- NOT policy-based: Created, managed and controlled through the ipsec command
- NOT part of IDS, but can be used within automation for IDS event processing
- DENY only (but also “simulate mode”)
- Installed “in front of” all other IP filters
- Maintained on DASD to protect restarted stacks from the time they come up
- Limited lifetime (~2 weeks max)
- Selectable scope:
  - Local – applies to a specific stack
  - Global – applies to all stacks on LPAR
- One Defense Manager Daemon per LPAR



Reports: IDS log reports

- trmdstat command produces reports based on IDS data recorded through syslogd
- Types of reports for logged events:
  - Overall summary reports for IDS
  - Event type *summary* reports for Attacks, Floods, Scans and Traffic Regulation (TCP and UDP)
  - Event type *detail* reports for the same
- For logged statistics, detail reports are available for Attacks, Floods, and Traffic Regulation (TCP and UDP)

Reports: IBM Z NetView support for IDS events

- IBM Z NetView supports Comm Server IDS events
- Traps IDS messages to z/OS console or syslogd
- Can take predefined actions based on event type
  - Route IDS messages to designated NetView consoles
  - e-mail notification to security administrator
  - Run trmdstat and attach output to e-mail
  - Use ssh to issue ipsec command to enable dynamic defensive filters

Agenda

- Function overview
- Events detected
- IDS actions and reports
- **Steps for validating IDS policy**
- For more information, Q&A

Steps for deploying and validating IDS policy

Tip: The z/OS Communications Server Network Configuration Assistant provides a very good initial set of IDS rules on which you can build

1. Configure policy for reporting actions only (no defensive actions)
2. Install policy on target z/OS system (recommend using default IDS policy)
3. Start pagent, syslogd and TRMD
4. Issue pasearch command to verify the correct policy is installed
5. Keep policy active for a trial period
6. Issue NETSTAT IDS to view active IDS policy and statistics
7. Run trmdstat reports to verify syslog messages for IDS events
8. Adjust the IDS policy as appropriate
9. Add defensive actions if necessary

Agenda

- Function overview
- Events detected
- IDS actions and reports
- Steps for validating IDS policy
- **For more information, Q&A**

For more information…

URL	Content
http://www.youtube.com/user/zOSCommServer	IBM Communications Server on YouTube
http://tinyurl.com/zoscsblog	IBM Communications Server blog
https://www.ibm.com/support/knowledgecenter/SSLTBW_2.4.0/com.ibm.zos.v2r4.csf/csf.htm	IBM Communications Server library

Questions?

IBM WW Z Security Conference

October 6-9, 2020

z/OS TCP/IP Intrusion Detection Services

Chris Meyer

z/OS Network Security Architect

meyerchr@us.ibm.com

Joshua Bennetone

z/OS Communications Server Developer

jbenneto@us.ibm.com