HP-UX IPFilter Firewall Dynamic Spam Control

Eric C. Scoredos
Network Architect
Hewlett-Packard

Executive Summary

- HP IPFilter Dynamic Connection Allocation

An IP-address based packet filtering system that can be used as an effective safeguard for mail services and other server systems:

- Dynamically limit connect requests to slow down potential spammers
- Limit configurable per IP address or range of IP addresses
- trusted partners can be configured to bypass connection limitations
- a range of IP addresses can be assigned a cumulative limit

- Based on enhancements to current HP system firewall product- HP IPFilter/9000 B9901AA
  Downloadable from www.software.hp.com
  HP IPFilter is based on the Open Source IPFilter Product Version 3.5

Spam e-mail Problem

- Contents of e-mail headers can be faked.
- Only the IP address of the SMTP packet which enters the receiving Mail Server's domain can be trusted.
- e-mail black lists and white lists need to be based on this IP address
- Improper SMTP relays as well as end-user IP addresses can be detected.
- Also improper behavior such as opening multiple e-mail connections to target SMTP servers can be controlled.

IPFilter DCA Feature Overview

Dynamic Connection Allocation (DCA) Features

- Allows you to control incoming TCP connections which pass through the IPFilter system
- The IPFilter DCA system is an intermediate system, but it can also run on end systems
- Uses stateful packet inspection to control the number of incoming TCP connections.
- Introduces new set of flexible rules [keep limit] for controlling incoming TCP connections.
- The connection allocation is achieved through a limit value which is the maximum number of connections that can be established from the ip sources specified by the rule.

What is IPFilter DCA?

- Extensions to current IPFilter product
- Lets IPFilter administrator add rules to control incoming connections by
  - ip address
  - ip address mask (CIDR)
  - ip address range
  - tcp destination port
  - tcp destination port range
- Adds maximum allowed connection limit count to IPFilter rules
- Rules can be for individual ip addresses or cumulative addresses (encompassing multiple ip source addresses)

DCA (Contd)

- A set of new commands help collect statistics about the connections which are being controlled.

  This includes:
  - o source and destination IP address,
  - o the allocated number of connections,
  - o the number of active connections,
  - o the number of times the allocated quota of connections was exceeded.

- Provides logging records which can serve as alert messages or as a summary of the connections made from an IP address. This helps to fine tune the rules configured by identifying IP addresses or subnets whose allowed concurrent connections could be further restricted or blocked all together.

Concept

Keep It Simple and Secure (KISS)

* ***Dynamically flow control connect requests from unknown sources before they reach the mail servers.***
* ***By rejecting the excessive connect requests, the Spammer will be slowed down..***

- IP address based filtering.
- Configure rules to filter e-mails per IP addresss or range of IP addresses.
- Specific IP addresses or ranges of IP addresses can be configure to by-pass filtering, effectively creating a "white list" of trusted partners
- Allows user to specify a limit for connect requests from unknown source IP addresses.
- Tunable flow control limit per ip address or range of ip addresses
- Allows dynamic update of ip lists being monitored without needing to restart IPfilter.

Advantages

- Simple
- Dynamic
- Effective
- Enhancement to existing IPFilter product.

Concerns:

- Maybe too restrictive. All unknown sources are subject to flow control.
- Cannot differentiate direct spamming from indirect spamming via open relay mail servers.

IPFilter DCA Uses

DCA : where can it be used

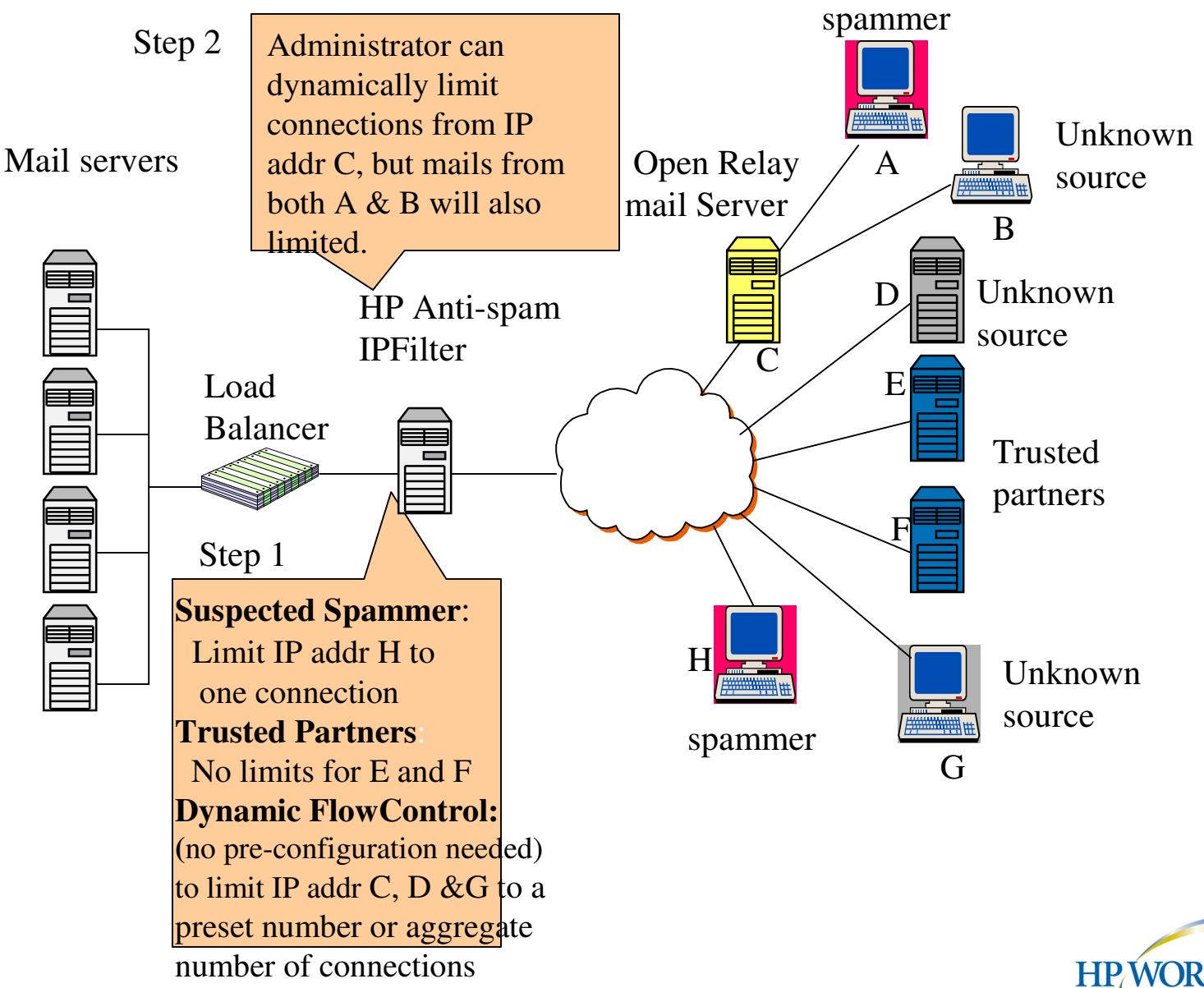
The DCA feature helps protect and mitigate affects of a DOS attack which involve a flood of TCP connections. Some possible uses could include:

1. Protect a mail server from a flood of SMTP connections. IP addresses or subnets from which a flood of connection requests originate (which are trying to flood the SMTP server and tie up its resources) can be slowed down. At the same time, known users can be given an unlimited connection limit. This ensures that customers and partners can still access the mail server.

2. Protect a LDAP server from a flood of bogus SSL connections or any other type connections which try to tie up the LDAP server.

3. Protect a server from Denial of Service "SYN" attacks by restricting the number of connections any one client can make to the servers

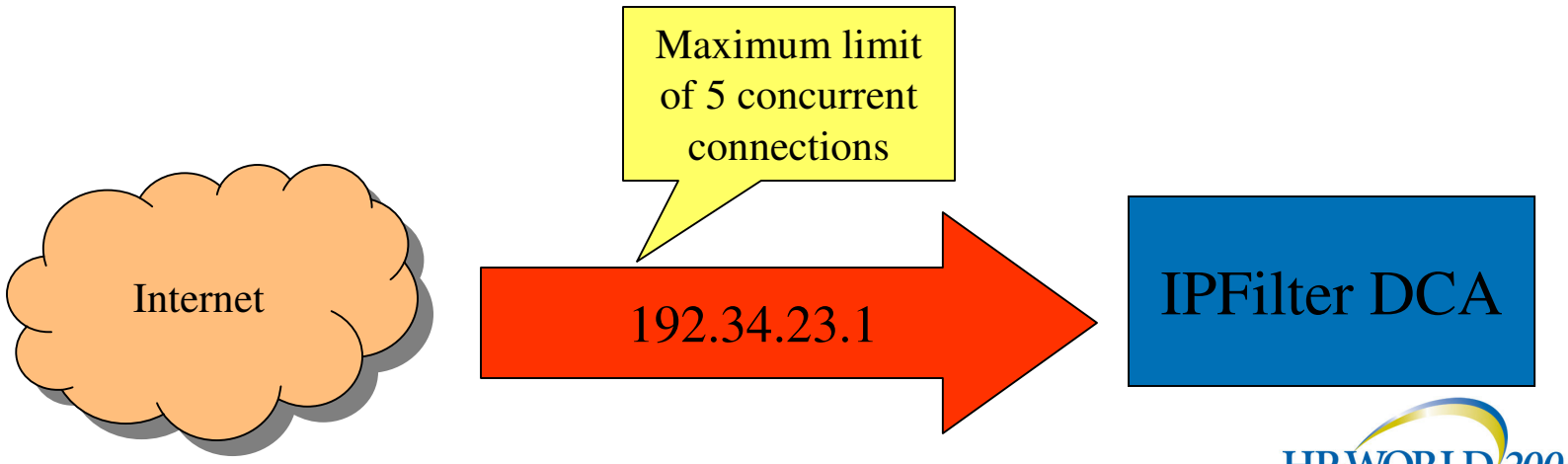
An example of HP IPFilter DCA Configuration



Examples of IPFilter DCA Rules

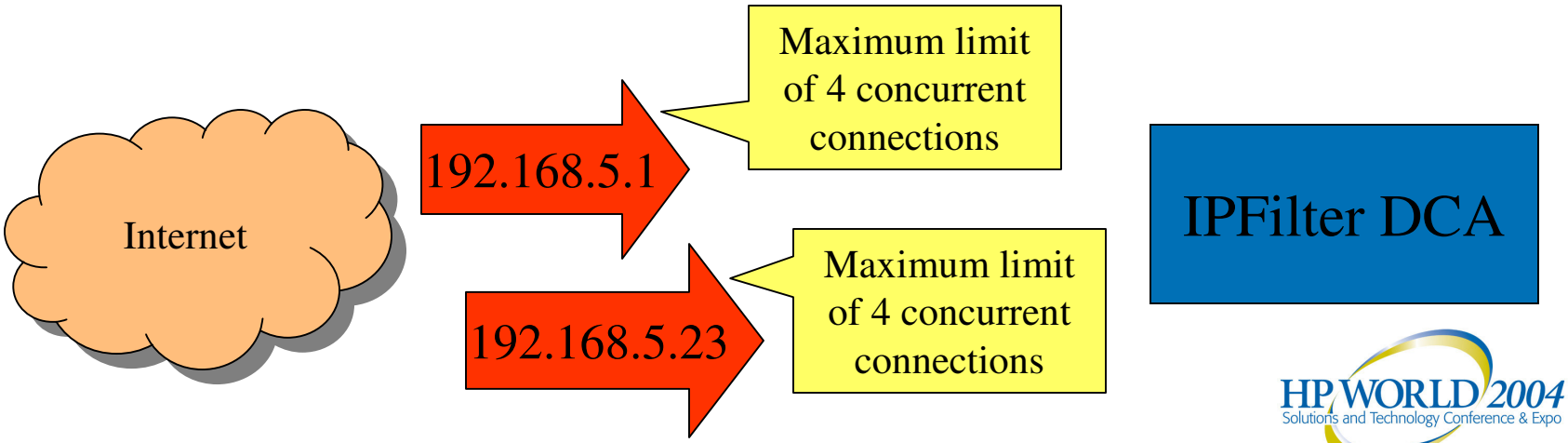
Filtering Rule 1

Individual connection limit per IP address
pass [rtn-reset] in quick proto tcp from 192.34.23.1 to any port = 25 keep limit 5
This rule will limit maximum concurrent connections from host 192.34.23.1 connecting to port 25 to 5. If [rtn-reset] option is specified, reset will be sent for over limit connection.



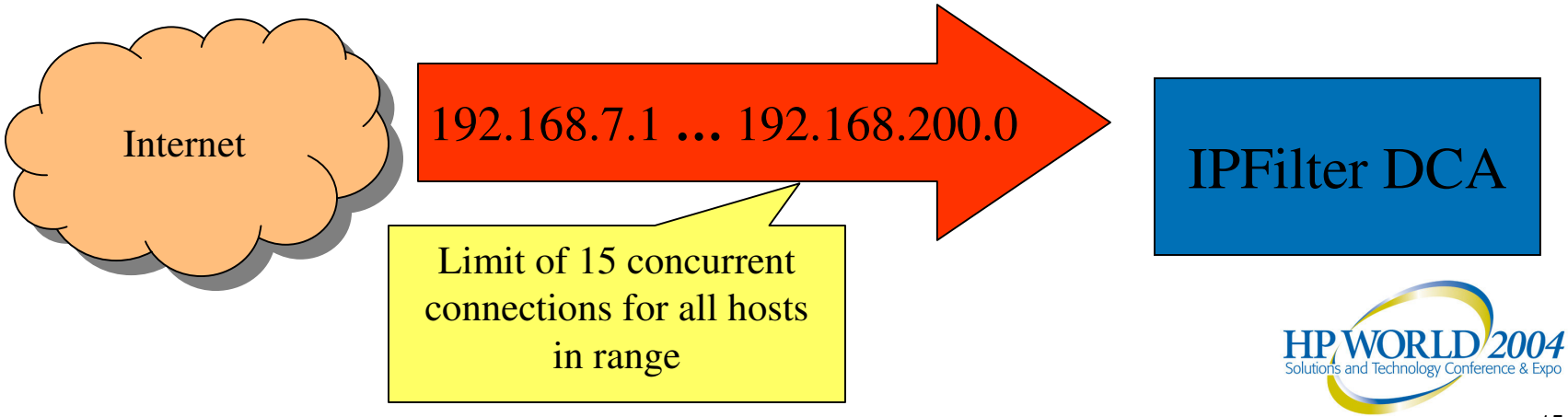
Filtering Rule 2

Connection limit for each individual IP address within a subnet
pass in quick proto tcp from 192.168.5.0/24 to any port = 25 keep limit 4
This rule will limit maximum concurrent connections from any individual host in subnet 192.168.5.0/24 connecting to port 25 to 4



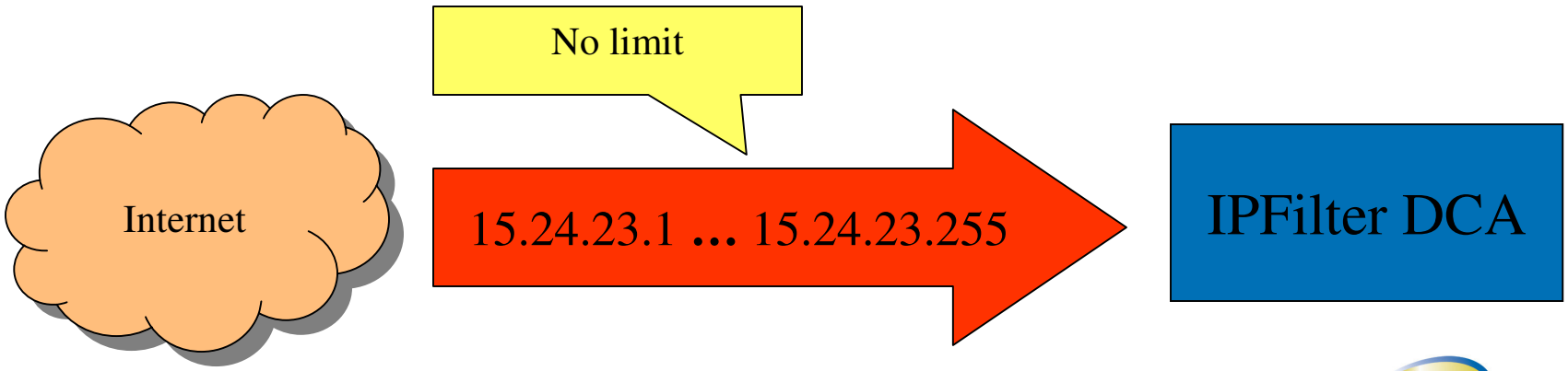
Filtering Rule 3

Connection limit for all IP addresses within an IP address range (cumulative total)
pass in quick proto tcp from 192.168.7.0-192.168.200.0 to any port = 25 keep limit 15 cumulative
This rule will limit cumulative concurrent connections from all hosts in the IP address range 192.168.7.0 – 192.168.200.0 connecting to destination port 25 to 15



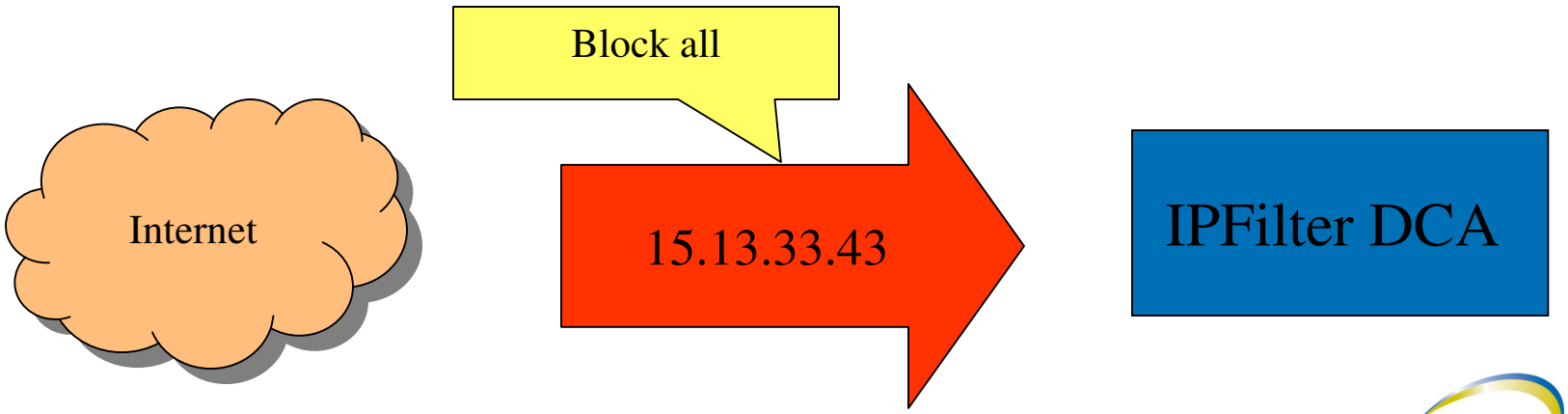
Filtering Rule 4

Unrestricted Hosts
pass in quick proto tcp from 15.24.23.0/24 to any port = 25 keep state
This rule will pass all connections from any hosts in subnet 15.24.23.0/24 connecting to port 25.



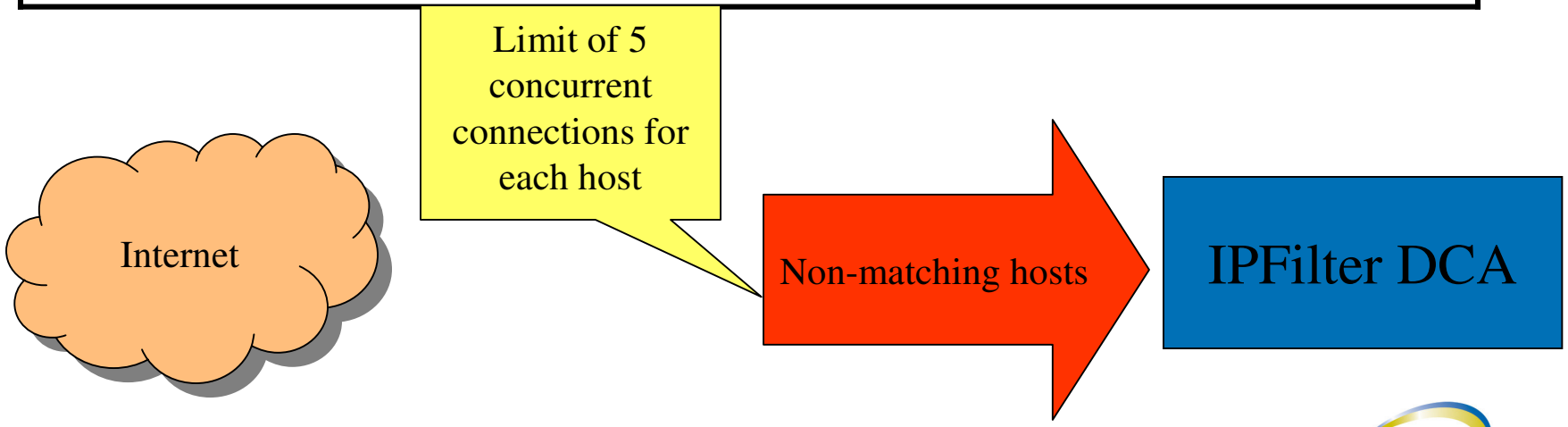
Filtering Rule 5

Fully Restricted Hosts
block in quick from 15.13.33.43 to any port = 25
This rule will block all connections from 15.13.33.43 connecting to port 25.



Filtering Rule 6

Default Limit for Unknown Hosts
pass in proto tcp from any to any port = 25 keep limit 5
This rule will specify default connection limit for other non-matching hosts connecting to port 25 to 5 [should be the next to last rule in the configuration file ("block in from any to any" is usually the last rule]



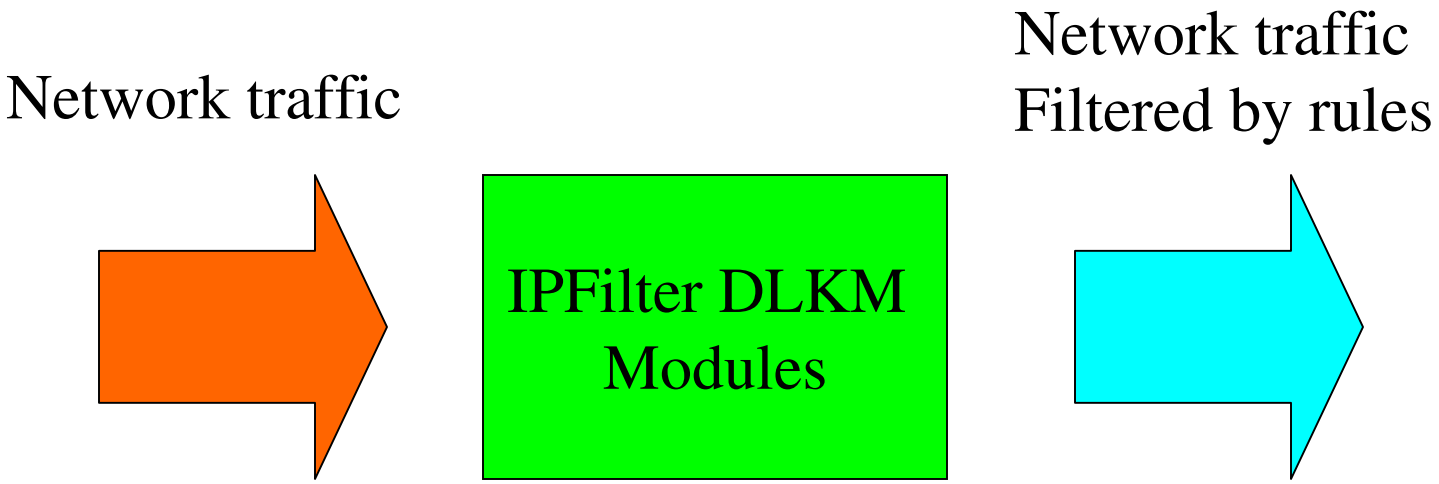
DCA Rules Syntax

The following is the complete syntax for creating DCA rules

```
pass [return-rst] in [log limit [freq <num>]] quick \

proto tcp \

from <ip | ip_subnet | ip_range | any > \

to <ip | ip_subnet | ip_range | any>  [port = port_num] \

keep limit <num> \

[cumulative]
```

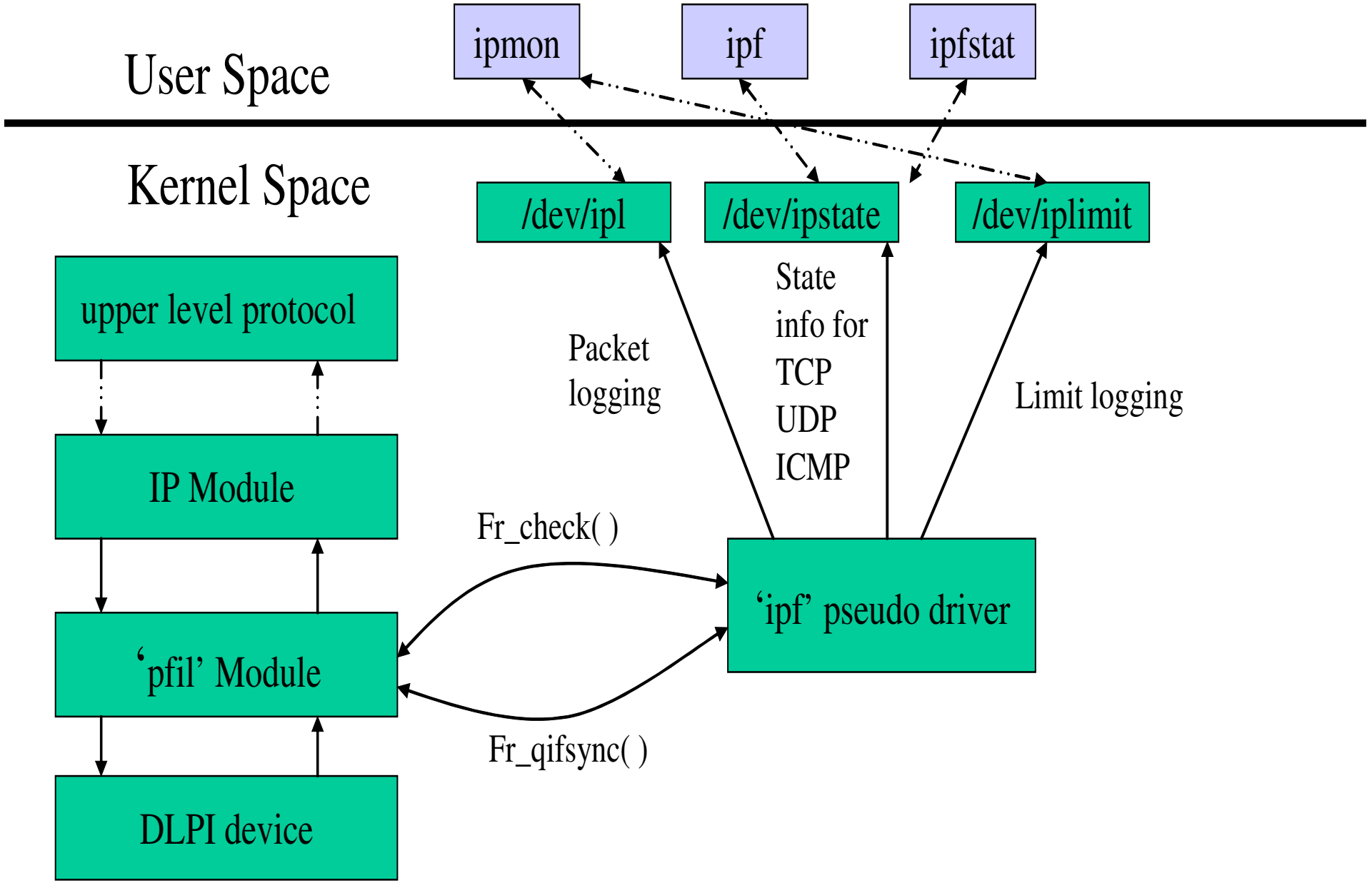
IPFilter DCA Architecture

Design of HP IPFilter DCA Filter



IP Packets forwarded/dropped
before entry to network stack

Overview of IPFilter Architecture



IPFilter DCA High Availability

IPFilter DCA High Availability

IPFilter DCA rules can be configured without the "flags S" keywords.

Since all "keep limit" connections are also "keep state", having stateful processing, TCP/IP packets associated with a current connection are processed by the "state table."

After a **System Switch-over,** the stand-by system will reconstruct the failed system's connections' state if the stand-by system has the same rule-set as the active system (and "flags S" is not used in the DCA rules).

IPFilter DCA is qualified to work with HP's MC/ServiceGuard High Availability solution.

IPFilter DCA Dynamic Rule Updates

DCA Rules : Dynamic Rule Update

Dynamically change the connection limit

- Useful when we want to change connection limits for suspicious IPs
- All subsequent connections are subject to new connection limits
- System does not require restarting for the new rules to take effect

Example:

The original rule was:

pass in quick proto tcp from 14.13.45.176 to any keep limit 15 cumulative

To change the limit to 1, the following new rule would be added:

pass in quick proto tcp from 14.13.45.176 to any keep limit 1 cumulative

Dynamically change logging frequency

- applies only to Alert log messages
- useful to manage the size of the logging files IPFilter DCA creates

Example:

pass in log limit freq 10 quick proto tcp from any to any keep limit 50

to

pass in log limit freq 100 quick proto tcp from any to any keep limit 50

DCA Rules : Dynamic Rule Update(Contd)

Both the connection limit and log frequency can be changed simultaneously

To Update a rule

configure the same rule but with a different connection limit and/or log frequency

Entire rule-sets can be changed with the

"ipf –s" command

Enter the new rule-set to the inactive rule-list with the command:

ipf –If <file-name>

The inactive rule-set can contain the same rules as the active rule-set plus whichever rules you want to change or update plus any new rules.

After the rule switch, both the individual and cumulative rules will be updated to the new rule-set connection limits and rule ordering (for example, entering a more specific range or subnet rule before a default rule).

Note that only new connections are affected by the new rules

New IPFilter DCA Logging Features

DCA Logging Features

New keywords in rules

'log limit' - This logs each connection which exceeds a configured limit in a 'keep limit' rule.

pass in log limit quick proto tcp from IP1 to Server port 25 keep limit 10

IP1 is allowed to open only 10 concurrent connections at a time. Any subsequent connection will be blocked. If 'log limit' is set it will be logged. So if 15 connections come from IP1, then the first 10 will be let through and the next 5 will be blocked and logged..

DCA Logging features(Cont'd)

More logging keywords

'log limit freq <num>'

This can/should be used to avoid 'flooding the log file. This can happen when an IP creates a huge number of connections which exceed the configured limit.

So if we set the log frequency to 3 and if we have 10 connections which exceeded the configured limit, we log:

- first time we exceeded limit .i.e the 11th connection (Exceeded = 1)
- we log the14th, 17th and 20th connection which exceeded limit (Exceeded = 4,7 and 10) , and when the all the connections are closed.

When all connections are closed, a summary log record for this limit entry is printed

DCA Logging features(Contd)

‘log limit' generates two types of log records

Alert Log records

These tell that an IP sources is trying to exceed its configured limit. The format is below:

*09/04/2003 10:35:40.970000 lan0 @0:3 b 15.13.106.175,51251 -> *
*15.13.137.135, 23 PR tcp len 20 48 -S K-L DEFAULT Configured 1 Current *
1 Exceeded 1 Freq 1 IN

Summary Log records

Summary log records are created when a limit entry ceases to exist after all the connections for that limit entry have been closed. This log record summarizes the connection activity of a particular IP. The format is given below:

*16/04/2003 13:16:50.270000 LIMIT LOG 15.13.106.175, * -> *
15.13.137.135,23 PR tcp Type 8 Cur Lim 1 Exceeded 4 RULE POS 1
First Time 13:15:50.150000

DCA Logging features(Contd)

New options to ipmon and new device files

- **/dev/iplimit**

A new device file is created for DCA logging. The log alert records go to /dev/ipl but the summary records are logged to /dev/iplimit. No change is required in the ipmon invocation,but to log the summary records, use ipmon with the **new '-A' option** as follows:

ipmon -A /dev/iplimit > $LOGDIR/limit_summary.log &

ipmon -r

This posts a summary record to the summary log file and also zeros out the block count for each limit entry.
e.g.

pass in log limit quick proto tcp from IP1 to Server keep limit 10

Generally summary records are created when the limit entries expire, but '*ipmon –r'* can be used to get the 'current summary' and to zero out the block count. You must use ipmon –r to generate a summary log record for cumulative rules.

New IPFilter DCA Commands

New Commands

New options to the ‘ipf’ command:

ipf –m [e | d | q | t]

This command toggles the DCA mode. By default DCA mode is disabled. It can also be set at boot time using the DCA_START flag in /etc/rc.config.d/ipfconf.

ipf –e: enables IPF DCA processing mode

ipf –d: disables IPF DCA processing mode

ipf –q: queries current IPF DCA processing mode

ipf –t: toggles current IPF DCA processing mode

New Commands (Contd)

ipf – [E | D | Q] <interface name>

The above set of commands allows us to enable/disable/query IPFilter processing on a given interface.

ipf –D lan0: disable IPFilter processing for traffic on lan0
ipf –E lan0: enable IPFilter processing on lan0
ipf –Q lan0: verify if IPFilter processing is enabled or disabled for lan0.

If a system has two interfaces [IN and OUT], IPFilter processing is enabled on both interfaces by default. But normally in IPF DCA mode processing, we only want to limit incoming connections on specific interfaces. Disabling outgoing interfaces whose incoming connections have already been examined improves performance.

New Commands (Contd)

New options to the ‘ipfstat’ command:

ipf – L : Shows general limit table stats

Connection Type	Active Limits
Individual	0
Subnet	0
Cumulative	0
Unknown IP	0
Total	0
No Memory	0
Logged Records	0
Log Failures	0
Limits Added	0
Add Failures	0

New Commands (Contd)

ipfstat –vL : shows verbose limit stats

Type	Rule	Src IP	Src Port	Dest IP	Dest Port	Limit	Current
S	@0:5	10.44.1.2	*	10.191.1.3	80	50000	4029 (0)
S	@0:16	10.55.1.2	*	10.191.1.3	80	50000	3722 (0)
S	@0:16	10.55.1.3	*	10.191.1.5	80	50000	4122 (0)
S	@0:7	10.46.1.2	*	10.191.1.4	80	50000	6185 (0)
S	@0:10	10.81.1.2	*	10.191.1.5	80	50000	6135 (0)
S	@0:19	10.72.1.2	*	10.191.1.5	80	50000	3535 (0)
S	@0:2	10.41.1.2	*	10.191.1.5	80	50000	6035 (0)
S	@0:12	10.83.1.2	*	10.191.1.3	80	50000	4222 (0)
S	@0:4	10.43.1.2	*	10.191.1.5	80	50000	4132 (0)
S	@0:14	10.85.1.2	*	10.191.1.5	80	50000	3806 (0)
S	@0:6	10.45.1.2	*	10.191.1.3	80	50000	4032 (0)
S	@0:17	10.56.1.2	*	10.191.1.4	80	50000	6579 (0)
S	@0:8	10.47.1.2	*	10.191.1.3	80	50000	4216 (0)
S	@0:9	10.80.1.2	*	10.191.1.3	80	50000	7027 (0)
S	@0:18	10.71.1.2	*	10.191.1.3	80	50000	3589 (0)
S	@0:1	10.40.1.2	*	10.191.1.3	80	50000	6534 (0)
S	@0:11	10.82.1.2	*	10.191.1.3	80	50000	7097 (0)
S	@0:13	10.84.1.2	*	10.191.1.5	80	50000	3950 (0)
C	@0:4895	10.57.1.2/32	*	any	78 >< 81	10	10 (130)
C	@0:4896	10.57.1.3/32	*	any	78 >< 81	10	10 (130)
C	@0:4897	10.57.1.4/32	*	any	78 >< 81	10	10 (130)
C	@0:4898	10.57.1.5/32	*	any	78 >< 81	10	10 (129)

New Commands (Contd)

ipfstat –r <rule #> : shows per rule limit stats

- # ipfstat -inh
- 146 @0:1 pass in quick proto tcp from any to any keep limit 2

- # ipfstat -vL
- Type Rule Src IP Src Port Dest IP Dest Port Limit Current

- U @0:1 15.16.100.30 * 15.15.118.25 513 2 1 (0)

- # ipfstat -r @:1

- Limit Type Unknown
- Group:Rule Number @0:1
- Configured Limit 2
- Current Limit 1
- Limit exceeded (# times) 0
- TCP RST sent (# times) 0

Using IPFilter DCA Commands With Other Spam Detecting Programs

Using IPFilter DCA With Other Spam Detecting Programs

Identification of spam sources can be programmatically detected by user-space or network-space programs that can then issue IPFilter DCA rules to block or limit the number of connections a spammer can make to the mail servers IPFilter DCA is defending at the firewall filter layer.

HP WORLD 2004

Solutions and Technology Conference & Expo

Co-produced by:

interex

shared knowledge • shared power

encompass

AN HP USER GROUP

RECOMMENDED TRAINING VENUE FOR THE

HP Certified Professional