

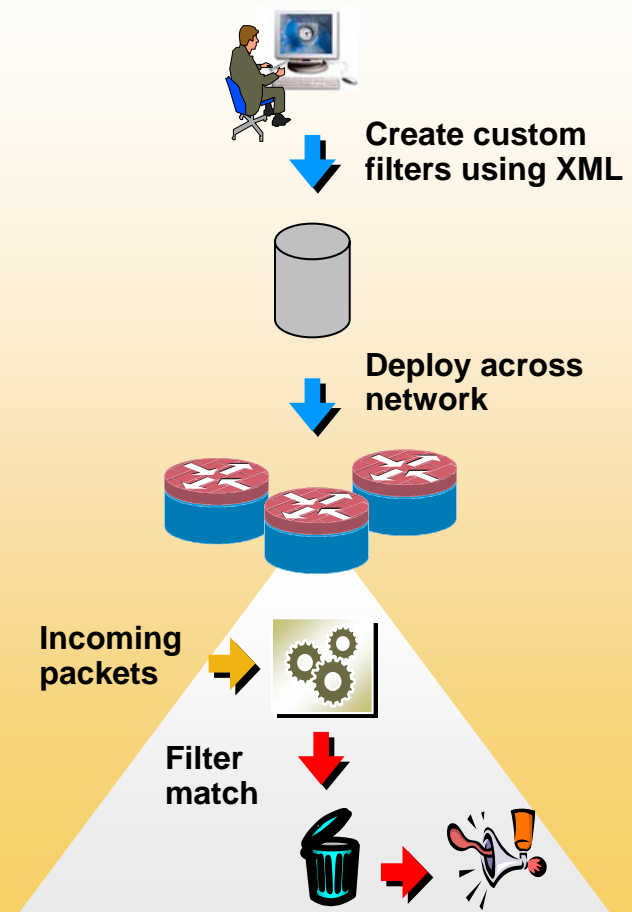
# Flexible Packet Matching



# Mitigating Attacks with Flexible Packet Matching (FPM)

- Next generation “Super ACL” pattern matching capability for more granular and customized packet filters
  - Ability to match on arbitrary bits of a packet at arbitrary depth (offset) in the packet header and payload.
    - Detects malicious patterns deep within the packet
- Provision via CLI or off-box via XML
- Easier and faster to deploy
  - Filters can be unstructured allowing customers to respond quickly
  - Filters install on routers without reload
- Platform support:  
800–3845, 7200, 7301

## Network-based blocking of known threats and attacks



# The Problem

- **Attacks are getting sophisticated—we need the ability to classify on multiple attributes within a packet**

**Example: Slammer's signature was a combination of port 1434, a packet length of 404 bytes, and a byte pattern within payload**

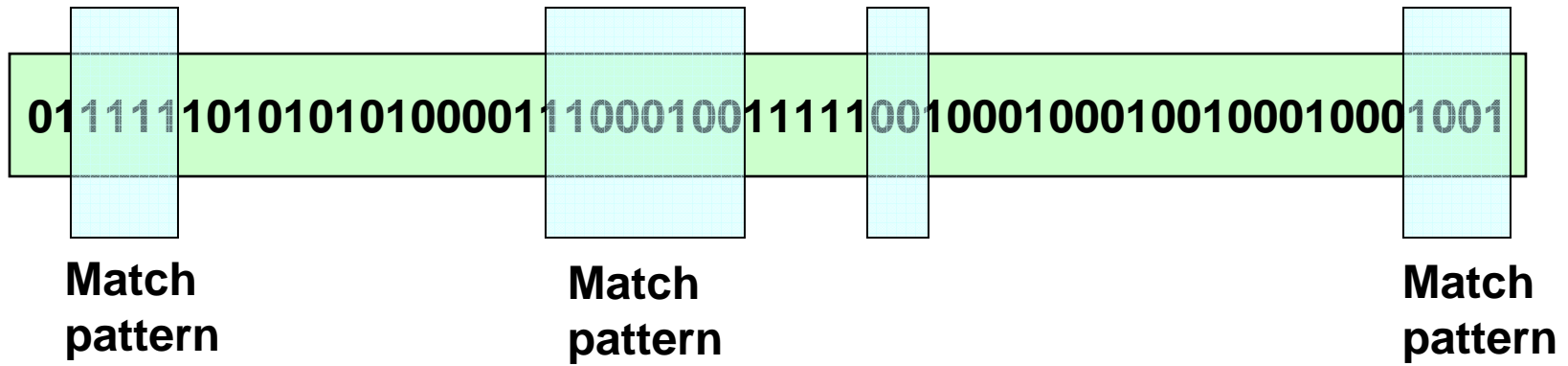
- **We need the ability to rapidly define and deploy classification and filtering mechanisms to mitigate new attacks**

**Current classification and filtering capabilities are traditionally limited to a field or specific to an event, and are developed one at a time**

# The Solution—What Is Needed?

**A versatile, powerful and rapidly deployable classification and filtering mechanism to complement and bridge the gap between existing mechanisms such as ACLs, NBAR, IPS signatures and Cisco guard Anomaly Detection filters.**

# Flexible Packet Matching Functionality



- Ability to match on arbitrary bits of a packet at arbitrary depth (offset) in the packet
- Today supports a depth of 256 bytes
- Layer 2–Layer 7 stateless classification and match capability

# Flexible Packet Matching (FPM)

## Overview

- Enhancement to existing Cisco® ACL functionality
- Users can define customized classification criteria for **stateless** traffic
- Once an attack vector is determined, FPM provides the ability to define the filtering (match) criteria against any portion of the packet
- Classification is based on multiple bit matching patterns and regular expression matches across the packet
- Supported on all access platforms: 800–3845, 7200, 7301

# Flexible Packet Matching Overview (cont.)

- **Describe packet filters using Class-Based Policy Language (CPL), CLI, or XML (TCDF)**
- **Define customized protocol header definition files (PHDFs)**
  - Support for defining PHDFs via XML
  - CLI provides descriptors for various fields within protocol header such as IP, IP options, TCP, UDP
  - PHDFs can be uploaded at run time
- **Standard PHDFs can be downloaded from CCO:**
  - At FCS: IP, UDP, TCP, ICMP, Ethernet
  - Coming: SNMP, HTTP, SMTP, DNS, GRE, IPSec
- **Basic flexible packet matching filter actions:**
  - Drop
  - Log
  - ICMP-Unreachable

# Basic FPM Capability

## Filter Match Operators

- **Relational operators**
  - Eq, NEq, Gt, Lt, GE, LE
  - Support bit mask
- **Logical operators**
  - AND, OR
- **String, match, and regular expressions**
  - Not the same regular expressions as BGP, but a subset;  
uses the same regular expression engine as IPS and NBAR
- **Arithmetic expressions for offsets on variable header length**
- **Values can be entered in decimal or hexadecimal**



# Flexible Packet Matching Use Case

## Slammer Filter

```
rtr(config)# class-map type stack match-all ip_udp
rtr(config-cmap)# description "match UDP over IP packets"
rtr(config-cmap)# match field ip protocol eq 17 next udp
```

Stack class defines IP—  
UDP protocol stack

```
rtr(config)# class-map type access-control match-all slammer
rtr(config-cmap) # description "match on slammer packets"
rtr(config-cmap)# match field udp dest-port eq 1434
rtr(config-cmap)# match field ip length eq 404
rtr(config-cmap)# match start I3-start offset 224 size 4 eq 0x04010101
```

Access-control class  
defines traffic pattern

UDP destination port eq  
1434

4B string pattern  
0x04010101 at 224B offset  
from IP header

```
rtr(config)# policy-map type access-control fpm_udp_policy
rtr(config-pmap)# description "policy for UDP based attacks"
rtr(config-pmap)# class slammer
rtr(config-pmap-c)# drop
```

Drop all packets matching class  
slammer

```
rtr(config)# policy-map type access-control fpm_policy
rtr(config-pmap)# description "drop worms and malicious attacks"
rtr(config-pmap)# class ip_udp
rtr(config-pmap-c)# service-policy fpm_udp_policy
```

Apply input/output service  
policy on per interface  
basis

```
rtr(config)# interface gigabitEthernet 0/1
rtr(config-if)# service-policy type access-control input fpm_policy
```

# How Do I Use FPM?

- **To prepare, download the PHDFs from CCO and load them onto your routers (IP, TCP, UDP, Ethernet)**
- **Minute zero: Determine that anomalous traffic is entering your network (sniffer traces, NetFlow)**
- **Inspect the packet structure of the anomalous traffic**
- **If you can use an ACL to mitigate it, use it!**
- **If you cannot use an ACL because it might block legitimate business traffic, use FPM to classify your traffic and assign an action to the policy: Drop, Log, ICMP-Unreachable**
- **If it is attack traffic, IPS, CICS will have updates coming along soon, but in those first few hours, use ACL/FPM**

# Reference

- **FPM on the Cisco® Website**

<http://www.cisco.com/go/fpm>

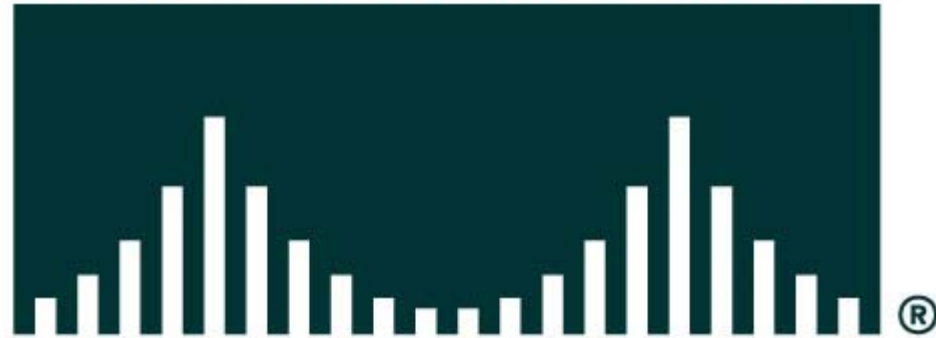
- **Protocol header definition files**

<http://www.cisco.com/cgi-bin/tablebuild.pl/fpm>

- **FPM deployment guide**

[http://www.cisco.com/en/US/products/ps6642/products\\_white\\_paper0900aecd803936f6.shtml](http://www.cisco.com/en/US/products/ps6642/products_white_paper0900aecd803936f6.shtml)

# CISCO SYSTEMS



EMPOWERING THE  
INTERNET GENERATION