Firewall Design and Deployment

Bancha Sae-Lao
MFEC Public Co.,Ltd.
Session Objectives

At the end of the session, you should have:

- Knowledge of common firewall deployment scenarios including firewall virtualization and High Availability
- A better understanding of how logging and packet capture can help profile network protocols and their behaviors
- General knowledge of Application Inspection in the firewall and how it can be used for advanced protocol filtering
- “Best Practice” suggestions for optimizing your firewall deployment
- Note: this session will NOT cover IOS Firewall
Agenda

- Introduction
- Firewall Deployment Modes
- Access Control
- High Availability Features
- Understanding NAT
- Logging and the Firewall
- Application Inspection
- ASA 5580 and the FWSM
- New Features in 8.3
- Q & A
Cisco Firewall Basics
What is Firewall?

- A firewall is a security device which is configured to permit, deny or proxy data connections set by the organization's **security policy**. Firewalls can either be hardware or software based.

- A firewall's basic task is to control traffic between computer networks with different zones of trust.

- Today's firewalls combine multilayer stateful packet inspection and multiprotocol application inspection.

- Modern firewalls have evolved by providing additional services such as VPN, IDS/IPS, and URL filtering.

- Despite these enhancements, the primary role of the firewall is to enforce security policy.

“A Firewall is one tool in your security toolbox”
Cisco Firewall – What is It?

- Adaptive Security Appliance (ASA) – firewall appliance, proprietary OS has one expansion slot for service modules. Ethernet and fiber ports on box. does not run IOS but has a similar look and feel

- FireWall Services Module (FWSM) – line card in Catalyst 6500 that provides firewall services. No physical interfaces, uses VLANs as “virtual interfaces”

- IOS Device running a firewall feature set in software (IOS-FW) – configuration is in IOS - not covered in this session

- Cisco’s firewall has been around over 15 years, PIX the legacy platform
Firewall Design
Simple Internet Firewall Design

Two interfaces: trusted and untrusted

Usually blocks all inbound access from untrusted networks and allows all access outbound from trusted network
Internet Firewall with DMZ

A perimeter network or DMZ (De-Militarized Zone) is a common design element used to add an additional interface to a Firewall.

Firewall DMZ interface allows traffic from trusted and semi-trusted networks as well as traffic from untrusted networks.
Internet Firewall with Multi DMZ

Adding more than one DMZ allows enforcement of different policies.

Complexity of packet flow increases significantly as more DMZ interfaces are added.
**Dual Firewall Design**

- Each firewall is configured for a specific purpose (inbound vs outbound connections)
- Often deployed at trust boundaries
- Logging from each firewall might be useful for forensics
Cisco Firewall Deployment Modes
Firewall Design – Modes of Operation

- **Routed Mode** is the traditional mode of the firewall. Two or more interfaces that separate L3 domains

- **Transparent Mode** is where the firewall acts as a bridge functioning mostly at L2

- **Multi-context** mode involves the use of virtual firewalls, which can be either routed or transparent mode

- **Mixed mode** is the concept of using virtualization to combine routed and transparent mode virtual firewalls

- Mixed mode is only supported on the FWSM today
Firewall - Routed Mode

- Traditional mode of the firewall
- Separates two L3 domains
- Often a NAT boundary
- Policy is applied to flows as the transit the firewall
Firewall – Transparent Mode

- Operates at layer 2, transparent to the network
- Drops into existing networks without re-addressing
- Simplifies internal firewalling & network segmentation

Transparent Firewall Mode

Existing Network
How Does Transparent Mode Work?

- Firewall functions like a bridge ("bump in the wire") at L2, only ARP packets pass without an explicit ACL (does not pass Cisco Discovery Protocol)
- Same subnet exists on inside and outside
- Different VLANs on inside and outside
- Instead of Extended ACLs, use an EtherType ACL to restrict or allow certain traffic types
- NAT is now supported in Transparent Firewall, requires 8.0.2+ on the ASA and 3.2+ on the FWSM
Firewall Transparent Mode Requirements

- A management IP is **required** for both management and for traffic to pass through the transparent firewall
- Set default gateways of hosts to L3 on far side of firewall, NOT the management IP of firewall
- Only two data interfaces are permitted in single context (non-virtualized) mode
- In multi-context mode an interface can not be shared among contexts (virtual firewalls)

Firewall – Transparent (L2) Mode

```
firewall transparent
hostname ciscoasa
!
interface GigabitEthernet0/0
  nameif outside
  security-level 0
!
interface GigabitEthernet0/1
  nameif inside
  security-level 100
```
Why Deploy Transparent Mode?

- Routers can establish routing protocols adjacencies through the firewall
- Protocols such as HSRP, VRRP, GLBP can cross the firewall
- Multicast streams can traverse the firewall
- Non-IP traffic can be allowed (IPX, MPLS, BPDUs)
- Deploy where IP address schemes can not be modified
- NO dynamic routing protocol support or VPN support
- NO QoS or DHCP Relay support
Firewall Design - Virtualization

Virtualization provides a way to create multiple firewalls in the same physical chassis

Maximum number of virtual firewalls is 50 for the ASA and 250 for the FWSM (not supported on ASA 5505)

Virtualization is a licensed feature

Commonly used to apply unique security policies in one physical chassis
Virtual Firewalls

- VLANs can be shared in routed mode, if needed
- Each context has its own policies (NAT, access lists, etc.)
Virtual Firewall on ASA and FWSM

- Context = a virtual firewall
- All virtualized firewalls must define a System context and an Admin context at a minimum

- There is no policy inheritance between contexts
- The system space uses the admin context for network connectivity; system space creates other contexts
Virtualization and Resource Management

- By default, all virtual firewalls (contexts) have access to unlimited physical resources in the ASA.

- To avoid exhausting system resources, the ASA can be configured to manage resources as a percentage or an absolute number.

- This is common in multi-tenant environments where one physical firewall is virtualized to serve multiple customers.

```plaintext
class gold
limit-resource mac-addresses 10000
limit-resource conns 15%
limit-resource rate conns 1000
limit-resource rate inspects 500
limit-resource hosts 9000
limit-resource asdm 5
limit-resource ssh 5
limit-resource rate syslogs 5000
limit-resource telnet 5
limit-resource xlates 36000
```
Unsupported Features with Virtualization

- Dynamic routing protocols (EIGRP, OSPF, RIP) are not supported
- Multicast routing is not supported (multicast bridging is supported)
- MAC addresses for virtual interfaces are automatically set to physical interface MAC
- Admin context can be used, but grants root privileges to other contexts, use with caution
- VPN services are not supported
Firewall Design -- Mixed Mode

- Mixed Mode is the concept of using virtual firewalls, some in routed mode and some in transparent (L2) mode.

- This is only supported on the FWSM today with 3.1 code or newer.

- Up to 8 pairs of interfaces are supported per context.
Firewall Access Control
Firewall Security Levels

β A security level is a number between 0 and 100 that determines how firewall rules are processed for the data plane.

β Security levels are tied to an interface: the inside or private side interface is always 100 (most trusted) and the outside or public interface is always 0 (least trusted).

β DMZ interfaces, if used, may be assigned numbers between 1 and 99.

β Traffic on the ASA is allowed by default from a higher security level interface to a lower security level interface.

β An ACL must explicitly permit traffic from a lower security level interface to a higher (e.g. outside to in).
Access Control Lists

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard</td>
<td>Used for routing protocols, not firewall rules</td>
</tr>
<tr>
<td>Extended</td>
<td>Source/destination port and protocol</td>
</tr>
<tr>
<td>Ethertype</td>
<td>Used with transparent mode</td>
</tr>
<tr>
<td>Webtype</td>
<td>Used for clientless SSL VPN</td>
</tr>
</tbody>
</table>

- Like Cisco IOS, ACLs are processed from top down, sequentially with an implicit deny all at the bottom.
- A criteria match will cause the ACL to be exited.
- ACLs are made up of Access Control Entries (ACE).
- Remarks can be added per ACE or ACL.
- ACLs can be enabled/disabled based on time ranges.
Understanding ACL Logging

Logging is optional, but is a best practice method for understanding what is happening to traffic as it passes through the firewall.

Each ACE can be logged by adding the keyword ‘log’ to the end of the ACE.

```bash
asa(config)# access-list MYACL deny ip any any log
```

This will create a system message 106100 to be generated every time a flow is created that meets the ACE criteria.

No other logs will be created for the same flow until the logging interval is exceeded, which by default is five minutes.

This prevents firewall resources from being consumed for repetitive packets from the same flow.
ACL New Features (8.3)

 Older versions of PIX code 6.x and earlier had issues with exceptionally large ACL entries (100k+)

 With the release of 7.0.x and 8.0.x code ACL processing was vastly improved to create a consistent performance response regardless of ACL size

 In 8.3 ACLs timestamps were modified to include the last time an ACL was hit, instead of just the more generic “hit-count”

 Fourth hash is the timestamp in UNIX Epoch format

```
asa(config)# sh access-list test brief
 access-list test; 3 elements; name hash: 0xcb4257a3
c10ca21 44ae5901 00000001 4a68aa7e
```
ACL New Features (8.3) - Continued

- New 8.3 command: `object-group-search access-control`
- Only use when available memory is depleted due to large object-group structure, disabled by default
- Indenting ACEs expanding from object-group

```console
asa# show access-list
access-list cached ACL log flows: total 0, denied 0 (deny-flow-max 4096)
  alert-interval 300
  access-list 101; 1 elements
  access-list 101 line 1 extended permit tcp any any (hitcnt=0)
  access-list 102; 2 elements
  access-list 102 line 1 extended permit icmp any any (hitcnt=0)
  access-list 102 line 2 extended permit ip any any (hitcnt=0)
  access-list 103; 3 elements
  access-list 103 line 1 extended permit object-group internet any any
    access-list 103 line 1 extended permit tcp any any (hitcnt=0)
    access-list 103 line 1 extended permit ip any any (hitcnt=0)
    access-list 103 line 1 extended permit icmp any any (hitcnt=0)
```
Object Groups Simplify Configurations

Object groups allow grouping of similar items for easing configuration and operational maintenance of the ASA firewall.

- Can be grouped by protocol, network or service.
- Can be nested for more granular configuration options.

```
(config)# object-group network ADMINS
(config-protocol)# description Administrator Addresses
(config-protocol)# network-object host 10.1.1.4
(config-protocol)# network-object host 10.1.1.78
(config-protocol)# network-object host 10.1.1.34

(config)# object-group service RAD-GROUP udp
(config-service)# description RADIUS Group
(config-service)# port-object eq radius
(config-service)# port-object eq radius-acct
```
ASA 8.3 Global Policies

- Until recently, ACLs were applied to firewall interfaces for inbound and outbound traffic.
- Release 8.3 adds the ability to configure Global Access Policies which are not tied to a specific interface.
- GA policies only affect traffic going through the firewall, not used with control-plane traffic.
- Interface ACLs take priority over Global Access Policies.
- Access control policies now reference the real (pre-NAT) address.
Understanding NAT

Network Address Translation
Understanding Network Address Translation (NAT) and the Firewall

NAT is simply a means of address translation; commonly this is a privately addressed source into a globally routed address pool (typically 1x1 mapping).

Port Address Translation (PAT) is the idea of mapping multiple source addresses into one “outside” address, commonly the interface address of the firewall (1 to many mapping).

NAT is available in both routed mode and transparent mode.

Most importantly, NAT is not required for the firewall to function and therefore is completely optional.
NAT Control

NAT control is the concept that a packet from a high security interface (e.g. “inside”) must match a NAT policy when traversing a lower level security interface (e.g. “outside”)

If the packet does not match a NAT policy, then it is dropped

NAT control is disabled by default**

** In certain cases it may be enabled after an upgrade
Dynamic NAT

- **Dynamic NAT** is the most common application of NAT on the Cisco firewall.
- It allocates addresses from a specified pool for hosts as they establish connections to meet the NAT policy.
- There is no relationship between a host and the its translated address, hence it’s “dynamic” (vs static).
Port Address Translation (PAT)

- **PAT** is best used in small networks where a global address pool isn’t available.
- Commonly the firewall interface address is used.
- Best way to conserve addresses as translates all addresses into one address and uses port numbers for tracking.

```
10.1.1.x /24

Firewall outside:172.16.1.100 /24

172.16.1.100:1025
172.16.1.100:1026
172.16.1.100:1027
...
172.16.1.100:2800
```
Static NAT

- **Static NAT** is the most common form of NAT when reachability is critical to a host.
- Addresses have a 1x1 relationship.
- Assuming the access policy permits it, the host is reachable both from the inside and outside networks.

**Diagram:**
- A web server with an IP address of 10.1.1.1.
- A NAT pool with addresses from 10.1.1.1 to 172.16.1.1.
- The NAT device translates the IP addresses as traffic moves between the web server and the outside network.
Static PAT

Static PAT is the option where only one global address is available and multiple services require reachability from the outside (e.g. HTTP, SSH, SMTP, etc.)

Static NAT only allows a 1x1 mapping, but static PAT allows multiple listeners on a single port

An excellent solution for hosting services on a small network
Three Options for Bypassing NAT

- Identity NAT (nat 0) – limits NAT on all interfaces, very little granularity, will not allow outside to inside connections even if permitted by ACL
- Static Identity NAT – based on interface, a host can be translated on one interface and not translated on another. Works with Policy NAT
- NAT exemption (nat 0 with Access Control List) – more granular than Identity NAT as it allows bidirectional communication between inside and outside hosts. The ACL allows for very specific NAT policies to be created
- NAT exemption is the most common method for bypassing NAT today
Policy NAT

In some cases it may be necessary to have a NAT policy that translates based on source AND destination.

While dynamic NAT only considers the source address, policy NAT looks at both source and destination including port numbers.

Very useful when an application (e.g. FTP, VoIP) has secondary channels that need a specific policy.

The source, destination and all relevant ports are assigned via an ACL.

Policy NAT does not support time-based ACLs.
Configuring NAT

NAT configuration requires at least two parts: a nat statement and a matching global statement

```
asa(config)# nat (inside) 1 10.1.2.0 255.255.255.0
asa(config)# global (outside) 1 172.16.1.3-172.16.1.10
```

Multiple nat statements can reference the same global

```
asa(config)# nat (inside) 1 10.1.2.0 255.255.255.0
asa(config)# nat (inside) 1 192.168.1.0 255.255.255.0
asa(config)# nat (dmz) 1 10.1.1.0 255.255.255.0
asa(config)# global (outside) 1 209.165.201.3-209.165.201.10
```

Multiple NAT ids can be used for NAT policy granular matching
Pre 8.3 NAT Best Practices

NAT and DNS resolution for internal hosts vs. external hosts

NAT Order of Operation matching—the first three use first match criteria

1. NAT Exemption
2. Static NAT and Static PAT (including Policy NAT)
3. Policy Dynamic NAT
4. Regular Dynamic NAT – order of statements doesn’t matter as is based upon best match criteria
NAT Redesign in ASA 8.3

Starting with the 8.3 release, NAT has been completely redesigned to simplify configuration and troubleshooting.

Follows “original packet” vs. “translated packet” model.

New features:

1. Unified NAT Table to view all NAT policies.
2. Object-based NAT: object can be created for hosts, networks or address ranges and NAT can be configured within the object.
3. Two NAT Options: Object-based (Auto) and Manual NAT.
4. Interface independent NAT.
NAT Order of Operation

NAT rules are applied via top down order with first match

Rules are processed in the following order:

1. Manual NAT rules
2. Object-based NAT rules
3. Manual (twice) NAT rules (translating both source and dest)

Use packet tracer in ASDM for validating NAT policy

For specific examples see the Configuration Guide here:
### ASA 8.3 Unified NAT Table

#### Network Object NAT (Rules 2-3)

<table>
<thead>
<tr>
<th>Match Criteria: Original Packet</th>
<th>Action: Translated Packet</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source Interface:</td>
<td>Source NAT Type:</td>
</tr>
<tr>
<td>Dest Interface:</td>
<td>Static</td>
</tr>
<tr>
<td>Source Address:</td>
<td>Source Address:</td>
</tr>
<tr>
<td>any</td>
<td>-- Original --</td>
</tr>
<tr>
<td>Destination:</td>
<td>Destination Address:</td>
</tr>
<tr>
<td>any</td>
<td>any</td>
</tr>
<tr>
<td>Service:</td>
<td>Service:</td>
</tr>
<tr>
<td>any</td>
<td>any</td>
</tr>
<tr>
<td>Destination Interface:</td>
<td></td>
</tr>
<tr>
<td>any</td>
<td></td>
</tr>
<tr>
<td>Destination Address:</td>
<td></td>
</tr>
<tr>
<td>any</td>
<td></td>
</tr>
<tr>
<td>Service:</td>
<td></td>
</tr>
<tr>
<td>any</td>
<td></td>
</tr>
</tbody>
</table>

**Options**
- Enable rule
- Translate DNS replies that match this rule

**Direction:** Both

**Description:**
Firewall High Availability
HA Feature – Interface Redundancy

- Up to 8 redundant interface pairs are allowed.
- Compatible with all firewall modes (routed/transparent and single/multiple) and all HA deployments (A/A and A/S)
- When the active physical interface fails, traffic fails to the standby physical interface and routing adjacencies, connection, and auth state won’t need to be relearned.
- NOT supported on ASA 5505 or FWSM

```plaintext
interface Redundant1
  member-interface GigabitEthernet0/2
  member-interface GigabitEthernet0/1
  no nameif
  no security-level
  no ip address
!
interface Redundant1.4
  vlan 4
  nameif inside
  security-level 100
  ip address 172.16.10.1 255.255.255.0
!
interface Redundant1.10
  vlan 10
  nameif outside
  security-level 0
  ip address 172.16.50.10 255.255.255.0
```
HA with Interface Redundancy

Before...

After with redundant interface
Caveats of Interface-level Redundancy

- Members can only operate in Active/Standby mode. No load sharing or link aggregation is supported at this time.
- Feature is available on ASA 5510 and above. On ASA 5505, similar capability is available through the built-in switch.
- No support for FWSM (no physical interfaces)
- Subinterfaces (dot1q) need to be built on top of the logical redundant interface, not physical member interfaces.
HA Feature – Route Tracking

Method for tracking the availability of static routes with the ability to install a backup route should the primary route fail.

Commonly used for static default routes, often in a dual ISP environment.

Uses ICMP echo replies to monitor the availability of a target host, usually the next hop gateway.

Can only be used in single routed mode.

```
asa(config)# sla monitor 123
asa(config-sla-monitor)# type echo protocol ip icmp Echo 10.1.1.1 interface outside
asa(config-sla-monitor-echo)# frequency 3
asa(config)# sla monitor 123 life forever start-time now
asa(config)# track 1 rtr 123 reachability
asa(config)# route outside 0.0.0.0 0.0.0.0 10.1.1.1 track 1
```
Firewall HA - Active/Standby

- Supported on all models including ASA 5505**
- Requires an additional “Plus” license (5505 and 5510 only)
- ASA only supports LAN Based failover (no serial cable).
- Both platforms must be identical in software, licensing, memory and interfaces (including SSM modules)
- Same mode (i.e. routed or transparent)
- Not recommended to share the state and failover link, use a dedicated link for each
- Preferably these cables will be connected into the same switch with no hosts

**ASA 5505 does not support stateful failover, only stateless
How Failover Works

- Failover link passes Hellos between active and standby units every 15 seconds (tunable from 3-15 seconds)
- After three missed hellos, primary unit sends hellos over all interfaces to check health of its peer
- Whether a failover occurs depends on the responses received
- Interfaces can be prioritized by specifically monitoring them for responses
- If the failed interface threshold is reached then a failover occurs

For more details refer to the Configuration Guide:  
What does Stateful Failover Mean?

<table>
<thead>
<tr>
<th>State Info Passed to Standby</th>
<th>Things NOT Passed to Standby</th>
</tr>
</thead>
<tbody>
<tr>
<td>NAT Translation Table</td>
<td>User authentication table</td>
</tr>
<tr>
<td>TCP connection states</td>
<td>Routing table information **</td>
</tr>
<tr>
<td>UDP connection states</td>
<td>State information for SSMs (IPS etc.)</td>
</tr>
<tr>
<td>ARP Table</td>
<td>DHCP Server Leases</td>
</tr>
<tr>
<td>L2 Bridge Table (Transparent Mode)</td>
<td>Stateful failover for phone proxy</td>
</tr>
<tr>
<td>HTTP State *</td>
<td></td>
</tr>
<tr>
<td>ISAKMP and IPSEC SA Table</td>
<td></td>
</tr>
</tbody>
</table>

* HTTP State is not passed by default for performance reasons; enable via ‘http replication state’

** Dynamic routing protocols must converge on their own accord during and immediately following failover
Failover Best Practices

- Back in ye olden times, the PIX firewall used a serial cable (RS-232) for failover
- The ASA uses dedicated ports for failover and failover ports will NOT pass traffic
- Recommended to use separate connections for failover and state if stateful failover is required
- Connection can either be via X-over cable or cabled into a switch in a dedicated VLAN (ASA supports Auto MDI/MDIX)
- Long distance LAN failover is supported if latency is less than 10ms and no more than 250ms
- IPv6 HA supported since 8.2.2
Firewall HA – Transparent Mode

- Transparent Firewall can run in A/S or A/A mode
- Since the firewall acts like a switch, Spanning Tree is recommended to control BPDU forwarding
- Care should be taken to ensure that STP root is as intended
- Ensure that topology is free of all loops!
A/S Failover in Transparent Mode

- Mandatory that no loops in network topology
- Switches connected to HA firewalls should be configured for STP, understand the implications
- Use RPVST (802.1w) and Port Fast feature on switches where possible
- No BPDU Guard or Loop Guard on ports connecting to firewalls
- Use caution if deploying transparent firewalls in Active/Active mode because BPDUs are forwarded by default

TAC Podcast on Transparent Firewall:
Firewall HA: Active/Active Failover

- Supported on all platforms except the 5505
- Requires an additional “Plus” license (5510 only)
- Requires virtualization which requires additional licensing
- Virtualization does not support VPN, multicast or routing protocols
- No load-balancing or load-sharing support today

VFW-1 Active
VFW-4 Standby
VFW-2 Standby
VFW-3 Active

Red = Virtual Pair 1
Blue = Virtual Pair 2
Firewall HA: A/A Failover with Asymmetric Routing support

- ASR mode adds support for asymmetric traffic flows through an A/A system
- A/A ASR is enabled by adding multiple A/A units to the same ASR Group
- When traffic is received on VFW-3 it has no entry in state table and therefore checks state information of other interfaces in ASR Group
- If no match, packet is dropped
- If matched, then rewrite L2 header and forward to other active firewall (VFW-1)
- VFWs in same ASR group must be L2 adjacent
Limitations of Active/Active Failover

- Need to guarantee a low-latency state sharing between two A/A firewalls to avoid a race condition if a return connection arrives prior to state information being received.

- Shared interface setup requires NAT.

- HTTP state information is NOT shared by default and must be explicitly configured.

- Layer 2 adjacency is required between the physical ASAs in an ASR-group.

- Multi-context ASA does not support VPN, multicast routing or dynamic routing protocols.
Firewall Logging
**Best Practice #1:**
**Know your network and the traffic in it**

- Logging today is more critical than in the past (SOX, HIPAA, PCI and other regulatory compliance)

- Logging is a good way to baseline and monitor traffic on your network in order to enforce security policy

- Log analysis makes it possible to audit security policy changes (i.e. changing perimeter ACL configs, denied traffic)

- Good starting point: SANS “Top 5 Log Reports”

- But......log files must be analyzed to be helpful!

Source:
www.sans.org/resources/top5_logreports.pdf
Overview: Logging with Syslog

- Defined in RFC 3164, syslog is a protocol that allows a host to send event information to a syslog server.

- Messages are commonly sent via UDP port 514 and are <1024 bytes.

- By default, syslog provides no concept of authentication or encryption.

- Events can be sent to a syslog server on any port between 1025 – 65535) via either UDP (default 514) or TCP (default 1470).

- Only use TCP based syslog for special circumstances due to performance limitations (connection-oriented).
ASA Event Format

A typical PIX/ASA 7.x syslog message:

```
Apr 12 2009 14:32:55 BIGFW : %ASA-5-111008: User 'enable_15' executed the 'ping 10.1.10.1'
```

- **Timestamp** – (disabled by default)
- **Device-ID** – (none by default), (interface name, ip address, hostname, context name or custom string up to 16 characters)
- **Facility code** (always %PIX or %ASA), **Severity level** (5) and **Event Message number** (111008)

**Message Text**

EMBLEM format is used by IOS routers and switches
 Syslog Severity Codes and Facilities

Severity Codes (0 – 7)

- 0 - Emergency
- 1 - Alert
- 2 - Critical
- 3 - Error
- 4 - Warning
- 5 - Notice
- 6 - Informational
- 7 - Debug

Severity codes are combined with facility codes to generate a message priority.

Syslog server uses these facility codes to organize event messages as they arrive.

Eight logging facilities are available (local0 – local8).

Local4 (20) is the default for all PIX/ASA and FWSM events.
# Logging Levels and Events

<table>
<thead>
<tr>
<th>Log Level</th>
<th>Alert</th>
<th>Event Messages</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Emergencies</td>
<td>Not used, only for RFC compliance</td>
</tr>
<tr>
<td>1</td>
<td>Alerts</td>
<td>Mostly failover-related events</td>
</tr>
<tr>
<td>2</td>
<td>Critical</td>
<td>Denied packets/connections</td>
</tr>
<tr>
<td>3</td>
<td>Errors</td>
<td>AAA failures, CPU/memory issues, routing issues, some VPN issues</td>
</tr>
<tr>
<td>4</td>
<td>Warnings</td>
<td>Denied conns due to ACL, IDS events, fragmentation, OSPF errors</td>
</tr>
<tr>
<td>5</td>
<td>Notifications</td>
<td>User and Session activity and firewall configuration changes</td>
</tr>
<tr>
<td>6</td>
<td>Informational</td>
<td>ACL logging, AAA events, DHCP activity, TCP/UDP connection and teardown</td>
</tr>
<tr>
<td>7</td>
<td>Debugging</td>
<td>Debug events, TCP/UDP request handling, IPSEC and SSL VPN connection information</td>
</tr>
</tbody>
</table>
ASA Syslog Destinations

<table>
<thead>
<tr>
<th>Console</th>
<th>Buffered Memory</th>
</tr>
</thead>
<tbody>
<tr>
<td>VTY (Telnet, SSH)</td>
<td>ASA Device Manager (ASDM)</td>
</tr>
<tr>
<td>Flash Memory</td>
<td>External Syslog server</td>
</tr>
<tr>
<td>Email</td>
<td>External FTP server</td>
</tr>
</tbody>
</table>

In addition multiple external syslog servers are supported.
Logging to an External Syslog Server

Enable logging (disabled by default)

`asa(config)# logging enable`

Send logs to an external syslog server

`asa(config)# logging host interface_name ip_address [tcp[/port] | udp[/port]] [format emblem]`

Set logging level

`hostname(config)# logging trap {severity_level | message_list}`

All system messages at that level and lower will be sent; for example, if logging is set to level 4 (Warnings), then firewall will send levels 1-4 by default
Best Practice #2: Log to Buffered Memory for shorter duration tuning/troubleshooting

- Log data can be sent to a reserved portion of memory much more efficiently than the console or interactive shells

- Buffered logging uses a 4096 byte circular memory buffer stored in volatile RAM

  ```console
  asa(config)# logging buffered {severity_level | message_list}
  ```

- It’s possible to adjust the depth of the buffer

  ```console
  asa(config)# logging buffer-size {4096 – 1048576}
  ```

- Use EXEC command `show logging` to view contents

- Timestamps are disabled by default, enable timestamps:

  ```console
  asa(config)# logging timestamp
  ```
Saving Internal Buffered Logs

An interesting feature of the PIX/ASA is that it can export the internal log buffer to an FTP server if it wraps.

```console
asa(config)# logging ftp-bufferwrap
asa(config)# logging ftp-server server path username password
```

- Manually save it to flash memory
  ```console
  asa(config)# logging savelog [savefile]
  ```

- Or save it to flash memory when buffer wraps
  ```console
  asa(config)# logging flash bufferwrap
  ```

**NOTE:** Logging the buffer to flash can create major performance issues due to time it takes to write to flash. Use only for short duration and with care.
Buffered Logging Configuration in ASDM

- Enable logging
- Send debug messages as syslog
- Send syslog in EMBLEM format

Logging to Internal Buffer
- Buffer Size: 4096 bytes

Save Buffer To:
- FTP Server
- Flash

ASDM Logging
- Queue Size: 4096
Logging to ASDM Real Time Viewer

 diversas

Messages are sent encrypted via SSL to ASDM

ASDM event data is stored in a circular memory buffer that is reserved specifically for ASDM logs

```
asa(config)# logging asdm {severity_level | message_list}
```

Similar to buffered logging, oldest messages are deleted first. Default message buffer is 100 messages

Buffer size can be increased up to 512 messages:

```
asa(config)# logging asdm-buffer-size num_of_msgs
```
ASDM Syslog Viewer “Show Rule”

Choosing this in the ASDM Log Viewer will then highlight the appropriate rule on the ASDM Security Policy page.

This only works for syslog ids 106100 and 106023, any other syslog will result in an error message.
Best Practice #3: Egress Filtering + Log

- Default behavior of the ASA is to allow all traffic from trusted zones (i.e. Inside) to any less trusted zone (lower security level)
- A corporate security policy should explicitly state which applications and protocols are allowed outbound
- Egress filtering can provide a means to detect malicious traffic from compromised internal hosts
- SANS “Top 5 Log Reports” highlights the requirement for egress filtering and logging
- Two options for egress filtering: active and passive
Active and Passive Egress Filtering

Actively denying all unspecified traffic:

```
asa(config)# access-list ACTIVE_FILTER extended [permit some traffic here] <log> interval 300
...
asa(config)# access-list ACTIVE_FILTER extended deny ip any any log interval 300
asa(config)# access-group ACTIVE_FILTER in interface inside
```

Passively monitoring all traffic:

```
asa(config)# access-list PASSIVE_FILTER extended permit ip any any log interval 300
asa(config)# access-group PASSIVE_FILTER in interface inside
```
Best Practice #4: Log to Syslog server(s)

- Best option for long-term archiving as well as network trending and analysis.
- CS-MARS can capture RFC compliant syslog as well as other proprietary messaging formats from Cisco and non-Cisco devices.
- Kiwi Syslog Daemon – runs on Windows.
- Syslogd on Linux and UNIX hosts.
- Sawmill for Windows and Linux.
- Splunk runs on pretty much everything, including Mac OS X.
- And many, many more.
Logging to CS-MARS

100 ICMP messages in 10 seconds might indicate scanning activity and is worth deeper investigation.
Log Analysis

The HACME Office

Case Study
Case Study: The HACME Remote Office

- Remotely monitored small remote office
- Default outbound policy; no egress filtering
- Complaints of suspicious activity sourced from 192.168.1.201
- Basic log analysis will show us general traffic/protocol patterns

```
asa(config)# logging enable
asa(config)# ! Enables logging
asa(config)# logging buffer-size 65536
asa(config)# ! Expand buffer to 64K
asa(config)# logging buffer 6
asa(config)# ! Send logs to buffered memory at level 6 (Informational)
```
Log Analysis is Critical

- Logging is a security analyst’s best friend
- Understanding how to read and interpret log data is critical for traffic analysis and forensics
- You can’t analyze what you don’t capture
- Automated tools and scripts make trending and pattern matching much easier
- Does your SECOPS staff understand how to interpret log data?
- Let’s begin the interactive portion of the session….
### A Typical Firewall Log File

<table>
<thead>
<tr>
<th>Event</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>%ASA-6-106100: access-list inside_access_in permitted udp inside/192.168.1.200(1047) -&gt; outside/24.25.5.150(53) hit-cnt 1 first hit [0xa925365e, 0x0]</td>
<td>Built outbound UDP connection 3601 for outside:24.25.5.150/53 (24.25.5.150/53) to inside:192.168.1.200/1047 (xx.77.67.190/1619)</td>
</tr>
<tr>
<td>%ASA-6-302015: Built outbound UDP connection 3601 for outside:24.25.5.150/53 to inside:192.168.1.200/1047 duration 0:00:00 bytes 147</td>
<td>Teardown UDP connection 3601 for outside:24.25.5.150/53 to inside:192.168.1.200/1047 duration 0:00:00 bytes 147</td>
</tr>
<tr>
<td>%ASA-6-106100: access-list inside_access_in permitted udp inside/192.168.1.200(1563) -&gt; outside/10.16.151.94(1029) hit-cnt 1 first hit [0xa925365e, 0x0]</td>
<td>Built outbound UDP connection 3602 for outside:10.16.151.94/1029 (10.16.151.94/1029) to inside:192.168.1.200/1563 (xx.77.67.190/1644)</td>
</tr>
<tr>
<td>%ASA-6-302016: Teardown UDP connection 3601 for outside:24.25.5.150/53 to inside:192.168.1.200/1047 duration 0:00:00 bytes 147</td>
<td>Teardown UDP connection 3601 for outside:24.25.5.150/53 to inside:192.168.1.200/1047 duration 0:00:00 bytes 147</td>
</tr>
<tr>
<td>%ASA-6-106100: access-list inside_access_in permitted udp inside/192.168.1.200(1563) -&gt; outside/10.16.151.94(1029) hit-cnt 1 first hit [0xa925365e, 0x0]</td>
<td>Built outbound UDP connection 3602 for outside:10.16.151.94/1029 (10.16.151.94/1029) to inside:192.168.1.200/1563 (xx.77.67.190/1644)</td>
</tr>
<tr>
<td>%ASA-6-302016: Teardown UDP connection 3544 for outside:171.68.10.143/1029 to inside:192.168.1.200/1530 duration 0:02:02 bytes 0</td>
<td>Teardown UDP translation from inside:192.168.1.200/1563 to outside:xx.77.67.190/1638 duration 0:02:13</td>
</tr>
</tbody>
</table>
Best Practice #5: Use Logging Filters

```bash
asa(config)# sh logging | ?

begin  Begin with the line that matches
exclude Exclude lines that match
grep    Include/exclude lines that match
include Include lines that match

asa(config)# sh logging | grep ?

-v     Exclude lines that match
LINE   Regular Expression

To see events generated by host 192.168.1.201

show logging | grep 192.168.1.201

NOTE: Verify there are no trailing spaces after host ip address
```
asa(config)#  sh logging | grep 192.168.1.201 !truncated

%ASA-6-305011: Built dynamic UDP translation from
  inside:192.168.1.201/1025 to outside:xx.77.67.190/1451

%ASA-6-302015: Built outbound UDP connection 2229 for
  outside:24.25.5.150/53 (24.25.5.150/53) to
  inside:192.168.1.201/1025 (xx.77.67.190/1451)

%ASA-6-302016: Teardown UDP connection 2229 for
  outside:24.25.5.150/53 to inside:192.168.1.201/1025 duration
  0:00:00 bytes 347

%ASA-6-305011: Built dynamic ICMP translation from
  inside:192.168.1.201/512 to outside:xx.77.67.190/10

%ASA-6-302020: Built ICMP connection for faddr 66.151.158.183/0
gaddr xx.77.67.190/10 laddr 192.168.1.201/512

%ASA-6-302020: Built ICMP connection for faddr 66.151.158.183/0
gaddr xx.77.67.190/10 laddr 192.168.1.201/512

%ASA-6-302021: Teardown ICMP connection for faddr 66.151.158.183/0
gaddr xx.77.67.190/10 laddr 192.168.1.201/512

%ASA-6-302021: Teardown ICMP connection for faddr 66.151.158.183/0
gaddr xx.77.67.190/10 laddr 192.168.1.201/512
asa(config)# sh logging | grep 192.168.1.201 !truncated

%ASA-6-305011: Built dynamic UDP translation from inside:192.168.1.201/1025 to outside:xx.77.67.190/1476

%ASA-6-302015: Built outbound UDP connection 2339 for outside:24.25.5.150/53 (24.25.5.150/53) to inside:192.168.1.201/1025 (xx.77.67.190/1476)

%ASA-6-302016: Teardown UDP connection 2339 for outside:24.25.5.150/53 to inside:192.168.1.201/1025 duration 0:00:00 bytes 256

%ASA-6-305011: Built dynamic TCP translation from inside:192.168.1.201/1434 to outside:xx.77.67.190/1961

%ASA-6-302013: Built outbound TCP connection 2340 for outside:64.102.255.95/21 (64.102.255.95/21) to inside:192.168.1.201/1434 (xx.77.67.190/1961)

%ASA-6-302014: Teardown TCP connection 2340 for outside:64.102.255.95/21 to inside:192.168.1.201/1434 duration 0:00:20 bytes 1793 TCP FINs
asa(config)#  sh logging | grep 192.168.1.201 

%ASA-6-302013: Built outbound TCP connection 1373 for 
outside:198.133.219.25/80 (198.133.219.25/80) to 
inside:192.168.1.201/1353 (xx.77.67.190/1630)

%ASA-6-302013: Built outbound TCP connection 1374 for 
outside:198.133.219.25/80 (198.133.219.25/80) to 
inside:192.168.1.201/1354 (xx.77.67.190/1631)

%ASA-5-304001: 192.168.1.201 Accessed URL 

%ASA-6-302014: Teardown TCP connection 1374 for 
outside:198.133.219.25/80 to inside:192.168.1.201/1354 
duration 0:00:00 bytes 1023 TCP FINs
HACME Office Sample Log #4

%ASA-6-302015: Built outbound UDP connection 6203 for
outside:24.25.5.150/53 (24.25.5.150/53) to
inside:192.168.1.201/1025 (xx.77.67.190/1990)

%ASA-6-305011: Built dynamic TCP translation from
inside:192.168.1.201/2224 to outside:xx.77.67.190/3241

%ASA-6-302013: Built outbound TCP connection 6204 for
outside:83.140.172.211/6667 (83.140.172.211/6667) to
inside:192.168.1.201/2224 (xx.77.67.190/3241)

%ASA-6-305011: Built dynamic UDP translation from
inside:192.168.1.201/2225 to outside:xx.77.67.190/2004

%ASA-6-302015: Built outbound UDP connection 6206 for
outside:10.16.151.86/1029 (10.16.151.86/1029) to
inside:192.168.1.201/2225 (xx.77.67.190/2004)

%ASA-6-302013: Built outbound TCP connection 6207 for
outside:83.140.172.211/6667 (83.140.172.211/6667) to
inside:192.168.1.201/2224 (xx.77.67.190/3241)
What runs on TCP 6667?

### Port Information

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Service</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>tcp</td>
<td>DarkFTP</td>
<td>[trojan] Dark FTP</td>
</tr>
<tr>
<td>tcp</td>
<td>Trinity</td>
<td>[trojan] Trinity</td>
</tr>
<tr>
<td>tcp</td>
<td>TheThing</td>
<td>[trojan] The Thing (modified)</td>
</tr>
<tr>
<td>tcp</td>
<td>SubSeven</td>
<td>[trojan] SubSeven</td>
</tr>
<tr>
<td>tcp</td>
<td>Subseven 2.1.4DefCon0</td>
<td>[trojan] Subseven 2.1.4 DefCon 0</td>
</tr>
<tr>
<td>tcp</td>
<td>ScheduleAgent</td>
<td>[trojan] ScheduleAgent</td>
</tr>
<tr>
<td>tcp</td>
<td>ScheduleAgent</td>
<td>[trojan] ScheduleAgent</td>
</tr>
<tr>
<td>tcp</td>
<td>Mosos</td>
<td>[trojan] Mosos</td>
</tr>
<tr>
<td>tcp</td>
<td>Manacrootkit</td>
<td>[trojan] Manacrootkit</td>
</tr>
<tr>
<td>tcp</td>
<td>kaitex</td>
<td>Kaitex Trojan</td>
</tr>
<tr>
<td>tcp</td>
<td>ircu</td>
<td>IRCU</td>
</tr>
<tr>
<td>tcp</td>
<td>irc</td>
<td>Internet Relay Chat</td>
</tr>
<tr>
<td>tcp</td>
<td>EGO</td>
<td>[trojan] EGO</td>
</tr>
<tr>
<td>tcp</td>
<td>WinSatan</td>
<td>[trojan] WinSatan</td>
</tr>
</tbody>
</table>

### User Comment

<table>
<thead>
<tr>
<th>Submitted By</th>
<th>Date</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deb Hale</td>
<td>2004-01-15 03:55:20</td>
<td>This port is used in conjunction with ports 901,902,903 by the Net-Devil virus.</td>
</tr>
<tr>
<td>Alex</td>
<td>2003-01-31 19:32:00</td>
<td>Be aware, this port is used by the W32.Netspree.Worm, see <a href="http://www.symantec.com">www.symantec.com</a> and search for the worm. (NAV don’t always pick it up...)</td>
</tr>
</tbody>
</table>

Log File After Egress Filtering Enabled

```bash
asa(config)# sh logging | grep 192.168.1.201 !truncated

%ASA-6-106100: access-list inside_access_in denied tcp
   inside/192.168.1.201(1761) -> outside/66.151.158.177(8200) hit-cnt 1 first hit [0xfe57d861, 0x0]

%ASA-6-106100: access-list inside_access_in denied tcp
   inside/192.168.1.201(1762) -> outside/66.151.158.177(8200) hit-cnt 1 first hit [0xfe57d861, 0x0]

%ASA-6-106100: access-list inside_access_in denied tcp
   inside/192.168.1.201(1763) -> outside/66.151.158.177(8200) hit-cnt 1 first hit [0xfe57d861, 0x0]

%ASA-6-302013: Built outbound TCP connection 4151 for
   outside:66.151.158.177/80 (66.151.158.177/80) to
   inside:192.168.1.201/1765 (xx.41.67.190/2699)

%ASA-5-304001: 192.168.1.201 Accessed URL
   66.151.158.177:/servlet/com.ec.ercbroker.servlets.PingServlet
```
Whose web site is that?

C:\Documents and Settings\ nslookup 66.151.158.177
  Server:  dns-133-lb-01.some.dns.com
  Address:  172.16.1.14

  Name:  poll.gotomypc.com
  Address:  66.151.158.177

• Our security policy forbids remote management of PCs
• How to block this application since it uses tcp/80?
• Application Inspection might provide a solution……..
Application Inspection
Modular Policy Framework (MPF)

Prior to 7.x All of My Flows Were Treated Pretty Much the Same

7.x and newer allow for more granular Application Inspection

PIX 6.3 Rules

PIX/ASA 7.x and 8.x Rules

Rules about HTTP
7.2 Adds Regular Expression Matching

- Regex matching provides an incredible amount of flexibility for application inspection.

- A regular expression is a string of characters that describes or matches a set of strings according to a certain syntax.

- Can focus on filename, content matching, string matching or any combination thereof.

- The PIX/ASA firewalls include a regex wizard that includes a useful testing facility.

Code Red v1

- Compromised Windows IIS web servers via buffer overflow in the indexing service
- Advisory released on June 18th, worm released July 12th, 2001
- 359K+ machines infected in the first 14 hours, with 2K per minute new infections at its peak
- Exploit string of Code Red v1:

```plaintext
GET /default.IDA?NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
Code Red vl Regex Matching

asa(config)# regex CodeRed1 "N{50,}"  
This string is looking for at least 50 occurrences of the character “N” which is a good indicator of CodeRed infected host activity
More Evil: Code Red II

♫ First seen in the wild on August 4, 2001 and spread faster than its predecessor, exploiting the same vulnerability

♫ More dangerous than Code Red v1 because it installed a trojaned version of explorer.exe in the root drive and root.exe in \inetpub\scripts

♫ Exploit string for Code Red v2:

```
GET /default.ida?XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXX%u9090%u6858%ucbd3%u7801%u9090%u6858%ucbd3%u7801%u9090%u6858%ucbd
3%u7801%u9090%u8190%u00c3%u0003%u8b00%u531b%u53ff
%u0078%u0000%u00=a HTTP/1.0
```

♫ Because of the backdoor Code Red II created, it was fairly trivial to detect its proliferation due to the fact that it attempted to call “root.exe” on the compromised web server
Code Red II Regex Matching

```bash
asa(config)# regex CodeRed2 "/root.exe"
```
Code Red I and II URI Filtering
The Evil Continues: Nimda

- September 18, 2001 over 1M hosts infected in 24 hours

- Multi-vector worm looking for any of several MSFT vulnerabilities (MIME, HTTP, etc.)

IIS Webserver Unicode Directory Traversal:

- GET /scripts/..%c1%9c../winnt/system32/cmd.exe?/c+dir
- GET /scripts/..%5c../winnt/system32/cmd.exe?/c+dir

- Mitigate in firewall by dropping any HTTP request that contains Non-ASCII headers

  asa(config)# **match request header non-ascii**
  drop-connection log
HACME Case Study

Continuing with Application Inspection
HACME Remote Office Part 2

• After implementing egress filtering, we noticed some interesting activity with host 192.168.1.201 talking via tcp/80 to Go2myPC.com

• Use the advanced HTTP inspection engine to implement a policy that blocks users from using that service, while allowing access to tcp/80 for allowed traffic
Case Study Part 2: Blocking Go2myPC

- Go2myPC uses tcp/8200 by default. If this is not available, the application will fall back to tcp/80 and tcp/443

- **Option #1**: Block and log all traffic types to domain name or ip address
  
  Not the best solution as this may not be possible in all contexts

- **Option #2**: Block and log tcp/8200 globally, and inspect all HTTP requests from 192.168.1.201
  
  Not a perfect solution, but let’s use the regex capabilities of the HTTP inspection engine

- We must understand the application that we’re attempting to filter, so we need to find a string that fingerprints the application

- The capture capability of the ASA is an excellent tool for this
Firewall Capture for Application Profiling

access-list HOSTCHECK extended permit ip host 192.168.1.201 any
access-list HOSTCHECK extended permit ip any host 192.168.1.201

asa>#capture CAP-HOST access-list HOSTCHECK interface inside

asa>#sh capture CAP-HOST

26 packets captured

...  
11: 192.168.1.201.1935 > 66.151.158.177.80: S
    4141954911:4141954911(0) win 64240 <mss 1460,nop,nop,sackOK>
12: 192.168.1.201.1936 > 66.151.158.177.8200: S
    999298953:999298953(0) win 64240 <mss 1460,nop,nop,sackOK>
13: 192.168.1.201.1937 > 66.151.158.177.443: S
    2462372042:2462372042(0) win 64240 <mss 1460,nop,nop,sackOK>
14: 66.151.158.177.80 > 192.168.1.201.1935: S
    2862815220:2862815220(0) ack 4141954912 win 8190 <mss 1380>
15: 192.168.1.201.1935 > 66.151.158.177.80: . ack 2862815221 win 64860
16: 192.168.1.201.1935 > 66.151.158.177.80: P
    4141954912:4141954975(63) ack 2862815221 win 64860
...

HTTP GET from website
Using Capture for Application Profiling

- An alternative option is to save the capture in .pcap format and view in a protocol analyzer.
- Using a browser: https://<fw_ipaddr>/capture/<capture name>/pcap/<filename>.pcap and save the file for analysis.
- Using Wireshark’s “Follow the TCP Stream” feature shows the startup of the Go2myPC application:

```
GET /servlet/com.ec.ercbroker.servlets.PingServlet
HTTP/1.0
HTTP/1.0 200 OK
Pragma: no-cache
Content-Type: text/plain
Content-Length: 41
ERCBroker broker http://www.gotomypc.com
```
Capture viewed in Wireshark
ASDM Regex Build Wizard

- This is the string to match on

- We chose to ignore case resulting in this string match
Case Study Part 2 – ASDM HTTP Inspect

- Our match criteria is to inspect the URI being sent by the client.

- Note that this HTTP inspect policy can be set to drop the connection, reset client and server (TCP RST) or just log the connection.

- Because our security policy does not allow for remote PC management, we are going to drop the connection and log.
Apply Service Policy To Interface

In this example, note that the service policy is only inspecting tcp/80. All other traffic is not bound to this particular policy.

We could have also selected more granular source and destination L3 addresses, if necessary.
Verification of HTTP Inspect

%ASA-6-302013: Built outbound TCP connection 5559 for
outside:66.151.158.177/80 (66.151.158.177/80) to
inside:192.168.1.201/1369 (xx.77.67.190/3184)

%ASA-6-106100: access-list inside_access_in denied tcp
inside/192.168.1.201(1370) -> outside/66.151.158.177(8200) hit-cnt 1 first hit [0xfe57d861, 0x0]

%ASA-5-304001: 192.168.1.201 Accessed URL
66.151.158.177:/servlet/com.ec.ercbroker.servlets.PingServlet

%ASA-5-415006: HTTP - matched request uri regex Block-Go2MyPC in
policy-map block-go2mypc URI matched - Dropping connection from
inside:192.168.1.201/1369 to outside:66.151.158.177/80

%ASA-6-302014: Teardown TCP connection 5559 for
outside:66.151.158.177/80 to inside:192.168.1.201/1369 duration
0:00:00 bytes 0 Flow closed by inspection

%ASA-5-304001: 192.168.1.201 Accessed URL
66.151.158.177:/servlet/com.ec.ercbroker.servlets.PingServlet

%ASA-5-415006: HTTP - matched request uri regex Block-Go2MyPC in
policy-map block-go2mypc URI matched - Dropping connection from
inside:192.168.1.201/1371 to outside:66.151.158.177/80
ASA 5580 and FWSM
### ASA 5580 and FWSM Comparison

<table>
<thead>
<tr>
<th>ASA 5580</th>
<th>FWSM</th>
</tr>
</thead>
<tbody>
<tr>
<td>4 RU Appliance with dual power supplies</td>
<td>Linecard for Catalyst 6500</td>
</tr>
<tr>
<td>Up to 10GB throughput</td>
<td>Up to 5GB throughput</td>
</tr>
<tr>
<td>Supports SSL VPN and IPSEC VPN termination</td>
<td>Supports IPSEC VPN for management ONLY</td>
</tr>
<tr>
<td>Line rate ACLs up to 500,000 ACEs supported</td>
<td>ACE limits vary by software release</td>
</tr>
<tr>
<td>Supports NetFlow</td>
<td>No Netflow support today</td>
</tr>
<tr>
<td>ASA codebase is same across platforms</td>
<td>FWSM code base is unique to FWSM</td>
</tr>
</tbody>
</table>
Which ASA code should I use?

- Tracking firewall code releases can be a very daunting task, even for Cisco folks
- Release Notes are your friend, read them!
- ASA Releases:
  - 7.0(8) GD status as of April 2008
  - 7.2 (4)
  - 8.0.4 ED status, introduced SSL VPN enhancements and EIGRP
  - 8.1.2 ED release for ASA 5580 platform ONLY
  - 8.2.1 ED released May 2009 for all ASA platforms
ASA 8.3 New Features
Some New Features in 8.3

(Network Object Optimization
ACL Time Stamp
Global Firewall Rules
NAT Simplification
Botnet Traffic Filters (BTF) Enhancement
Licensing Changes)
Botnet Traffic Filtering (BTF)
Q and A

Thank you