

Cisco Applied Mitigation Bulletin: Identifying and Mitigating Exploitation of the TCP State Manipulation Denial of Service Vulnerabilities in Multiple Cisco Products

<http://www.cisco.com/warp/public/707/cisco-amb-20090908-tcp24.shtml>

Revision 1.0

For Public Release 2009 September 08 1700 UTC (GMT)

Please provide your [feedback](#) on this document.

Contents

[Cisco Response](#)
[Device-Specific Mitigation and Identification](#)
[Additional Information](#)
[Revision History](#)
[Cisco Security Procedures](#)
[Related Information](#)

Cisco Response

This Applied Mitigation Bulletin is a companion document to the PSIRT Security Advisory *TCP State Manipulation Denial of Service Vulnerabilities in Multiple Cisco Products* and provides identification and mitigation techniques that administrators can deploy on Cisco network devices.

Vulnerability Characteristics

Multiple Cisco products contain multiple vulnerabilities when processing specially crafted Transmission Control Protocol (TCP) packets for an established TCP session. These vulnerabilities can be exploited remotely without authentication and without end-user interaction. Successful exploitation of these vulnerabilities may result in a denial of service (DoS) condition. Repeated attempts to exploit these vulnerabilities could result in a sustained DoS condition. The attack vector for exploitation is through

TCP.

These vulnerabilities have been assigned CVE identifier CVE-2008-4609.

The Cisco Nexus 5000 device contains an additional vulnerability when processing specially crafted TCP packets for an established TCP session. This vulnerability can be exploited remotely without authentication and without end-user interaction. Successful exploitation of this vulnerability may cause the affected device to crash. Repeated attempts to exploit this vulnerability could result in a sustained DoS condition. The attack vector for exploitation is through TCP.

This vulnerability has been assigned CVE identifier CVE-2009-0627.

Note: To successfully exploit these vulnerabilities, an attacker must complete the TCP three-way handshake and establish a TCP session with the vulnerable device.

Information about vulnerable, unaffected, and fixed software is available in the PSIRT Security Advisory, which is available at the following link: <http://www.cisco.com/warp/public/707/cisco-amb-20090908-tcp24.shtml>.

Mitigation Technique Overview

Cisco devices provide several countermeasures for these vulnerabilities. Administrators are advised to consider these protection methods to be general security best practices for infrastructure devices and the traffic that transits the network. This section of the document provides an overview of these techniques.

Cisco IOS[®] Software can provide effective means of exploit prevention using the following methods:

- Infrastructure access control lists (iACLs)
- Transit access control lists (tACLs)
- Receive access control lists (rACLs)
- Control Plane Policing (CoPP)
- Management Plane Protection (MPP)

These protection mechanisms filter and drop packets that are attempting to exploit these vulnerabilities.

Effective means of exploit prevention can also be provided by the Cisco ASA 5500 Series Adaptive Security Appliance, the Cisco PIX 500 Series Security Appliance, and the Firewall Services Module (FWSM) for Cisco Catalyst 6500 Series switches and Cisco 7600 Series routers using the following:

- Transit access control lists (tACLs)
- TCP Normalization
- Management Service Connection Filtering

These protection mechanisms filter and drop packets that are attempting to exploit these vulnerabilities.

Effective use of Cisco Embedded Event Manager (EEM) provides visibility into potential attacks attempting to exploit these vulnerabilities.

Simple Network Management Protocol (SNMP) provides a means for detecting and clearing hung TCP connections.

Effective use of Cisco Intrusion Prevention System (IPS) event actions provides visibility into and protection against attacks that attempt to exploit these vulnerabilities.

Cisco IOS NetFlow flow records can provide visibility into network-based exploitation attempts.

Cisco IOS Software, Cisco ASA appliances, Cisco PIX security appliances, and FWSM firewalls can provide visibility through syslog messages and the counter values displayed in the output from **show** commands.

Risk Management

Organizations are advised to follow their standard risk evaluation and mitigation processes to determine the potential impact of these vulnerabilities. Triage refers to sorting projects and prioritizing efforts that are most likely to be successful. Cisco has provided documents that can help organizations develop a risk-based triage capability for their information security teams. [Risk Triage for Security Vulnerability Announcements](#) and [Risk Triage and Prototyping](#) can help organizations develop repeatable security evaluation and response processes.

Device-Specific Mitigation and Identification



Caution: The effectiveness of any mitigation technique depends on specific customer situations such as product mix, network topology, traffic behavior, and organizational mission. As with any configuration change, evaluate the impact of these configurations prior to applying the change.

Specific information about mitigation and identification is available for these devices:

- [Cisco IOS Routers and Switches](#)
- [Cisco IOS NetFlow](#)
- [Cisco ASA, PIX, and FWSM Firewalls](#)

Cisco IOS Routers and Switches

Mitigation: Infrastructure Access Control Lists

To protect infrastructure devices and minimize the risk, impact, and effectiveness of direct infrastructure attacks, administrators are advised to deploy infrastructure access control lists (iACLs) to perform policy enforcement of traffic sent to infrastructure equipment. Administrators can construct an iACL by explicitly permitting only authorized traffic sent to infrastructure devices in accordance with existing security policies and configurations. For the maximum protection of infrastructure devices, deployed iACLs should be applied in the ingress direction on all interfaces to which an IP address has been configured. An iACL workaround cannot provide complete protection against these vulnerabilities when the attack originates from a trusted source address.

The iACL policy denies unauthorized TCP packets that are sent to affected devices. In the following example, 192.168.60.0/24 is the IP address space that is used by the affected devices, and the host at 192.168.100.1 is considered a trusted source that requires access to the affected devices. Care should be taken to allow required traffic for routing and administrative access prior to denying all unauthorized traffic. Whenever possible, infrastructure address space should be distinct from the address space used

for user and services segments. Using this addressing methodology will assist with the construction and deployment of iACLs.

Additional information about iACLs is in [Protecting Your Core: Infrastructure Protection Access Control Lists](#).

```
ip access-list extended Infrastructure-ACL-Policy

!
!-- Include explicit permit statements for trusted
!-- sources that require access to specific TCP
!-- services
!

permit tcp host 192.168.100.1 192.168.60.0 0.0.0.255

!
!-- The following vulnerability-specific access control
!-- entry (ACE) can aid in identification of attacks
!

deny tcp any 192.168.60.0 0.0.0.255

!
!-- Explicit deny ACE for traffic sent to addresses
!-- configured within the infrastructure address space
!

deny ip any 192.168.60.0 0.0.0.255

!
!-- Permit/deny all other Layer 3 and Layer 4 traffic
!-- in accordance with existing security policies and
!-- configurations
!
!-- Apply iACL to interfaces in the ingress direction
!

interface GigabitEthernet0/0
 ip access-group Infrastructure-ACL-Policy in

!
```

Note that filtering with an interface access list will elicit the transmission of ICMP unreachable messages back to the source of the filtered traffic. Generating these messages could have the undesired effect of increasing CPU utilization on the device. In Cisco IOS Software, ICMP unreachable generation is limited to one packet every 500 milliseconds by default. ICMP unreachable message generation can be disabled using the interface configuration command **no ip unreachable**. ICMP unreachable rate limiting can be changed from the default using the global configuration command **ip icmp rate-limit unreachable interval-in-ms**.

Mitigation: Transit Access Control Lists

To protect the network from traffic that enters the network at ingress access points, which may include Internet connection points, partner and supplier connection points, or VPN connection points,

administrators are advised to deploy transit access control lists (tACLs) to perform policy enforcement. Administrators can construct a tACL by explicitly permitting only authorized traffic to enter the network at ingress access points or permitting authorized traffic to transit the network in accordance with existing security policies and configurations. A tACL workaround cannot provide complete protection against these vulnerabilities when the attack originates from a trusted source address.

The tACL policy denies unauthorized TCP packets that are sent to affected devices. In the following example, 192.168.60.0/24 is the IP address space that is used by the affected devices, and the host at 192.168.100.1 is considered a trusted source that requires access to the affected devices. Care should be taken to allow required traffic for routing and administrative access prior to denying all unauthorized traffic.

Additional information about tACLs is in [Transit Access Control Lists: Filtering at Your Edge](#).

```
!-- Include any explicit permit statements for trusted
!-- sources that require access to the vulnerable TCP
!-- services
!

access-list 150 permit tcp host 192.168.100.1 192.168.60.0 0.0.0.255

!
!-- The following vulnerability-specific access control
!-- entry (ACE) can aid in identification of attacks
!

access-list 150 deny tcp any 192.168.60.0 0.0.0.255

!
!-- Permit/deny all other Layer 3 and Layer 4 traffic
!-- in accordance with existing security policies and
!-- configurations
!
!-- Explicit deny for all other IP traffic
!

access-list 150 deny ip any any

!
!-- Apply tACL to interfaces in the ingress direction
!

interface GigabitEthernet0/0
 ip access-group 150 in

!
```

Note that filtering with an interface access list will elicit the transmission of ICMP unreachable messages back to the source of the filtered traffic. Generating these messages could have the undesired effect of increasing CPU utilization on the device. In Cisco IOS Software, ICMP unreachable generation is limited to one packet every 500 milliseconds by default. ICMP unreachable message generation can be disabled using the interface configuration command **no ip unreachable**. ICMP unreachable rate limiting can be changed from the default using the global configuration command **ip icmp rate-limit unreachable interval-in-ms**.

Mitigation: Receive Access Control Lists

For distributed platforms, Receive ACLs (rACL) may be an option starting in Cisco IOS Software Versions 12.0(21)S2 for the 12000 (GSR), 12.0(24)S for the 7500, and 12.0(31)S for the 10720. The Receive ACL protects the device from harmful traffic before the traffic can impact the route processor. Receive ACLs are designed to only protect the device on which it is configured. On the 12000, transit traffic is never affected by an rACL. Because of this scenario, the destination IP address "any" used in the example ACL entries below only refer to the physical or virtual IP addresses of the router. On the 12000, 7500, and 10720, transit traffic is never affected by a receive ACL. Receive ACLs are considered a network security best practice and should be considered a long-term addition to good network security, as well as a workaround for these vulnerabilities. An rACL workaround cannot provide complete protection against these vulnerabilities when the attack originates from a trusted source address.

The rACL policy denies unauthorized TCP packets that are sent to the affected Cisco IOS device. In the following example, 192.168.60.0/24 is the IP address space that is used by the affected device, and the host at 192.168.100.1 is considered a trusted source that requires access to the affected devices. Care should be taken to allow required traffic for routing and administrative access prior to denying all unauthorized traffic.

Additional information about rACLs is in [GSR: Receive Access Control Lists](#) and [IP Receive ACL](#).

```
!-- Include any explicit permit statements for trusted
!-- sources that require access to the vulnerable TCP
!-- services
!

access-list 160 permit tcp host 192.168.100.1 192.168.60.0 0.0.0.255

!
!-- Deny all other sources to the vulnerable TCP services
!

access-list 160 deny tcp any 192.168.60.0 0.0.0.255

!
!-- Permit/deny all other Layer 3 and Layer 4 traffic
!-- in accordance with existing security policies and
!-- configurations
!
!-- Explicit deny for all other IP traffic received by
!-- the route processor (RP)
!

access-list 160 deny ip any any

!
!-- Apply access list 160 policy to the 'receive' path
!-- on the affected IOS device
!

ip receive access-list 160

!
```

Mitigation: Control Plane Policing

Control Plane Policing (CoPP) can be used to block untrusted TCP access to the device. Cisco IOS Software Releases 12.0S, 12.2SX, 12.2S, 12.3T, 12.4, and 12.4T support the CoPP feature. CoPP may be configured on a device to protect the management and control planes to minimize the risk and effectiveness of direct infrastructure attacks by explicitly permitting, and if configured, rate-limiting only authorized traffic that is sent to infrastructure devices in accordance with existing security policies and configurations. The following example, which uses 192.168.100.1 to represent a trusted host, can be adapted to your network.

```
!-- Deny TCP traffic from trusted hosts to all IP addresses
!-- configured on all interfaces of the affected device so
!-- that it will be allowed by the CoPP feature
!

access-list 111 deny tcp host 192.168.100.1 any

!
!-- Permit all other TCP traffic sent to all IP addresses
!-- configured on all interfaces of the affected device
!-- so it will be policed and dropped by the CoPP feature
!

access-list 111 permit tcp any any

!
!-- Permit (Police or Drop)/Deny (Allow) all other Layer3
!-- and Layer4 traffic in accordance with existing security
!-- policies and configurations for traffic that is authorized
!-- to be sent to infrastructure devices
!
!-- Create a Class-Map for traffic to be policed by the CoPP
!-- feature
!

class-map match-all drop-TCP-class
  match access-group 111

!
!-- Create a Policy-Map that will be applied to the Control-
!-- Plane of the device.
!

policy-map drop-TCP-traffic
  class drop-TCP-class
    drop

!
!-- Apply the Policy-Map to the Control-Plane of the device
!

control-plane
  service-policy input drop-TCP-traffic

!
```

In the above CoPP example, the access control list entries (ACEs), which match the potential exploit packets with the "permit" action, result in these packets being discarded by the policy-map "drop" function. Packets that match the "deny" action (not shown) are not affected by the policy-map drop function.

Please note that in the 12.2S and 12.0S Cisco IOS trains the policy-map syntax is different:

```
!  
policy-map drop-TCP-traffic  
  class drop-TCP-class  
    police 32000 1500 1500 conform-action drop exceed-action drop  
!
```

Additional information on the configuration and use of the CoPP feature can be found at [Control Plane Policing Implementation Best Practices](#) and [Control Plane Policing, IOS Software Release 12.2S](#).

Mitigation: Management Plane Protection

The Management Plane Protection (MPP) feature in Cisco IOS Software provides the capability to restrict the interfaces on which network management packets are allowed to enter a device. The MPP feature allows a network operator to designate one or more router interfaces as management interfaces. Device management traffic is permitted to enter a device only through these management interfaces. After MPP is enabled, no interfaces except designated management interfaces will accept network management traffic that is destined to the device. MPP cannot provide complete protection against these vulnerabilities when the attack connects to the address for the configured management interface.

Restricting management packets to designated interfaces provides greater control over management of a device, providing more security for that device. Other benefits include improved performance for data packets on non-management interfaces, support for network scalability, need for fewer access control lists (ACLs) to restrict access to a device, and management packet floods on switching and routing interfaces are prevented from reaching the CPU.

Additional information about MPP can be found at [Management Plane Protection](#).

The following example shows how to configure and enable the MPP to only allow SSH and HTTPS on the GigabitEthernet0/0 interface:

```
!-- Enter the control-plane host configuration mode.  
!  
control-plane host  
  
!  
!-- Configure the GigabitEthernet 0/0 interface to  
!-- be a management interface that only accepts SSH  
!-- and HTTPS management protocols  
  
management-interface GigabitEthernet 0/0 allow ssh https
```

!

Identification: Infrastructure Access Control Lists

After the administrator applies the iACL to an interface, the **show ip access-lists** command will identify the number of TCP packets that have been filtered on interfaces on which the iACL is applied. Administrators should investigate filtered packets to determine whether they are attempts to exploit these vulnerabilities. Example output for **show ip access-lists Infrastructure-ACL-Policy** follows:

```
router#show ip access-lists Infrastructure-ACL-Policy
Extended IP access list Infrastructure-ACL-Policy
 10 permit tcp host 192.168.100.1 192.168.60.0 0.0.0.255 (61 matches)
 20 deny tcp any 192.168.60.0 0.0.0.255 (72 matches)
 30 deny ip any 192.168.60.0 0.0.0.255 (59 matches)
router#
```

In the preceding example, access list *Infrastructure-ACL-Policy* has dropped **72 TCP** packets for access control list entry (ACE) line 20.

For additional information about investigating incidents using ACE counters and syslog events, reference the [Identifying Incidents Using Firewall and IOS Router Syslog Events](#) Applied Intelligence white paper.

Administrators can use Embedded Event Manager to provide instrumentation when specific conditions are met, such as ACE counter hits. The Applied Intelligence white paper [Embedded Event Manager in a Security Context](#) provides additional details about how to use this feature.

Identification: Transit Access Control Lists

After the administrator applies the tACL to an interface, the **show ip access-lists** command will identify the number of TCP packets that have been filtered. Administrators are advised to investigate filtered packets to determine whether they are attempts to exploit these vulnerabilities. Example output for **show ip access-lists 150** follows:

```
router#show ip access-lists 150
Extended IP access list 150
 10 permit tcp host 192.168.100.1 192.168.60.0 0.0.0.255
 20 deny tcp any 192.168.60.0 0.0.0.255 (97 matches)
 30 deny ip any any
router#
```

In the preceding example, access list *150* has dropped **97 TCP** packets for access control list entry (ACE) line 20.

For additional information about investigating incidents using ACE counters and syslog events, reference the [Identifying Incidents Using Firewall and IOS Router Syslog Events](#) Applied Intelligence white paper.

Administrators can use Embedded Event Manager to provide instrumentation when specific conditions are met, such as ACE counter hits. The Applied Intelligence white paper [Embedded Event Manager in a Security Context](#) provides additional details about how to use this feature.

Identification: Control Plane Policing

After the administrator applies the CoPP policy to the control-plane, the **show policy-map control-plane** command will display the number of TCP packets that have been filtered. Administrators are advised to investigate filtered packets to determine whether they are attempts to exploit these vulnerabilities. Example output for **show policy-map control-plane** follows:

```
router#show policy-map control-plane
Control Plane

Service-policy input: drop-TCP-traffic

Class-map: drop-TCP-class (match-all)
  371 packets, 14840 bytes
  5 minute offered rate 1024000 bps, drop rate 1024000 bps
  Match: access-group 111
  drop

Class-map: class-default (match-any)
  113 packets, 12588 bytes
  5 minute offered rate 0 bps, drop rate 0 bps
  Match: any

router#
```

In the preceding example, CoPP policy-map *drop-TCP-traffic* has dropped **371 TCP** packets for class-map *drop-TCP-class*.

The Simple Network Management Protocol (SNMP) can also be used to query a CoPP-enabled Cisco IOS device for counter values to determine how many packets have been dropped using the Cisco IOS Class-Based Quality of Service (QoS) Management Information Base (MIB) *CISCO-CLASS-BASED-QOS-MIB*. The Object Identifier (OID) 1.3.6.1.4.1.9.9.166 provides read access to all QoS configurations. Configuration information available through this MIB includes all class-map, policy-map, match statements, and action configuration parameters. Statistics available through this MIB include summary counts and rates by traffic class before and after any configured QoS policies are enforced. Detailed feature-specific statistics are available for select policy map features.

The following example shows how to query a device using SNMP GET requests by way of the Net-SNMP command, **snmpget**, on a UNIX host to retrieve the CoPP counter values:

Note: These counter values are identical to the counter values in the preceding example output from **show policy-map control-plane**.

```
unix#
unix# snmpget -m all -v 2c -c k0mm_n1tY 192.168.60.1 1.3.6.1.4.1.9.9.166.1.15.1
enterprises.cisco.ciscoMgmt.ciscoCBQoS-MIB.ciscoCBQoS-MIB-Objects.cbQoSClassMapSta
cbQoS-CMStatsTable.cbQoS-CMStatsEntry.cbQoS-CMPrePolicyPkt.1035.1037 = Counter32:
unix#
unix# snmpget -m all -v 2c -c k0mm_n1tY 192.168.60.1 1.3.6.1.4.1.9.9.166.1.15.1
enterprises.cisco.ciscoMgmt.ciscoCBQoS-MIB.ciscoCBQoS-MIB-Objects.cbQoSClassMapSta
cbQoS-CMStatsTable.cbQoS-CMStatsEntry.cbQoS-CMPrePolicyByte.1035.1037 = Counter32:
unix#
```

Identification: Access List Logging

The **log** and **log-input** access control list (ACL) option will cause packets that match specific ACEs to be logged. The **log-input** option enables logging of the ingress interface in addition to the packet source and destination IP addresses and ports.



Caution: Access control list logging can be very CPU intensive and must be used with extreme caution. Factors that drive the CPU impact of ACL logging are log generation, log transmission, and process switching to forward packets that match log-enabled ACEs.

For Cisco IOS Software, the **ip access-list logging interval** *interval-in-ms* command can limit the effects of process switching induced by ACL logging. The **logging rate-limit** *rate-per-second* [**except** *loglevel*] command limits the impact of log generation and transmission.

The CPU impact from ACL logging can be addressed in hardware on the Cisco Catalyst 6500 Series switches and Cisco 7600 Series routers with Supervisor Engine 720 or Supervisor Engine 32 using optimized ACL logging.

For additional information about the configuration and use of ACL logging, reference the [Understanding Access Control List Logging](#) Applied Intelligence white paper.

Identification: Embedded Event Manager

A Tcl-based Embedded Event Manager (EEM) policy can be used on vulnerable Cisco IOS devices to identify and detect a hung, long-lived, or indefinite TCP connection caused by these vulnerabilities. This policy allows administrators to monitor the TCP connections on a Cisco IOS device, and when EEM detects potential exploitation of these vulnerabilities, the EEM policy can trigger a response by sending a syslog message or an SNMP trap and clear the TCP connection. The example EEM policy provided in this document is based on a Tcl script that monitors and parses the output from two commands at defined intervals, produces a syslog message when the monitor threshold reaches its configured value, and can reset the TCP connection.

The identification and detection method uses a regular expression (regex) to parse the output from the **show tcp brief | include <CONNECTION-STATE>** and **show tcp tcb <0x0-0xFFFFFFFF>** commands. The regex parses the output for the *Transmission Control Block (TCB)*, *Source IP Address*, *Destination IP Address*, and *TCP Connection State* values and then validates the state of the TCP connection.

Script Implementation

To instrument and deploy the Tcl script, four EEM variables are used for the monitoring interval value expressed as an integer in seconds, the threshold value expressed as an integer for the number of retransmits to monitor, the clear connection value, which determines if the script should clear the hung TCP connection, expressed as *yes* value, and a connection state value or list of connection states. The four variables are:

- EEM_MONITOR_INTERVAL
- EEM_MONITOR_THRESHOLD
- EEM_MONITOR_CLEAR
- EEM_MONITOR_STATES

When the preceding variables have been configured with a value or values, the script will send a syslog message for each TCP connection identified and detected as hung. The script will then clear the connection if the *EEM_MONITOR_CLEAR* is defined with a value of *yes*.

EEM Detecting Hung TCP Connection

```
%HA_EM-6-LOG: monitor-sockets.tcl: Connection in FINWAIT1 found with
20 retransmissions: 192.168.60.1.443<-->192.168.1.1.41714
```

EEM Detecting And Clearing Hung TCP Connection

```
%HA_EM-6-LOG: monitor-sockets.tcl: Connection in FINWAIT1 found and
cleared with 20 retransmissions: 192.168.60.1.22<-->192.168.1.1.1273
```

The Tcl script can be copied to the vulnerable Cisco IOS device, and if the Cisco IOS file system supports directories, a directory for EEM policies can be created. The following example copies the script to the **EEM** directory of the **disk0:** file system:

```
router#mkdir disk0:/eem
Create directory filename [eem]?
Created dir disk0:/eem
router#copy <location of tcl script> disk0:/eem/monitor-sockets.tcl
```

The Tcl script can then be registered as an EEM policy, and the values for the script variables can be set with the following global configuration commands:

```
!
!-- Location where the Tcl script will be stored

event manager directory user policy disk0:/eem

!
!-- Define variable and set the monitoring interval
!-- as an integer (expressed in seconds)

event manager environment EEM_MONITOR_INTERVAL 60

!
!-- Define variable and set the threshold value as
!-- an integer for the number of retransmissions
!-- that determine if the TCP connection is hung
!-- (a recommended value to use is 15)

event manager environment EEM_MONITOR_THRESHOLD 15

!
!-- Define variable and set the value to "yes" to
!-- enable the clearing of hung TCP connections

event manager environment EEM_MONITOR_CLEAR yes

!
!-- Define variable and set to the TCP connection
!-- state or states the script will monitor which
!-- can be a single state or a space-separated list
```

```

!-- of states

event manager environment EEM_MONITOR_STATES FINWAIT1

!
!-- Register the script as an EEM policy

event manager policy monitor-sockets.tcl

!

```

The Tcl script is available for [download](#) at the [Cisco Beyond: Embedded Event Manager \(EEM\) Scripting Community](#).

Identification: Detecting and Clearing Hung TCP Connection Using SNMP

Simple Network Management Protocol (SNMP) can be used to detect and clear hung TCP connections on Cisco IOS devices that support the **TCP-MIB** and **CISCO-TCP-MIB** modules. Details describing this technique along with a script that can be used are available in the [How to Detect and Clear Hung TCP Connections using SNMP](#) white paper.

Cisco IOS NetFlow

Identification: Traffic Flow Identification Using NetFlow Records

Administrators can configure Cisco IOS NetFlow on Cisco IOS routers and switches to aid in the identification of traffic flows that may be attempts to exploit these vulnerabilities. Administrators are advised to investigate flows to determine whether they are attempts to exploit these vulnerabilities or whether they are legitimate traffic flows.

```

router#show ip cache flow
IP packet size distribution (132534914 total packets):
  1-32   64   96  128  160  192  224  256  288  320  352  384  416  448  480
  .002 .263 .250 .076 .075 .017 .070 .029 .073 .001 .001 .000 .001 .000 .003

  512  544  576 1024 1536 2048 2560 3072 3584 4096 4608
  .000 .002 .059 .051 .016 .000 .000 .000 .000 .000 .000

IP Flow Switching Cache, 4456704 bytes
  445 active, 65091 inactive, 13130729 added
  408766378 aged polls, 0 flow alloc failures
  Active flows timeout in 2 minutes
  Inactive flows timeout in 60 seconds
IP Sub Flow Cache, 533256 bytes
  445 active, 15939 inactive, 13130729 added, 13130729 added to flow
  0 alloc failures, 0 force free
  1 chunk, 48 chunks added
  last clearing of statistics never

```

Protocol	Total Flows	Flows /Sec	Packets /Flow	Bytes /Pkt	Packets /Sec	Active(Sec) /Flow	Idle(Sec) /Flow
TCP-Telnet	32402	0.0	9	46	0.0	3.5	29.3
TCP-FTP	6081	0.0	1	49	0.0	2.7	46.9
TCP-FTPD	3882	0.0	2219	500	2.1	0.2	41.9
TCP-WWW	207392	0.0	18	322	0.9	13.5	42.6
TCP-SMTP	6915	0.0	2	94	0.0	3.6	37.4

TCP-X	3859	0.0	1	42	0.0	0.0	43.4
TCP-BGP	3825	0.0	1	42	0.0	0.0	42.5
TCP-NNTP	3818	0.0	1	42	0.0	0.0	42.5
TCP-Frag	193	0.0	1	40	0.0	0.0	60.6
TCP-other	9288386	2.3	6	223	14.1	0.7	23.9
UDP-DNS	297998	0.0	3	65	0.2	26.2	50.2
UDP-NTP	461984	0.1	1	76	0.1	5.9	58.2
UDP-TFTP	373	0.0	1	60	0.0	0.2	60.4
UDP-other	1233452	0.3	46	129	14.3	26.3	50.6
ICMP	1085188	0.2	1	82	0.3	0.3	60.4
IGMP	189666	0.0	2	37	0.1	57.0	38.6
IP-other	304870	0.0	9	91	0.7	94.4	15.3
Total:	13130284	3.3	10	196	33.3	7.0	31.6

SrcIf	SrcIPAddress	DstIf	DstIPAddress	Pr	SrcP	DstP	Pkts
Gi0/0	192.168.100.76	Gi0/1	192.168.60.10	06	84EB	6A8E	10
Gi0/0	192.168.100.209	Gi0/1	192.168.60.10	06	0618	EB86	21
Gi0/0	192.168.1.1	Gi0/1	192.168.60.96	06	0592	0050	4
Gi0/0	192.168.100.33	Gi0/1	192.168.60.53	06	D00A	33DB	13
Gi0/1	192.168.150.70	Gi0/0	192.168.208.80	11	0035	0E94	3
Gi0/0	192.168.7.38	Gi0/1	192.168.60.10	06	0427	0050	5
Gi0/0	192.168.3.11	Gi0/1	192.168.60.9	06	0618	0050	7
Gi0/0	192.168.5.59	Gi0/1	192.168.60.128	06	072B	0016	4
Gi0/0	192.168.100.1	Gi0/1	192.168.60.176	06	FDBD	0050	5
Gi0/0	192.168.208.127	Gi0/0	172.18.104.132	06	85BE	1A29	16
Gi0/0	192.168.2.1	Gi0/1	192.168.60.10	06	09A2	0016	3
Gi0/0	192.168.4.2	Gi0/1	192.168.60.10	06	04AF	01BB	9
Gi0/0	192.168.100.1	Gi0/1	192.168.60.252	06	F135	0050	18
Gi0/1	192.168.132.44	Gi0/0	10.89.245.149	11	007B	007B	9
Gi0/0	192.168.6.1	Gi0/1	192.168.60.78	06	05B9	01BB	6

router#

In the preceding example, there are multiple flows for **TCP (Protocol (Pr) hex value 06)**. This traffic is sourced from untrusted hosts (that is, not from 192.168.100.1) and sent to addresses within the 192.168.60.0/24 address block, which is used for infrastructure devices. Administrators are advised to compare these flows to baseline utilization for TCP traffic sent to the infrastructure devices and also investigate the flows to determine whether flows that are sourced from untrusted hosts or networks are legitimate.

To view only the traffic flows for TCP packets sent to or from infrastructure devices, the command **show ip cache flow | include SrcIf|192.168.60.*_06_** will display the related TCP NetFlow records as shown here:

```
router#show ip cache flow | include SrcIf|192.168.60.*_06_
```

SrcIf	SrcIPAddress	DstIf	DstIPAddress	Pr	SrcP	DstP	Pkts
Gi0/0	192.168.100.76	Gi0/1	192.168.60.10	06	84EB	6A8E	10
Gi0/0	192.168.100.209	Gi0/1	192.168.60.10	06	0618	EB86	21
Gi0/0	192.168.1.1	Gi0/1	192.168.60.96	06	0592	0050	4
Gi0/0	192.168.100.33	Gi0/1	192.168.60.53	06	D00A	33DB	13
Gi0/0	192.168.7.38	Gi0/1	192.168.60.10	06	0427	0050	5
Gi0/0	192.168.3.11	Gi0/1	192.168.60.9	06	0618	0050	7
Gi0/0	192.168.5.59	Gi0/1	192.168.60.128	06	072B	0016	4
Gi0/0	192.168.2.1	Gi0/1	192.168.60.10	06	09A2	0016	3
Gi0/0	192.168.4.2	Gi0/1	192.168.60.10	06	04AF	01BB	9
Gi0/0	192.168.6.1	Gi0/1	192.168.60.78	06	05B9	01BB	6

router#

Cisco ASA, PIX, and FWSM Firewalls

Mitigation: Transit Access Control Lists

To protect the network from traffic that enters the network at ingress access points, which may include Internet connection points, partner and supplier connection points, or VPN connection points, administrators are advised to deploy tACLs to perform policy enforcement. Administrators can construct a tACL by explicitly permitting only authorized traffic to enter the network at ingress access points or permitting authorized traffic to transit the network in accordance with existing security policies and configurations. A tACL workaround cannot provide complete protection against these vulnerabilities when the attack originates from a trusted source address.

The tACL policy denies unauthorized TCP packets that are sent to affected devices. In the following example, 192.168.60.0/24 is the IP address space that is used by the affected devices, and the host at 192.168.100.1 is considered a trusted source that requires access to the affected devices. Care should be taken to allow required traffic for routing and administrative access prior to denying all unauthorized traffic.

Additional information about tACLs is in [Transit Access Control Lists: Filtering at Your Edge](#).

```
!  
!-- Include any explicit permit statements for trusted  
!-- sources that require access on the vulnerable TCP  
!-- services  
!  
access-list tACL-Policy extended permit tcp host 192.168.100.1 192.168.60.0 255  
  
!  
!-- The following vulnerability-specific access control  
!-- entry (ACE) can aid in identification of attacks  
!  
access-list tACL-Policy extended deny tcp any 192.168.60.0 255.255.255.0  
  
!  
!-- Permit/deny all other Layer 3 and Layer 4 traffic  
!-- in accordance with existing security policies and  
!-- configurations  
!  
!-- Explicit deny for all other IP traffic  
!  
access-list tACL-Policy extended deny ip any any  
  
!  
!-- Apply tACL to interface(s) in the ingress direction  
!  
access-group tACL-Policy in interface outside  
  
!
```

Mitigation: TCP Normalization

The *TCP Normalization* feature identifies abnormal packets that the security appliance can act on when they are detected; for example, the security appliance can allow, drop, or clear the packets. TCP normalization helps protect the security appliance from attacks. The TCP normalizer includes non-configurable actions and configurable actions. Typically, non-configurable actions that drop or clear connections apply to packets that are considered malicious.

Additional information about TCP Normalization is available in the [Cisco Security Appliance Command Line Configuration Guide, Preventing Network Attacks](#).

Mitigation: Management Service Connection Filtering

Management service filtering enables administrators to permit or deny hosts or networks authorized access to the management services (SSH, Telnet, or HTTPS) that are configured on the affected device. The following configuration example shows how administrators would only permit SSH or HTTPS access from the trusted device at IP address 192.168.100.1:

```
!-- Include any explicit 'ssh' or 'http' statements for
!-- trusted hosts or networks that require access to the
!-- SSH or HTTPS management services
!

ssh 192.168.100.1 255.255.255.255 inside
ssh timeout 10
ssh version 2
http server enable
http 192.168.100.1 255.255.255.255 inside

!
```

Note: The `http ip_address subnet_mask interface_name` command does not filter SSL-based WebVPN or SSLVPN connections sent to the firewall.

Additional information about filtering access to management services is available in the [Cisco Security Appliance Command Line Configuration Guide, Managing System Access](#).

Identification: Transit Access Control Lists

After the tACL has been applied to an interface, administrators can use the `show access-list` command to identify the number of TCP packets that have been filtered. Administrators are advised to investigate filtered packets to determine whether they are attempts to exploit these vulnerabilities. Example output for `show access-list tACL-Policy` follows:

```
firewall#show access-list tACL-Policy
access-list tACL-Policy line 1 extended permit tcp host 192.168.100.1
 192.168.60.0 255.255.255.0 (hitcnt=35)
access-list tACL-Policy line 2 extended deny tcp any 192.168.60.0
 255.255.255.0 (hitcnt=107)
access-list tACL-Policy line 3 extended deny ip any any (hitcnt=836)
firewall#
```

In the preceding example, access list *tACL-Policy* has dropped **107 TCP** packets sent to affected devices and received from an untrusted host or network. In addition, syslog message *106023* can provide valuable information, which includes the source and destination IP address, the source and destination port numbers, and the IP protocol for the denied packet.

Identification: Firewall Access List Syslog Messages

Firewall syslog message *106023* will be generated for packets denied by an access control entry (ACE) that does not have the **log** keyword present. Additional information about this syslog message is in [Cisco Security Appliance System Log Message - 106023](#).

Information about configuring syslog for the Cisco ASA 5500 Series Adaptive Security Appliance or the Cisco PIX 500 Series Security Appliance is in [Monitoring the Security Appliance - Configuring and Managing Logs](#). Information about configuring syslog on the FWSM for Cisco Catalyst 6500 Series switches and Cisco 7600 Series routers is in [Monitoring the Firewall Services Module](#).

In the following example, the **show logging | grep regex** command extracts syslog messages from the logging buffer on the firewall. These messages provide additional information about denied packets that could indicate potential attempts to exploit these vulnerabilities that is described in this document. It is possible to use different regular expressions with the **grep** keyword to search for specific data in the logged messages.

Additional information about regular expression syntax is in [Creating a Regular Expression](#).

```
firewall#show logging | grep 106023
Mar 25 2009 00:15:13: %ASA-4-106023: Deny tcp src outside:192.168.1.18/2719
    dst inside:192.168.60.191/80 by access-group "tACL-Policy"
Mar 25 2009 00:15:13: %ASA-4-106023: Deny tcp src outside:192.168.7.200/1258
    dst inside:192.168.60.33/23 by access-group "tACL-Policy"
Mar 25 2009 00:15:13: %ASA-4-106023: Deny tcp src outside:192.168.4.99/4810
    dst inside:192.168.60.240/443 by access-group "tACL-Policy"
Mar 25 2009 00:15:13: %ASA-4-106023: Deny tcp src outside:192.168.5.100/3011
    dst inside:192.168.60.115/22 by access-group "tACL-Policy"
Mar 25 2009 00:15:13: %ASA-4-106023: Deny tcp src outside:192.168.2.88/4527
    dst inside:192.168.60.38/443 by access-group "tACL-Policy"
Mar 25 2009 00:15:13: %ASA-4-106023: Deny tcp src outside:192.168.3.175/2950
    dst inside:192.168.60.250/22 by access-group "tACL-Policy"
Mar 25 2009 00:15:13: %ASA-4-106023: Deny tcp src outside:192.168.6.199/3819
    dst inside:192.168.60.250/80 by access-group "tACL-Policy"
firewall#
```

In the preceding example, the messages logged for the tACL *tACL-Policy* show **TCP** packets sent to the address block assigned to affected devices.

Additional information about syslog messages for ASA and PIX security appliances is in [Cisco Security Appliance System Log Messages](#). Additional information about syslog messages for the FWSM is in [Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Services Module Logging System Log Messages](#).

For additional information about investigating incidents using syslog events, reference the [Identifying Incidents Using Firewall and IOS Router Syslog Events](#) Applied Intelligence white paper.

Identification: TCP Normalization

The *TCP Normalization* feature can detect and drop some of the TCP state manipulation attacks by default. Specifically, the *Bad TCP flags (bad-tcp-flags)* counter may indicate potential exploitation of these vulnerabilities. Administrators can use the **show asp drop frame | grep TCP** command to display counters for TCP-based traffic analyzed by the TCP Normalizer. Example output for **show asp drop frame | grep TCP** follows:

```
firewall#show asp drop frame | grep TCP
  Invalid TCP Length (invalid-tcp-hdr-length)                36
  First TCP packet not SYN (tcp-not-syn)                    21744
  Bad TCP flags (bad-tcp-flags)                            3
  TCP failed 3 way handshake (tcp-3whs-failed)             474
  TCP RST/FIN out of order (tcp-rstfin-ooo)
  TCP SEQ in SYN/SYNACK invalid (tcp-seq-syn-diff)
  TCP packet failed PAWS test (tcp-paws-fail)                26
firewall#
```

In the preceding example, the *TCP Normalization* feature detected and dropped **35** TCP packets with bad flags. The *Bad TCP flags (bad-tcp-flags)* counter is incremented and the packet is dropped when the security appliance receives a TCP packet with invalid TCP flags in the TCP header. For example, a packet with both SYN and FIN TCP flags set will be dropped. The packet corruption may be caused by a bad cable or noise on the line. It may also be that a TCP endpoint is sending corrupted packets and an attack is in progress. Please use the packet *capture* feature to learn more about the origin of the packet.

Identification: Management Service Connection Filtering

After filtering for management services has been applied to an interface, administrators can use the **show logging | grep 71000(3|5): TCP** command to identify the number of TCP packets that have been filtered. Administrators are advised to investigate filtered packets to determine whether they are attempts to exploit these vulnerabilities.

Firewall syslog message *710003* will be generated for packets denied access to the management services on the affected device. Additional information about this syslog message is in [Cisco Security Appliance System Log Message - 710003](#).

Firewall syslog message *710005* will be generated for packets denied access to the management services on the affected device. Additional information about this syslog message is in [Cisco Security Appliance System Log Message - 710005](#).

Example output for **show logging | grep 71000(3|5): TCP** follows:

```
firewall#show logging | grep 71000(3|5): TCP
Mar 25 2009 10:07:31: %ASA-3-710003: TCP access denied by ACL from
  192.168.1.63/21932 to outside:192.168.60.254/22
Mar 25 2009 10:07:31: %ASA-7-710005: TCP request discarded from
  192.168.1.63/21932 to outside:192.168.60.254/22
Mar 25 2009 10:07:34: %ASA-3-710003: TCP access denied by ACL from
  192.168.6.63/11983 to outside:192.168.60.254/22
Mar 25 2009 10:07:34: %ASA-7-710005: TCP request discarded from
  192.168.6.63/11983 to outside:192.168.60.254/22
Mar 25 2009 10:07:40: %ASA-3-710003: TCP access denied by ACL from
  192.168.2.63/18402 to outside:192.168.60.254/443
Mar 25 2009 10:07:40: %ASA-7-710005: TCP request discarded from
  192.168.2.63/18402 to outside:192.168.60.254/443
Mar 25 2009 10:07:52: %ASA-3-710003: TCP access denied by ACL from
```

```
192.168.5.63/14588 to outside:192.168.60.254/443
Mar 25 2009 10:07:52: %ASA-7-710005: TCP request discarded from
192.168.5.63/14588 to outside:192.168.60.254/443
Mar 25 2009 10:08:16: %ASA-3-710003: TCP access denied by ACL from
192.168.3.63/15269 to outside:192.168.60.254/22
Mar 25 2009 10:08:16: %ASA-7-710005: TCP request discarded from
192.168.3.63/15269 to outside:192.168.60.254/22
Mar 25 2009 10:09:04: %ASA-3-710003: TCP access denied by ACL from
192.168.4.63/33314 to outside:192.168.60.254/22
Mar 25 2009 10:09:04: %ASA-7-710005: TCP request discarded from
192.168.4.63/33314 to outside:192.168.60.254/22
firewall#
```

In the preceding example, management service connection filtering has dropped packets sent to the affected device and received from an untrusted host or network. In addition, syslog messages 710003 and 710003 can provide valuable information, which includes the source and destination IP address, the source and destination port numbers, and the IP protocol for the denied packet.

Additional Information

THIS DOCUMENT IS PROVIDED ON AN "AS IS" BASIS AND DOES NOT IMPLY ANY KIND OF GUARANTEE OR WARRANTY, INCLUDING THE WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR USE. YOUR USE OF THE INFORMATION ON THE DOCUMENT OR MATERIALS LINKED FROM THE DOCUMENT IS AT YOUR OWN RISK. CISCO RESERVES THE RIGHT TO CHANGE OR UPDATE THIS DOCUMENT AT ANY TIME.

Revision History

Revision 1.0	2009-September-08	Initial public release.
--------------	-------------------	-------------------------

Cisco Security Procedures

Complete information on reporting security vulnerabilities in Cisco products, obtaining assistance with security incidents, and registering to receive security information from Cisco, is available on Cisco's worldwide website at

http://www.cisco.com/en/US/products/products_security_vulnerability_policy.html. This includes instructions for press inquiries regarding Cisco security notices. All Cisco security advisories are available at <http://www.cisco.com/go/psirt>.

Related Information

- [Cisco Applied Mitigation Bulletins](#)
- [Cisco Security Center](#)
- [Cisco Product Security Incident Response Team](#)
- [Cisco Security Vulnerability Policy](#)
- [Cisco Guide to Harden Cisco IOS Devices](#)
- [Cisco IOS NetFlow - Home Page on Cisco.com](#)
- [Cisco IOS NetFlow White Papers](#)
- [NetFlow Performance Analysis](#)

- [Cisco Network Foundation Protection White Papers](#)
 - [Cisco Network Foundation Protection Presentations](#)
 - [A Security-Oriented Approach to IP Addressing](#)
 - [Control Plane Policing Implementation Best Practices](#)
 - [Understanding Control Plane Protection](#)
 - [Cisco Firewall Products - Home Page on Cisco.com](#)
 - [Common Vulnerabilities and Exposures \(CVE\)](#)
-

Help us help you.

Please rate this document.

- Excellent
- Good
- Average
- Fair
- Poor

This document solved my problem.

- Yes
- No
- Just browsing

Suggestions for improvement:

(256 character limit)

Home	How to Buy	Login	Profile	Feedback	Site Map	Help
----------------------	----------------------------	-----------------------	-------------------------	--------------------------	--------------------------	----------------------

[Contacts & Feedback](#) | [Help](#) | [Site Map](#)

© 2008 - 2009 Cisco Systems, Inc. All rights reserved. [Terms & Conditions](#) | [Privacy Statement](#) | [Cookie Policy](#) | [Trademarks of Cisco Systems, Inc.](#)