Concepts & Examples
ScreenOS Reference Guide

Volume 4:
Attack Detection and Defense Mechanisms

Release 5.3.0, Rev. B

Juniper Networks, Inc.
1194 North Mathilda Avenue
Sunnyvale, CA 94089
USA
408-745-2000
www.juniper.net

Part Number: 093-1662-000, Revision B

FCC Statement

The following information is for FCC compliance of Class A devices: This equipment has been tested and found to comply with the limits for a Class A digital device, pursuant to part 15 of the FCC rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. The equipment generates, uses, and can radiate radio-frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference, in which case users will be required to correct the interference at their own expense.

The following information is for FCC compliance of Class B devices: The equipment described in this manual generates and may radiate radio-frequency energy. If it is not installed in accordance with Juniper Network's installation instructions, it may cause interference with radio and television reception. This equipment has been tested and found to comply with the limits for a Class B digital device in accordance with the specifications in part 15 of the FCC rules. These specifications are designed to provide reasonable protection against such interference in a residential installation. However, there is no guarantee that interference will not occur in a particular installation.

If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures:

- Reorient or relocate the receiving antenna.
- Increase the separation between the equipment and receiver.
- Consult the dealer or an experienced radio/TV technician for help.
- Connect the equipment to an outlet on a circuit different from that to which the receiver is connected.

Caution: Changes or modifications to this product could void the user's warranty and authority to operate this device.

Disclaimer

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR JUNIPER NETWORKS REPRESENTATIVE FOR A COPY.

Writers: Richard Biegel, Sharmila Kumar, Carrie Nowocin, Jozef Wroblewski

Editor: Lisa Eldridge

Table of Contents

About This Volume vii

Document Conventions viii
CLI Conventions viii
Illustration Conventions ix
Naming Conventions and Character Types x
WebUI Conventions xi
Juniper Networks Documentation xii

Chapter 1 Protecting a Network 1

Stages of an Attack 2
Detection and Defense Mechanisms 2
Exploit Monitoring 5
Example: Monitoring Attacks from the Untrust Zone 5

Chapter 2 Reconnaissance Deterrence 7

IP Address Sweep 8
Port Scanning 9
Network Reconnaissance Using IP Options 10
Operating System Probes 12
SYN and FIN Flags Set 12
FIN Flag Without ACK Flag 13
TCP Header Without Flags Set 14
Evasion Techniques 15
FIN Scan 15
Non-SYN Flags 15
IP Spoofing 18
Example: L3 IP Spoof Protection 20
Example: L2 IP Spoof Protection 22
IP Source Route Options 23

Chapter 3 Denial-of-Service (DoS) Attack Defenses 27

Firewall DoS Attacks 28
Session Table Flood 28
Source-Based and Destination-Based Session Limits 28
Example: Source-Based Session Limiting 29
Example: Destination-Based Session Limiting 30
Aggressive Aging 30
Example: Aggressively Aging Out Sessions 32
SYN-ACK-ACK Proxy Flood 32

Network DoS Attacks 34
SYN Flood 34
SYN Cookie 44
ICMP Flood 46
UDP Flood 47
Land Attack 48
OS-Specific DoS Attacks 49
Ping of Death 49
Teardrop Attack 50
WinNuke 51

Chapter 4 Content Monitoring and Filtering 53

Fragment Reassembly 54
Malicious URL Protection 54
Application Layer Gateway 55
Example: Blocking Malicious URLs in Packet Fragments 56
Antivirus Scanning 57
Scanning FTP Traffic 58
Scanning HTTP Traffic 59
HTTP MIME Extensions 60
HTTP Webmail 61
Scanning IMAP and POP3 Traffic 61
Scanning SMTP Traffic 63
Updating the AV Pattern File 64
Example: Automatic Update 66
Example: Manual Update 66
Spyware and Phishing Protection 67
Policy-Based AV Scanning 68
AV Scanner Global Settings 69
AV Resource Allotment 69
Fail-Mode Behavior 69
Maximum Content Size and Maximum Messages 70
HTTP Keep-Alive 70
HTTP Trickling 71
AV Scanner Profile Settings 72
Initiating an AV Profile 72
Example: Scanning All Traffic Types 73
Example: AV Scanning for SMTP and HTTP Traffic Only 73
AV Profile Settings 74
Anti-Spam Filtering 77
Black Lists and White Lists 77
Basic Configuration 78
Filtering Spam Traffic 78
Dropping Spam Messages 78
Defining a Black List 79
Defining a White List 79
Defining a Default Action 79
Defining a Spam-Blocking List 80

Web Filtering 80
Integrated Web Filtering 81
Example: Enable Web Filtering 82
Example: URL Category 83
Example: Web-Filtering Profile 85
SurfControl Servers 88
Web-Filtering Cache 88
Example: Cache Parameters 88
Redirect Web Filtering 89
Virtual Systems Support 90
Configuring Redirect Web Filtering 91
Example: Web Filtering Configuration 94

Chapter 5 Deep Inspection 97

Overview 98
Attack Object Database Server 102
Predefined Signature Packs 102
Updating Signature Packs 103
Before You Start Downloading 104
Example: Immediate Update 105
Example: Automatic Updates 106
Example: Automatic Notification and Immediate Update 107
Example: Manual Update 108
Attack Objects and Groups 110
Supported Protocols 112
Stateful Signatures 115
TCP Stream Signatures 116
Protocol Anomalies 116
Attack Object Groups 117
Changing Severity Levels 117
Example: Deep Inspection for P2P 118
Disabling Attack Objects 119
Attack Actions 120
Example: Attack Actions—Close Server, Close, Close Client 121
Brute Force Attack Actions 127
Brute Force Attack Objects 128
Brute Force Attack Target 128
Brute Force Attack Timeout 129
Example 1 129
Example 2 129
Example 3 130
Attack Logging 130
Example: Disabling Logging per Attack Group 130
Mapping Custom Services to Applications 132
Example: Mapping an Application to a Custom Service 133
Example: Application-to-Service Mapping for HTTP Attacks 135

Customized Attack Objects and Groups....136
User-Defined Stateful Signature Attack Objects....136
Regular Expressions....137
Example: User-Defined Stateful Signature Attack Objects....138
TCP Stream Signature Attack Objects....140
Example: User-Defined Stream Signature Attack Object....141
Configurable Protocol Anomaly Parameters....142
Example: Modifying Parameters....142
Negation....143
Example: Attack Object Negation....143
Granular Blocking of HTTP Components....147
ActiveX Controls....148
Java Applets....148
EXE Files....148
ZIP Files....148

Chapter 6 Suspicious Packet Attributes 151

ICMP Fragments....152
Large ICMP Packets....153
Bad IP Options....154
Unknown Protocols....155
IP Packet Fragments....156
SYN Fragments....157

Appendix A Contexts for User-Defined Signatures A-I

Index....IX-I

About This Volume

Volume 4: Attack Detection and Defense Mechanisms describes the Juniper Networks security options available in ScreenOS. Many of these are SCREEN options that you can enable at the security zone level. SCREEN options apply to traffic reaching the Juniper Networks security device through any interface bound to a zone for which you have enabled such options. SCREEN options offer protection against IP address and port scans, denial-of-service (DoS) attacks, and other kinds of malicious activity. You can apply other network security options, such as web filtering, antivirus checking, and intrusion detection and prevention (IDP), at the policy level. These options only apply to traffic under the jurisdiction of the policies in which they are enabled.

NOTE: The subject of policies is only presented in this volume peripherally, as it applies to the network security options that you can enable at the policy level. For a complete examination of policies, see "Policies" on page 163.

This volume contains the following chapters:

- Chapter 1, "Protecting a Network," outlines the basic stages of an attack and the firewall options available to combat the attacker at each stage.
- Chapter 2, "Reconnaissance Deterrence," describes the options available for blocking IP address sweeps, port scans, and attempts to discover the type of operating system (OS) of a targeted system.
- Chapter 3, "Denial-of-Service (DoS) Attack Defenses," explains firewall, network, and OS-specific DoS attacks and how ScreenOS mitigates such attacks.
- Chapter 4, "Content Monitoring and Filtering," describes how to protect HyperText Transfer Protocol (HTTP) and File Transfer Protocol (FTP) users from malicious uniform resource locators (URLs) and how to configure the Juniper Networks security device to work with third-party products to provide antivirus scanning, anti-spam, and web filtering.
- Chapter 5, "Deep Inspection," describes how to configure the Juniper Networks security device to obtain IDP attack object updates, how to create user-defined attack objects and attack object groups, and how to apply IDP at the policy level.
- Chapter 6, "Suspicious Packet Attributes," presents several SCREEN options that protect network resources from potential attacks indicated by unusual IP and ICMP packet attributes.

- Appendix A, “Contexts for User-Defined Signatures,” provides descriptions of contexts that you can specify when defining a stateful signature attack object.

Document Conventions

This document uses several types of conventions, which are introduced in the following sections:

- “CLI Conventions” on this page
- “Illustration Conventions” on page ix
- “Naming Conventions and Character Types” on page x
- “WebUI Conventions” on page xi

CLI Conventions

The following conventions are used to present the syntax of CLI commands in examples and in text.

In examples:

- Anything inside square brackets [ ] is optional.
- Anything inside braces **{ }** is required.
- If there is more than one choice, each choice is separated by a pipe (|). For example:

  ```
  set interface { ethernet1 | ethernet2 | ethernet3 } manage
  ```

  means “set the management options for the ethernet1, ethernet2, *or* ethernet3 interface.”

- Variables are in *italic* type:

  ```
  set admin user name1 password xyz
  ```

In text:

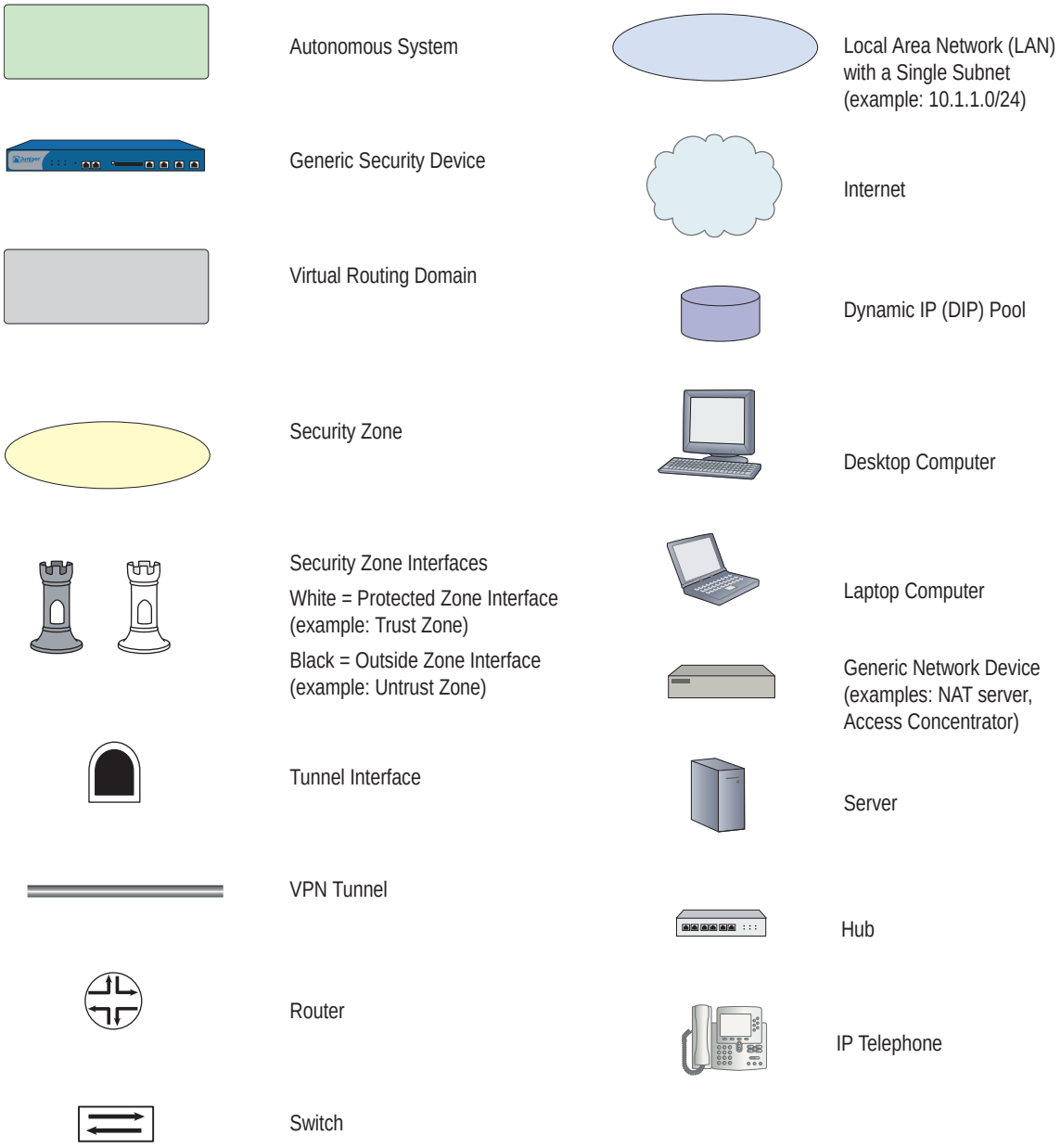
- Commands are in **boldface** type.
- Variables are in *italic* type.

NOTE: When typing a keyword, you only have to enter enough letters to identify the word uniquely. For example, entering **set adm u kath j12fmt54** is enough to enter the command **set admin user kathleen j12fmt54**. Although you can use this shortcut when entering commands, all the commands documented here are presented in their entirety.

Illustration Conventions

The following figure shows the basic set of images used in illustrations throughout this document.

Figure 1: Images in Illustrations



Naming Conventions and Character Types

ScreenOS employs the following conventions regarding the names of objects—such as addresses, admin users, auth servers, IKE gateways, virtual systems, VPN tunnels, and zones—defined in ScreenOS configurations:

- If a name string includes one or more spaces, the entire string must be enclosed within double quotes ("); for example:

  set address trust "local LAN" 10.1.1.0/24

- Any leading spaces or trailing text within a set of double quotes are trimmed; for example, **" local LAN "** becomes **"local LAN"**.

- Multiple consecutive spaces are treated as a single space.

- Name strings are case-sensitive, although many CLI keywords are case-insensitive. For example, **"local LAN"** is different from **"local lan"**.

ScreenOS supports the following character types:

- Single-byte character sets (SBCS) and multiple-byte character sets (MBCS). Examples of SBCS are ASCII, European, and Hebrew. Examples of MBCS—also referred to as double-byte character sets (DBCS)—are Chinese, Korean, and Japanese.

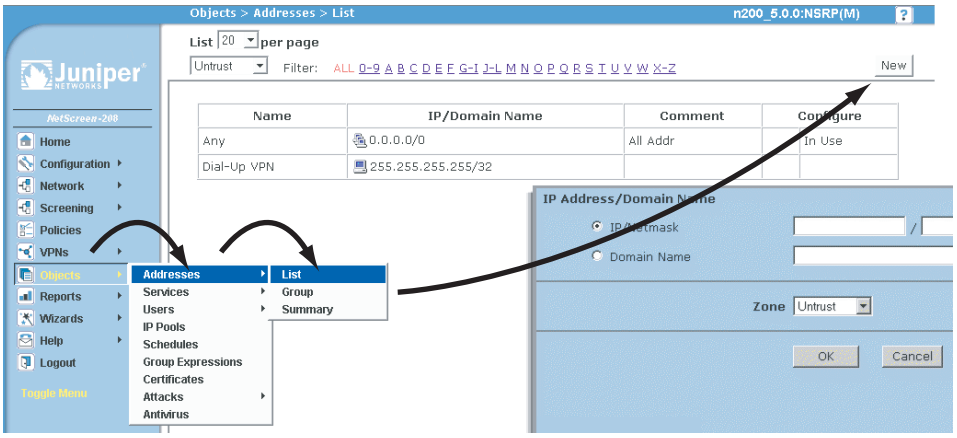
- ASCII characters from 32 (0x20 in hexadecimals) to 255 (0xff), except double quotes ("), which have special significance as an indicator of the beginning or end of a name string that includes spaces.

NOTE: A console connection only supports SBCS. The WebUI supports both SBCS and MBCS, depending on the character sets that your browser supports.

WebUI Conventions

A chevron (>) shows the navigational sequence through the WebUI, which you follow by clicking menu options and links. The following figure shows the following path to the address configuration dialog box—Objects > Addresses > List > New:

Figure 2: WebUI Navigation



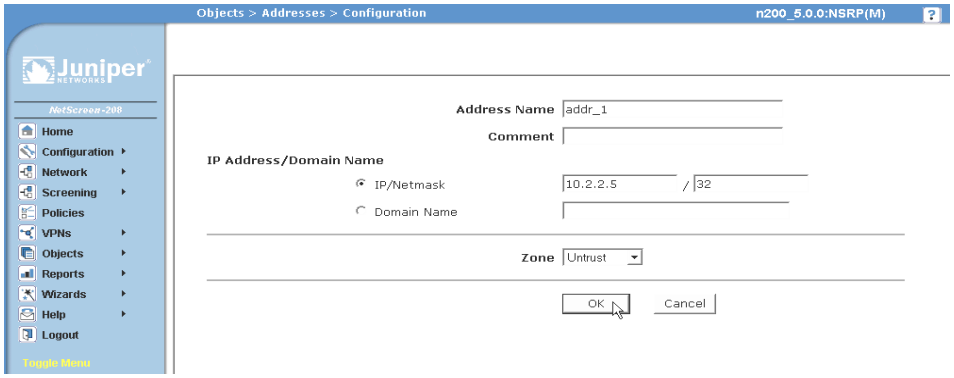
To perform a task with the WebUI, you first navigate to the appropriate dialog box, where you then define objects and set parameters. The set of instructions for each task is divided into navigational path and configuration settings:

The next figure lists the path to the address configuration dialog box with the following sample configuration settings:

Objects > Addresses > List > New: Enter the following, and then click **OK**:

Address Name: addr_1
IP Address/Domain Name:
IP/Netmask: (select), 10.2.2.5/32
Zone: Untrust

Figure 3: Navigational Path and Configuration Settings



Juniper Networks Documentation

To obtain technical documentation for any Juniper Networks product, visit www.juniper.net/techpubs/.

For technical support, open a support case using the Case Manager link at http://www.juniper.net/support/ or call 1-888-314-JTAC (within the United States) or 1-408-745-9500 (outside the United States).

If you find any errors or omissions in this document, please contact us at the email address below:

techpubs-comments@juniper.net

Chapter 1

Protecting a Network

There can be many reasons for invading a protected network. The following list contains some common objectives:

- Gathering the following kinds of information about the protected network:
  - Topology
  - IP addresses of active hosts
  - Numbers of active ports on active hosts
  - Operating systems of active hosts
- Overwhelming a host on a protected network with bogus traffic to induce a Denial-of-Service (DoS)
- Overwhelming the protected network with bogus traffic to induce a network-wide DoS
- Overwhelming a firewall with bogus traffic to induce a DoS for the network behind it
- Causing damage to and stealing data from a host on a protected network
- Gaining access to a host on a protected network to obtain information
- Gaining control of a host to launch other exploits
- Gaining control of a firewall to control access to the network that it protects

ScreenOS provides detective and defensive tools for uncovering and thwarting the efforts of attackers to achieve the above objectives when they attempt to target a network protected by a Juniper Networks security device.

This chapter presents an overview of the main stages of an attack and the various defense mechanisms that you can employ to thwart an attack at each stage:

- "Stages of an Attack" on page 2
- "Detection and Defense Mechanisms" on page 2
- "Exploit Monitoring" on page 5

Stages of an Attack

Each attack typically progresses in two major stages. In the first stage, the attacker gathers information, and in the second stage he or she launches the attack.

1. Perform reconnaissance.
    a. Map the network and determine which hosts are active (IP address sweep).
    b. Discern which ports are active (port scans) on the hosts discovered by the IP address sweep.
    c. Determine the operating system (OS), which might expose a weakness in the OS or suggest an attack to which that particular OS is susceptible.
2. Launch the attack.
    a. Conceal the origin of the attack.
    b. Perform the attack.
    c. Remove or hide evidence.

Detection and Defense Mechanisms

An exploit can be an information-gathering probe or an attack to compromise, disable, or harm a network or network resource. In some cases, the distinction between the two objectives of an exploit can be unclear. For example, a barrage of TCP SYN segments might be an IP address sweep with the intent of triggering responses from active hosts, or it might be a SYN flood attack with the intent of overwhelming a network so that it can no longer function properly. Furthermore, because an attacker usually precedes an attack by performing reconnaissance on the target, we can consider information-gathering efforts as a precursor to an impending attack—that is, they constitute the first stage of an attack. Thus, the term "exploit" encompasses both reconnaissance and attack activities, and the distinction between the two is not always clear.

Juniper Networks provides various detection methods and defense mechanisms at the zone and policy levels to combat exploits at all stages of their execution:

- SCREEN options at the zone level
- Firewall policies at the inter-, intra-, and super-zone policy levels (*super-zone* here means in global policies, where no security zones are referenced).

NOTE: Although the VLAN and MGT zones are function zones and not security zones, you can set SCREEN options for them. The VLAN zone supports the same set of SCREEN options as a Layer 3 security zone. (Layer 2 security zones support an additional SYN flood option that Layer 3 zones do not: Drop Unknown MAC). Because the following SCREEN options do not apply to the MGT zone, they are not available for that zone: SYN flood protection, SYN-ACK-ACK proxy flood protection, HTTP component blocking, and WinNuke attack protection.

To secure all connection attempts, Juniper Networks security devices use a dynamic packet-filtering method known as stateful inspection. Using this method, the security device notes various components in the IP packet and TCP segment headers— source and destination IP addresses, source and destination port numbers, and packet sequence numbers—and maintains the state of each TCP session and pseudo UDP session traversing the firewall. (The device also modifies session states based on changing elements such as dynamic port changes or session termination.) When a responding TCP packet arrives, the device compares the information reported in its header with the state of its associated session stored in the inspection table. If they match, the responding packet is allowed to pass the firewall. If the two do not match, the packet is dropped.

ScreenOS SCREEN options secure a zone by inspecting, then allowing or denying, all connection attempts that require crossing an interface bound to that zone. The security device then applies firewall policies, which can contain content filtering and intrusion detection and prevention (IDP) components, to the traffic that passes the SCREEN filters.

A Juniper Networks firewall provides the following sets of defense mechanisms:

- Reconnaissance deterrence
  - IP address sweep
  - Port scanning
  - Operating system probes
  - Evasion techniques
- Content monitoring and filtering
  - Fragment reassembly
  - Antivirus scanning
  - Anti-spam filtering
  - Web filtering
- Deep inspection
  - Stateful signatures
  - Protocol anomalies
  - Granular blocking of HTTP components

- Denial-of-Service (DoS) attack defenses
  - Firewall DoS attacks
    - Session table flood
    - SYN-ACK-ACK proxy flood
  - Network DoS attacks
    - SYN flood
    - ICMP flood
    - UDP flood
  - OS-specific DoS attacks
    - Ping of death
    - Teardrop attack
    - WinNuke
- Suspicious packet attributes
  - ICMP fragments
  - Large ICMP packets
  - Bad IP options
  - Unknown protocols
  - IP packet fragments
  - SYN fragments

ScreenOS network-protection settings operate at two levels: security zone and policy. The Juniper Networks security device performs reconnaissance deterrence and DoS attack defenses at the security zone level. In the area of content monitoring and filtering, the security device applies fragment reassembly at the zone level and antivirus (AV) scanning and uniform resource locator (URL) filtering at the policy level. The device applies IDP at the policy level, except for the detection and blocking of HTTP components, which occurs at the zone level. Zone-level firewall settings are SCREEN options. A network protection option set in a policy is a component of that policy.

Exploit Monitoring

Although you typically want the security device to block exploits, there might be times when you want to gather intelligence about them. You might want to learn specifically about a particular exploit—to discover its intention, its sophistication, and possibly (if the attacker is careless or unsophisticated) its source.

If you want to gather information about an exploit, you can let it occur, monitor it, analyze it, perform forensics, and then respond as delineated in a previously prepared incident response plan. You can instruct the security device to notify you of an exploit, but, instead of taking action, the device allows the exploit to transpire. You can then study what occurred, and try to understand the attacker's method, strategy, and objectives. Increased understanding of the threat to the network can then allow you to better fortify your defenses. Although a smart attacker can conceal his or her location and identity, you might be able to gather enough information to discern where the attack originated. You also might be able to estimate the attacker's capabilities. This kind of information allows you to gauge your response.

Example: Monitoring Attacks from the Untrust Zone

In this example, IP spoofing attacks from the Untrust zone have occurred on a daily basis, usually between 21:00 PM and 0:00 AM. Instead of dropping the packets with the spoofed source IP addresses, you want the security device to notify you of their arrival but allow them to pass, perhaps directing them to a honeypot (a decoy network server that is designed to lure attackers and then record their actions during an attack) that you have connected on the DMZ interface connection. At 20:55 PM, you change the firewall behavior from notification and rejection of packets belonging to a detected attack to notification and acceptance. When the attack occurs, you can then use the honeypot to monitor the attacker's activity after crossing the firewall. You might also work in cooperation with the upstream ISP to begin tracking the source of the packets back to their source.

WebUI

Screening > Screen (Zone: Untrust): Enter the following, then click **Apply**:

Generate Alarms without Dropping Packet: (select)
IP Address Spoof Protection: (select)

CLI

```
set zone untrust screen alarm-without-drop
set zone untrust screen ip-spoofing
save
```

Chapter 2

Reconnaissance Deterrence

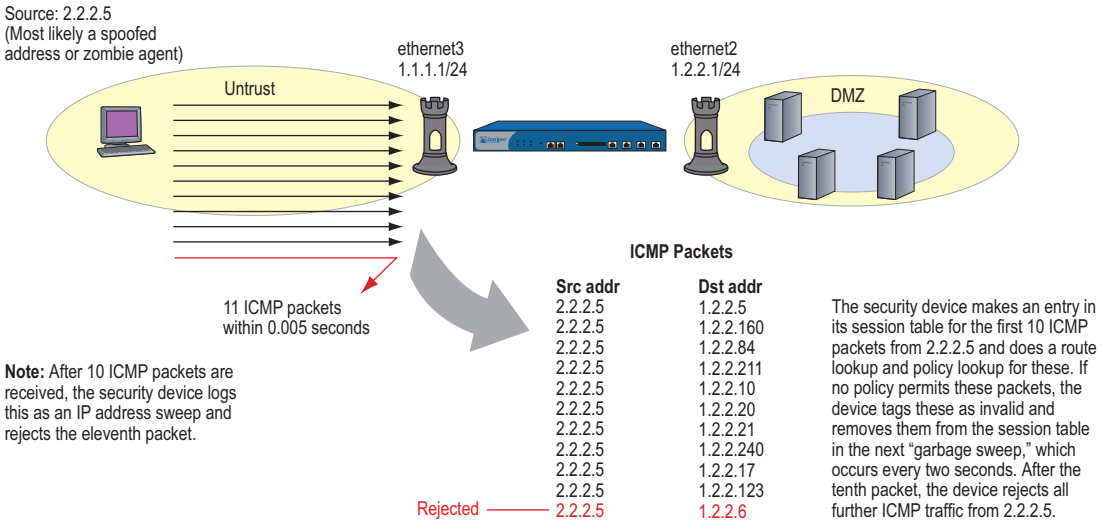
Attackers can better plan their attack when they first know the layout of the targeted network (which IP addresses have active hosts), the possible entry points (which port numbers are active on the active hosts), and the constitution of their victims (which operating system the active hosts are running). To gain this information, attackers must perform reconnaissance. Juniper Networks provides several SCREEN options to deter attackers' reconnaissance efforts and thereby hinder them from obtaining valuable information about the protected network and network resources.

- "IP Address Sweep" on page 8
- "Port Scanning" on page 9
- "Network Reconnaissance Using IP Options" on page 10
- "Operating System Probes" on page 12
  - "SYN and FIN Flags Set" on page 12
  - "FIN Flag Without ACK Flag" on page 13
  - "TCP Header Without Flags Set" on page 14
- "Evasion Techniques" on page 15
  - "FIN Scan" on page 15
  - "Non-SYN Flags" on page 15
  - "IP Spoofing" on page 18
  - "IP Source Route Options" on page 23

IP Address Sweep

An address sweep occurs when one source IP address sends 10 ICMP packets to different hosts within a defined interval (5000 microseconds is the default). The purpose of this scheme is to send ICMP packets—typically echo requests—to various hosts in the hopes that at least one replies, thus uncovering an address to target. The security device internally logs the number of ICMP packets to different addresses from one remote source. Using the default settings, if a remote host sends ICMP traffic to 10 addresses in 0.005 seconds (5000 microseconds), the security device flags this as an address sweep attack, and rejects all further ICMP echo requests from that host for the remainder of the specified threshold time period. The device detects and drops the tenth packet that meets the address sweep attack criterion.

Figure 1: Address Sweep



Consider enabling this SCREEN option for a security zone only if there is a policy permitting ICMP traffic from that zone. Otherwise, you do not need to enable it. The lack of such a policy denies all ICMP traffic from that zone, precluding an attacker from successfully performing an IP address sweep anyway.

To block IP address sweeps originating in a particular security zone, do either of the following:

WebUI

Screening > Screen (Zone: select a zone name): Enter the following, then click **Apply**:

IP Address Sweep Protection: (select)
Threshold: (enter a value to trigger IP address sweep protection)

NOTE: The value unit is microseconds. The default value is 5000 microseconds.

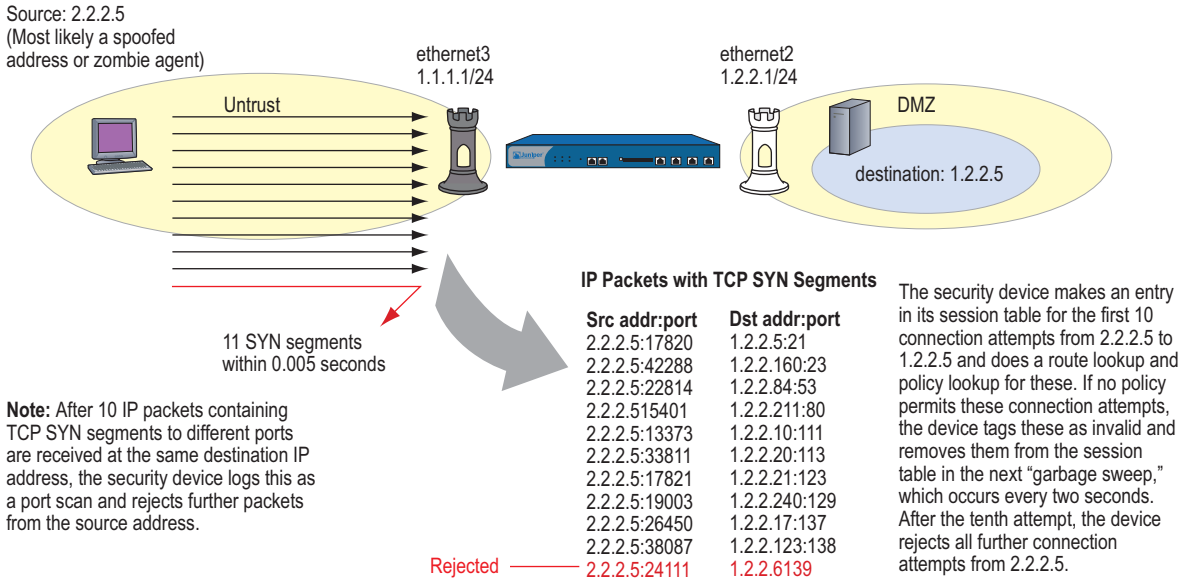
CLI

set zone *zone* screen ip-sweep threshold *number*
set zone *zone* screen ip-sweep

Port Scanning

A port scan occurs when one source IP address sends IP packets containing TCP SYN segments to 10 different ports at the same destination IP address within a defined interval (5000 microseconds is the default). The purpose of this scheme is to scan the available services in the hopes that at least one port will respond, thus identifying a service to target. The security device internally logs the number of different ports scanned from one remote source. Using the default settings, if a remote host scans 10 ports in 0.005 seconds (5000 microseconds), the device flags this as a port scan attack, and rejects all further packets from the remote source for the remainder of the specified timeout period. The device detects and drops the tenth packet that meets the port scan attack criterion.

Figure 2: Port Scan



To block port scans originating in a particular security zone, do either of the following:

WebUI

Screening > Screen (Zone: select a zone name): Enter the following, then click **Apply**:

Port Scan Protection: (select)
Threshold: (enter a value to trigger protection against port scans)

NOTE: The value unit is microseconds. The default value is 5000 microseconds.

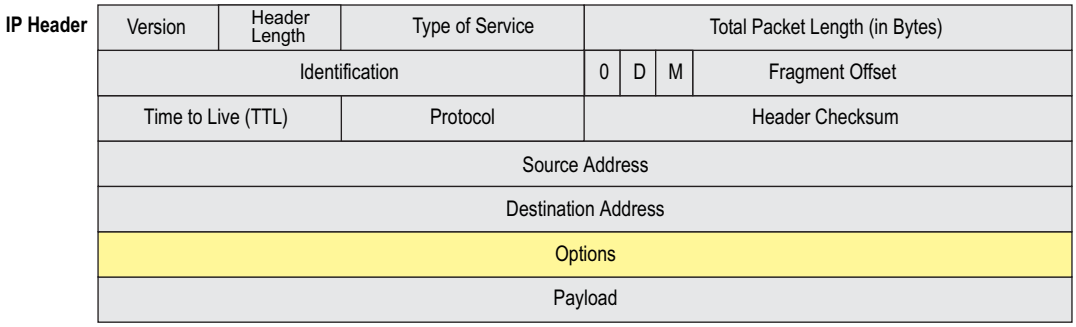
CLI

set zone *zone* screen port-scan threshold *number*
set zone *zone* screen port-scan

Network Reconnaissance Using IP Options

The Internet Protocol standard RFC 791, *Internet Protocol*, specifies a set of options to provide special routing controls, diagnostic tools, and security. These options appear after the destination address in an IP packet header, as shown in Figure 3.

Figure 3: Routing Options



RFC 791 states that these options are "unnecessary for the most common communications" and, in reality, they rarely appear in IP packet headers. When they do appear, they are frequently being put to some illegitimate use. Table 1 lists the IP options and their accompanying attributes.

Table 1: IP Options and Attributes

Type	Class	Number	Length	Intended Use	Nefarious Use
End of Options	0[1]	0	0	Indicates the end of one or more IP options.	None.
No Options	0	1	0	Indicates there are no IP options in the header.	None.
Security	0	2	11 bits	Provides a way for hosts to send security, TCC (closed user group) parameters, and Handling Restriction Codes compatible with Department of Defense (DoD) requirements. (This option, as specified in RFC 791, *Internet Protocol,* and RFC 1038, *Revised IP Security Option,* is obsolete.)	Unknown. However, because it is obsolete, its presence in an IP header is suspect.
Loose Source Route	0	3	Varies	Specifies a partial route list for a packet to take on its journey from source to destination. The packet must proceed in the order of addresses specified, but it is allowed to pass through other routers in between those specified.	Evasion. The attacker can use the specified routes to hide the true source of a packet or to gain access to a protected network. (See "IP Source Route Options" on page 23.)
Record Route	0	7	Varies	Records the IP addresses of the network devices along the path that the IP packet travels. The destination machine can then extract and process the route information. (Due to the size limitation of 40 bytes for both the option and storage space, this can only record up to 9 IP addresses.)	Reconnaissance. If the destination host is a compromised machine in the attacker's control, he or she can glean information about the topology and addressing scheme of the network through which the packet passed.
Stream ID	0	8	4 bits	(Obsolete) Provided a way for the 16-bit SATNET stream identifier to be carried through networks that did not support the stream concept.	Unknown. However, because it is obsolete, its presence in an IP header is suspect.

Type	Class	Number	Length	Intended Use	Nefarious Use
Strict Source Route	0	9	Varies	Specifies the complete route list for a packet to take on its journey from source to destination. The last address in the list replaces the address in the destination field.	Evasion. An attacker can use the specified routes to hide the true source of a packet or to gain access to a protected network. (See "IP Source Route Options" on page 23.)
Timestamp	2[2]	4		Records the time (in Universal Time[3]) when each network device receives the packet during its trip from the point of origin to its destination. The network devices are identified by IP number. This option develops a list of IP addresses of the routers along the path of the packet and the duration of transmission between each one.	Reconnaissance. If the destination host is a compromised machine in the attacker's control, he or she can glean information about the topology and addressing scheme of the network through which the packet passed.

1.The class of options identified as "0" was designed to provide extra packet or network control.
2.The class of options identified as "2" was designed diagnostics, debugging, and measurement.
3.The timestamp uses the number of milliseconds since midnight Universal Time (UT). UT is also known as "Greenwich Mean Time" (GMT), which is the basis for the international time standard.

The following SCREEN options detect IP options that an attacker can use for reconnaissance or for some unknown but suspect purpose:

- **Record Route:** The security device detects packets where the IP option is 7 (Record Route) and records the event in the SCREEN counters list for the ingress interface.
- **Timestamp:** The security device detects packets where the IP option list includes option 4 (Internet Timestamp) and records the event in the SCREEN counters list for the ingress interface.
- **Security:** The security device detects packets where the IP option is 2 (security) and records the event in the SCREEN counters list for the ingress interface.
- **Stream ID:** The security device detects packets where the IP option is 8 (Stream ID) and records the event in the SCREEN counters list for the ingress interface.

To detect packets with the above IP options set, do either of the following, where the specified security zone is the one from which the packets originate:

WebUI

Screening > Screen (Zone: select a zone name): Enter the following, then click **Apply**:

IP Record Route Option Detection: (select)
IP Timestamp Option Detection: (select)
IP Security Option Detection: (select)
IP Stream Option Detection: (select)

CLI

set zone *zone* screen ip-record-route
set zone *zone* screen ip-timestamp-opt
set zone *zone* screen ip-security-opt
set zone *zone* screen ip-stream-opt

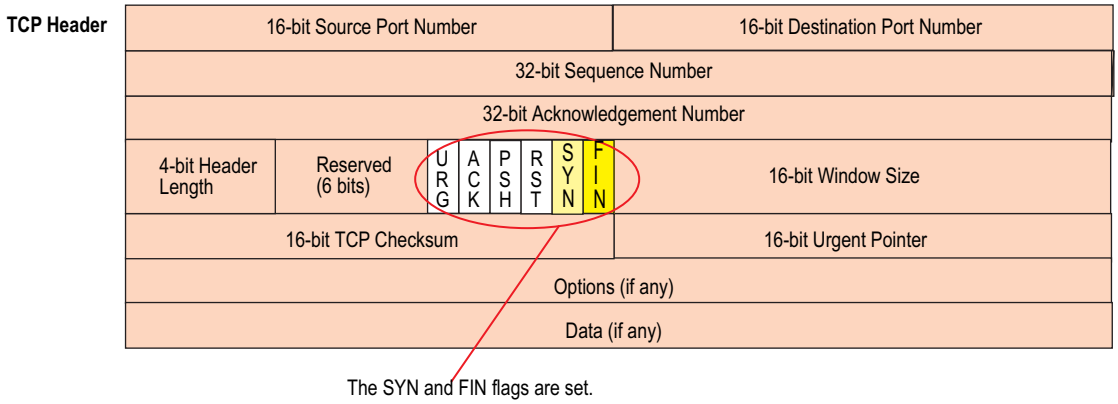
Operating System Probes

Before launching an exploit, an attacker might try to probe the targeted host to learn its operating system (OS). With that knowledge, he can better decide which attack to launch and which vulnerabilities to exploit. A Juniper Networks security device can block reconnaissance probes commonly used to gather information about OS types.

SYN and FIN Flags Set

Both the SYN and FIN control flags are not normally set in the same TCP segment header. The SYN flag synchronizes sequence numbers to initiate a TCP connection. The FIN flag indicates the end of data transmission to finish a TCP connection. Their purposes are mutually exclusive. A TCP header with the SYN and FIN flags set is anomalous TCP behavior, causing various responses from the recipient, depending on the OS. See Figure 4.

Figure 4: TCP Header with SYN and FIN Flags Set



An attacker can send a segment with both flags set to see what kind of system reply is returned and thereby determine what kind of OS is on the receiving end. The attacker can then use any known system vulnerabilities for further attacks.

When you enable this SCREEN option, the security device checks if the SYN and FIN flags are set in TCP headers. If it discovers such a header, it drops the packet.

To block packets with both the SYN and FIN flags set, do either of the following, where the specified security zone is the one from which the packets originate:

WebUI

Screening > Screen (Zone: select a zone name): Select **SYN and FIN Bits Set Protection**, then click **Apply**.

CLI

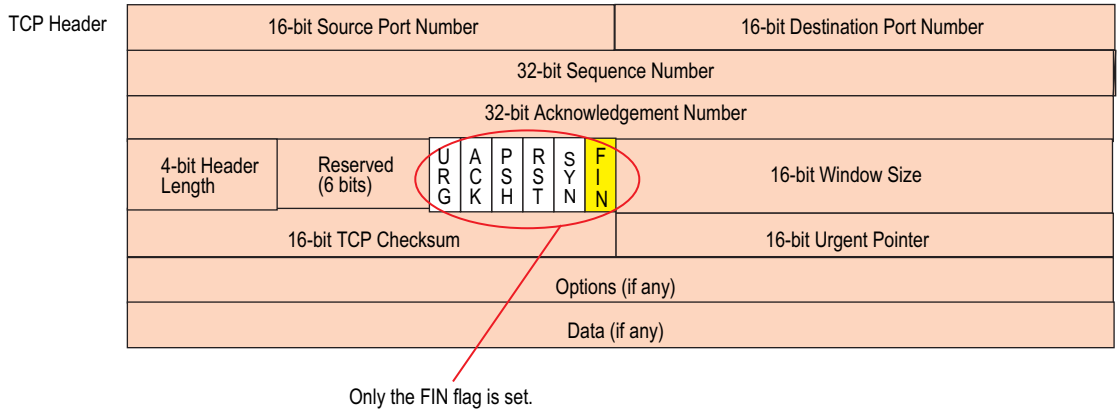
set zone *zone* screen syn-fin

FIN Flag Without ACK Flag

Figure 5 shows TCP segments with the FIN control flag set (to signal the conclusion of a session and terminate the connection). Normally, TCP segments with the FIN flag set also have the ACK flag set (to acknowledge the previous packet received). Because a TCP header with the FIN flag set but not the ACK flag is anomalous TCP behavior, there is no uniform response to this. The OS might respond by sending a TCP segment with the RST flag set. Another might completely ignore it. The victim's response can provide the attacker with a clue as to its OS. (Other purposes for sending a TCP segment with the FIN flag set are to evade detection while performing address and port scans and to evade defenses on guard for a SYN flood by performing a FIN flood instead. For information about FIN scans, see "FIN Scan" on page 15.)

NOTE: Vendors have interpreted RFC 793, *Transmission Control Protocol*, variously when designing their TCP/IP implementations. When a TCP segment arrives with the FIN flag set but not the ACK flag, some implementations send RST segments. Some drop the packet without sending an RST.

Figure 5: TCP Header with FIN Flag Set



When you enable this SCREEN option, the security device checks if the FIN flag is set but not the ACK flag in TCP headers. If it discovers a packet with such a header, it drops the packet.

To block packets with the FIN flag set but not the ACK flag, do either of the following, where the specified security zone is the one from which the packets originate:

WebUI

Screening > Screen (Zone: select a zone name): Select **FIN Bit with No ACK Bit in Flags Protection**, then click **Apply**.

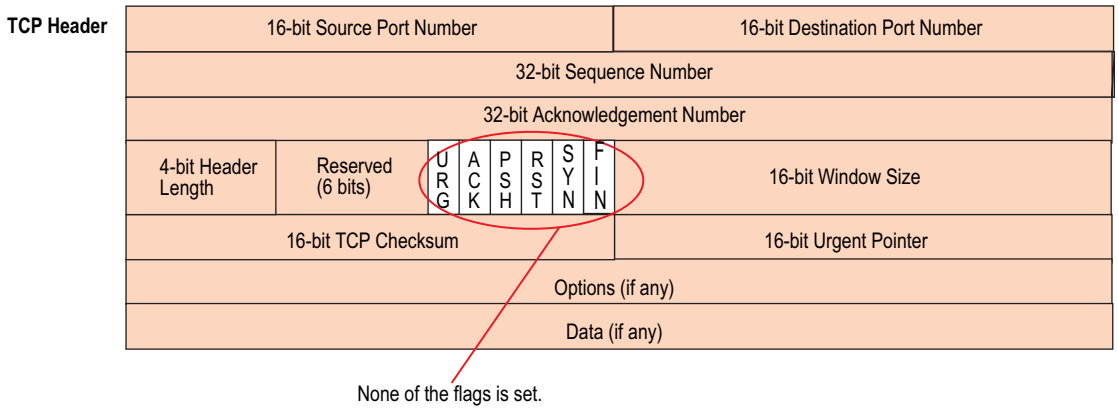
CLI

set zone *zone* screen fin-no-ack

TCP Header Without Flags Set

A normal TCP segment header has at least one flag control set. A TCP segment with no control flags set is an anomalous event. Because different operating systems respond differently to such anomalies, the response (or lack of response) from the targeted device can provide a clue as to the type of OS it is running. See Figure 6.

Figure 6: TCP Header with No Flags Set



When you enable the security device to detect TCP segment headers with no flags set, the device drops all TCP packets with a missing or malformed flags field.

To block packets with no flags set, do either of the following, where the specified security zone is the one from which the packets originate:

WebUI

Screening > Screen (Zone: select a zone name): Select **TCP Packet without Flag Protection**, then click **Apply**.

CLI

set zone *zone* screen tcp-no-flag

Evasion Techniques

Whether gathering information or launching an attack, it is generally expected that the attacker avoids detection. Although some IP address and port scans are blatant and easily detectable, more wily attackers use a variety of means to conceal their activity. Such techniques as using FIN scans instead of SYN scans—which attackers know most firewalls and intrusion detection programs detect—indicate an evolution of reconnaissance and exploit techniques to evade detection and successfully accomplish their tasks.

FIN Scan

A FIN scan sends TCP segments with the FIN flag set in an attempt to provoke a response (a TCP segment with the RST flag set) and thereby discover an active host or an active port on a host. An attacker might use this approach rather than perform an address sweep with ICMP echo requests or an address scan with SYN segments because he or she knows that many firewalls typically guard against the latter two approaches—but not necessarily against FIN segments. The use of TCP segments with the FIN flag set might evade detection and thereby help the attacker succeed in his or her reconnaissance efforts.

To thwart a FIN scan, you can do either or both of the following:

- Enable the SCREEN option that specifically blocks TCP segments with the FIN flag set but not the ACK flag, which is anomalous for a TCP segment:

  WebUI: Screening > Screen: Select the zone to which you want to apply this SCREEN option from the Zone drop-down list, and then select **FIN Bit With No ACK Bit in Flags Protection**.

  CLI: Enter **set zone** *name* **screen fin-no-ack**, in which *name* is the name of the zone to which you want to apply this SCREEN option

- Change the packet processing behavior to reject all non-SYN packets that do not belong to an existing session by entering the CLI command: **set flow tcp-syn-check**. (For more information about SYN flag checking, see "Non-SYN Flags" on page 15.)

NOTE: Changing the packet flow to check that the SYN flag is set for packets that do not belong to existing sessions also thwarts other types of non-SYN scans, such as a null scan (when no TCP flags are set).

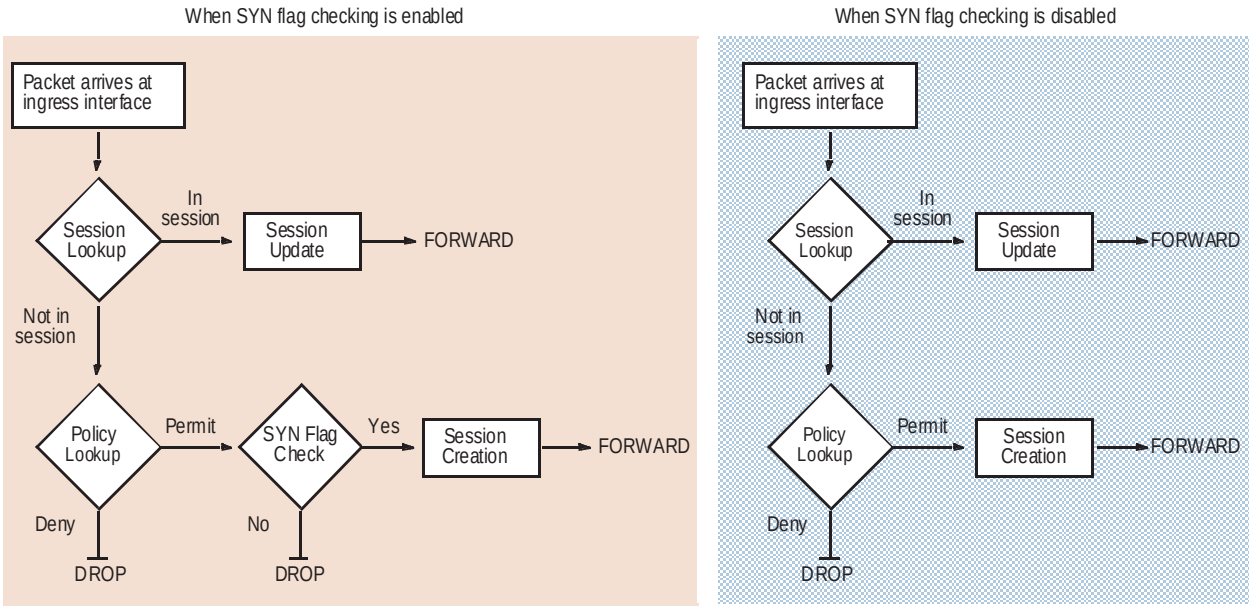
Non-SYN Flags

By default, the security device checks for SYN flags in the first packet of a session and rejects any TCP segments with non-SYN flags attempting to initiate a session. You can leave this packet flow as is or change it to so that the device does not enforce SYN flag checking before creating a session. Figure 7 on page 16 illustrates packet flow sequences when SYN flag checking is enabled and when it is disabled.

NOTE: By default, checking for the TCP SYN flag in the initial packet of a session is enabled when you install a Juniper Networks security device running ScreenOS 5.1.0 or higher. If you upgrade from a release prior to ScreenOS 5.1.0, SYN checking remains disabled by default—unless you have previously changed the default behavior.

These packet flows are the same whether the ingress interface is operating at Layer 3 (Route or NAT mode) or at Layer 2 (Transparent mode).

Figure 7: SYN Flag Checking



When the security device with SYN flag checking enabled receives a non-SYN TCP segment that does not belong to an existing session, it drops the packet and sends the source host to a TCP RST—unless the code bit of the initial non-SYN TCP packet is also RST. In that case, the security device simply drops the packet.

You can enable and disable SYN checking with the following CLI commands:

```
set flow tcp-syn-check
unset flow tcp-syn-check
```

Not checking for the SYN flag in the first packets offers the following advantages:

- **NSRP with Asymmetric Routing:** In an active/active NSRP configuration in a dynamic routing environment, a host might send the initial TCP segment with the SYN flag set to one security device (Device-A) but the SYN/ACK might be routed to the other security device in the cluster (Device-B). If this asymmetric routing occurs after Device-A has synchronized its session with Device-B, all is well. On the other hand, if the SYN/ACK response reaches Device-B before Device-A synchronizes the session and SYN checking is enabled, Device-B rejects the SYN/ACK, and the session cannot be established. With SYN checking disabled, Device-B accepts the SYN/ACK response—even though there is no existing session to which it belongs—and creates a new session table entry for it.

- **Uninterrupted Sessions:** If SYN checking is enabled and you add a security device operating in Transparent mode to a working network, it disrupts all existing sessions, which must then be restarted. For lengthy sessions, such as large data transfers or database backups, this can be a troublesome disruption. Similarly, if you reset the device or even change a component in the core section of a policy and SYN checking is enabled, all existing sessions or those sessions to which the policy change applies are disrupted and must be restarted. Disabling SYN checking avoids such disruptions to network traffic flows.

NOTE: A solution to this scenario is to install the security device with SYN checking disabled initially. Then, after a few hours—when established sessions are running through the device—enable SYN checking.

The core section in a policy contains the following main components: source and destination zones, source and destination addresses, one or more services, and an action.

However, note that the above advantages exact the following security sacrifices:

- **Reconnaissance Holes:** When an initial TCP segment with a non-SYN flag—such as ACK, URG, RST, FIN—arrives at a closed port, many operating systems (Windows, for example) respond with a TCP segment that has the RST flag set. If the port is open, then the recipient does not generate any response.

  By analyzing these responses or lack thereof, an intelligence gatherer can perform reconnaissance on the protected network and also on the ScreenOS policy set. If he sends a TCP segment with a non-SYN flag set and the policy permits it through, the destination host receiving such a segment might drop it and respond with a TCP segment that has the RST flag set. Such a response informs the perpetrator of the presence of an active host at a specific address and that the targeted port number is closed. The intelligence gatherer also learns that the firewall policy permits access to that port number on that host.

  By enabling SYN flag checking, the security device drops TCP segments without a SYN flag if they do not belong to an existing session. It does not return a TCP RST segment. Consequently, the scanner gets no replies regardless of the policy set or whether the port is open or closed on the targeted host.

- **Session Table Floods:** If SYN checking is disabled, an attacker can bypass the ScreenOS SYN flood protection feature by flooding a protected network with a barrage of TCP segments that have non-SYN flags set. Although the targeted hosts drop the packets—and possibly send TCP RST segments in reply—such a flood can fill up the session table of the security device. When the session table is full, the device cannot process new sessions for legitimate traffic.

  By enabling SYN checking and SYN flood protection, you can thwart this kind of attack. Checking that the SYN flag is set on the initial packet in a session forces all new sessions to begin with a TCP segment that has the SYN flag set. SYN flood protection then limits the number of TCP SYN segments per second so that the session table does not become overwhelmed.

NOTE: For information about session table floods, see "Session Table Flood" on page 28. For information about SYN floods, see "SYN Flood" on page 34.

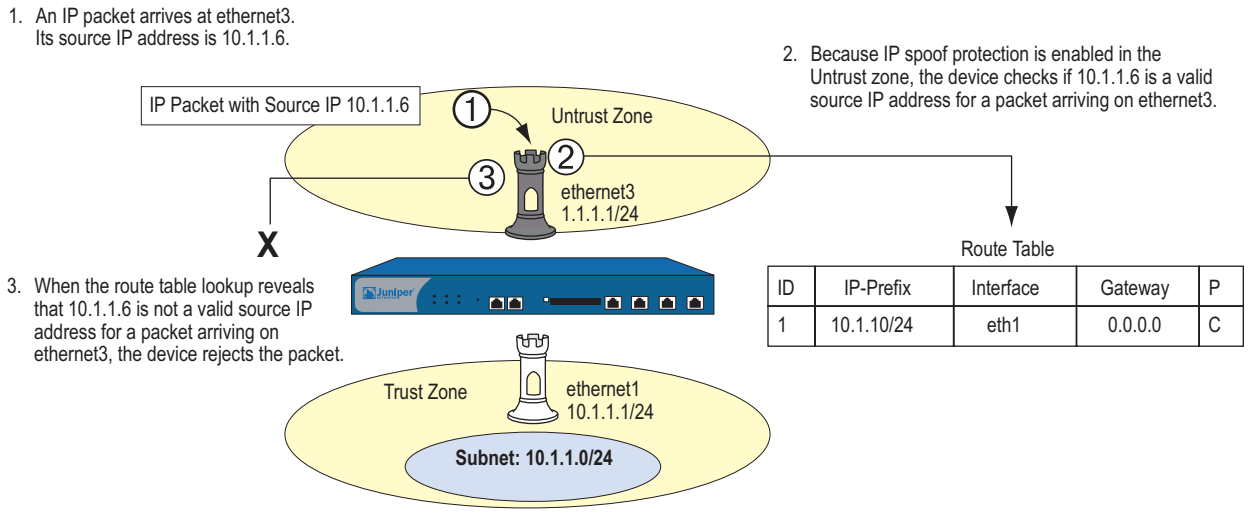
If you do not need SYN checking disabled, Juniper Networks strongly recommends that it be enabled (its default state for an initial installation of ScreenOS). You can enable it with the following command: **set flow tcp-syn-check**. With SYN checking enabled, the security device rejects TCP segments with non-SYN flags set unless they belong to an established session.

IP Spoofing

One method of attempting to gain access to a restricted area of the network is to insert a bogus source address in the packet header to make the packet appear to come from a trusted source. This technique is called IP spoofing. ScreenOS has two IP spoofing detection methods, both of which accomplish the same task: determining that the packet came from a location other than that indicated in its header. The method that a Juniper Networks security device uses depends on whether it is operating at Layer 3 or Layer 2 in the OSI Model.

- **Layer 3**—When interfaces on the security device are operating in Route or NAT mode, the mechanism to detect IP spoofing relies on route table entries. If, for example, a packet with source IP address 10.1.1.6 arrives at ethernet3, but the security device has a route to 10.1.1.0/24 through ethernet1, IP spoof checking notes that this address arrived at an invalid interface—as defined in the route table, a valid packet from 10.1.1.6 can only arrive via ethernet1, not ethernet3. Therefore, the device concludes that the packet has a spoofed source IP address and discards it.

Figure 8: Layer 3 IP Spoofing

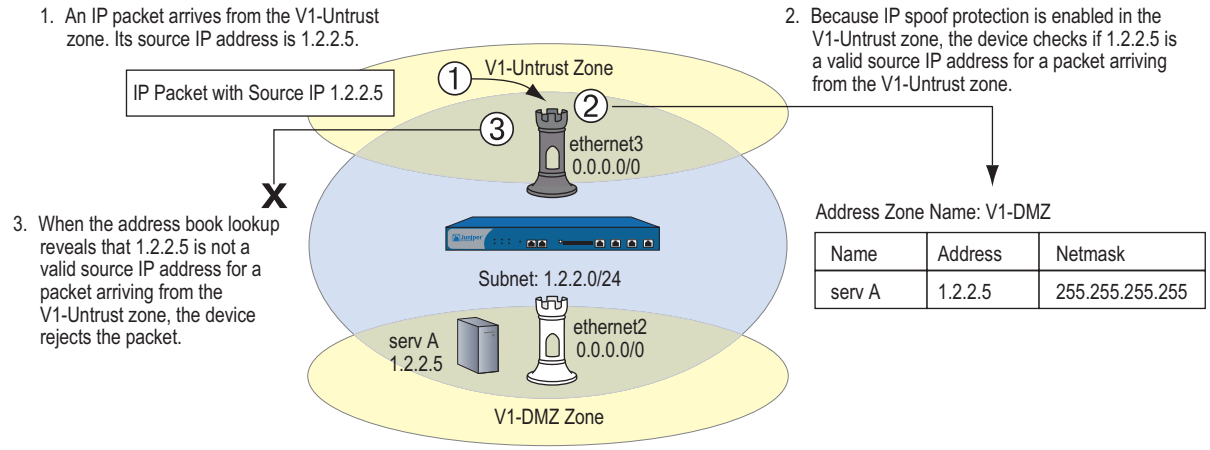


If the source IP address in a packet does not appear in the route table, by default the security device allows that packet to pass (assuming that a policy exists permitting it). Using the following CLI command—where the specified security zone is the one from which the packets originate—you can instruct the security device to drop any packet whose source IP address is not in the route table:

set zone *zone* screen ip-spoofing drop-no-rpf-route

- **Layer 2**—When interfaces on the security device are operating in Transparent mode, the IP spoof checking mechanism makes use of the address book entries. For example, you define an address for "serv A" as 1.2.2.5/32 in the V1-DMZ zone. If a packet with source IP address 1.2.2.5 arrives at a V1-Untrust zone interface (ethernet3), IP spoof checking notes that this address arrived at an invalid interface. The address belongs to the V1-DMZ zone, not to the V1-Untrust zone, and is accepted only at ethernet2, which is bound to V1-DMZ. The device concludes that the packet has a spoofed source IP address and discards it.

Figure 9: Layer 2 IP Spoofing



Be careful when defining addresses for the subnet that straddles multiple security zones. In Figure 9, 1.2.2.0/24 belongs to both the V1-Untrust and V1-DMZ zones. If you configure the security device as follows, the device will block traffic from the V1-DMZ zone that you want it to permit:

- You define an address for 1.2.2.0/24 in the V1-Untrust zone.
- You have a policy permitting traffic from any address in the V1-DMZ zone to any address in the V1-Untrust zone (**set policy from v1-dmz to v1-untrust any any any permit**).
- You enable IP spoof checking.

Because addresses in the V1-DMZ zone are also in the 1.2.2.0/24 subnet, when traffic from these addresses reaches ethernet2, the IP spoof check refers to the address book and finds 1.2.2.0/24 in the V1-Untrust zone. Consequently, the security device blocks the traffic.

Example: L3 IP Spoof Protection

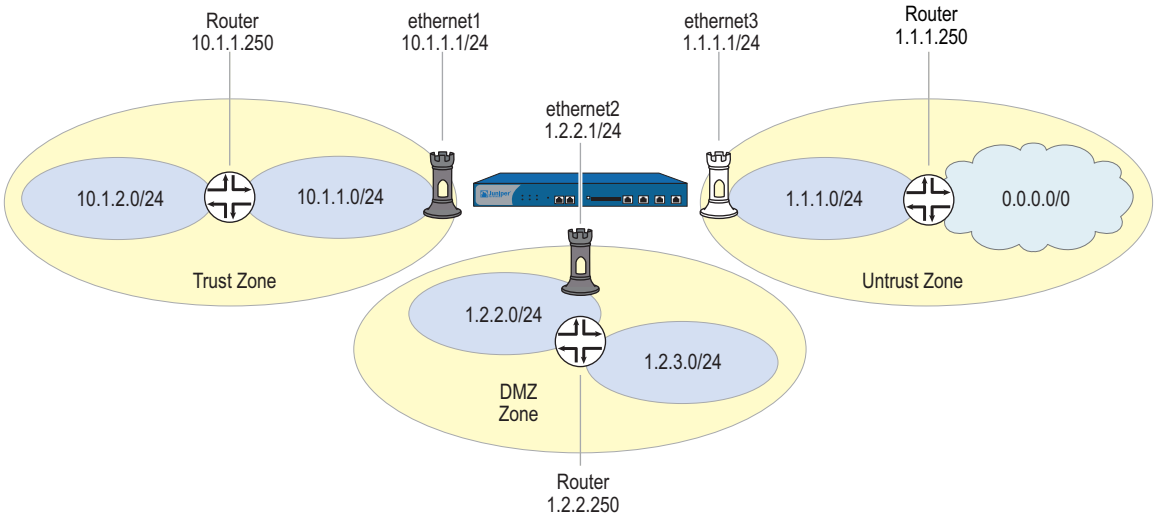
In this example, you enable IP spoof protection for the Trust, DMZ, and Untrust zones for a Juniper Networks security device operating at Layer 3. By default, the device automatically makes entries in the route table for the subnets specified in interface IP addresses. In addition to these automatic route table entries, you manually enter the three routes shown in the following table:

Destination	Egress Interface	Next Gateway
10.1.2.0/24	ethernet1	10.1.1.250
1.2.3.0/24	ethernet2	1.2.2.250
0.0.0.0/0	ethernet3	1.1.1.250

If you enable the IP spoof protection SCREEN option but do not enter the above three routes, the device drops all traffic from the addresses in the "Destination" column and enters alarms in the event log. For example, if a packet with the source address 10.1.2.5 arrives at ethernet1 and there is no route to the 10.1.2.0/24 subnet via ethernet1, the device determines that packet has arrived at an invalid interface and drops it.

All the security zones in this example are in the trust-vr routing domain.

Figure 10: Example of Layer 3 IP Spoofing



WebUI

1. **Interfaces**

   Network > Interfaces > Edit (for ethernet1): Enter the following, then click **Apply**:

   Zone Name: Trust
   Static IP: (select this option when present)
   IP Address/Netmask: 10.1.1.1/24

   Enter the following, then click **OK**:

   Interface Mode: NAT

   Network > Interfaces > Edit (for ethernet2): Enter the following, then click **OK**:

   Zone Name: DMZ
   Static IP: (select this option when present)
   IP Address/Netmask: 1.2.2.1/24

   Network > Interfaces > Edit (for ethernet3): Enter the following, then click **OK**:

   Zone Name: Untrust
   Static IP: (select this option when present)
   IP Address/Netmask: 1.1.1.1/24

2. **Routes**

   Network > Routing > Routing Entries > trust-vr New: Enter the following, then click **OK**:

   Network Address/Netmask: 10.1.2.0/24
   Gateway: (select)
   Interface: ethernet1
   Gateway IP Address: 10.1.1.250

   Network > Routing > Routing Entries > trust-vr New: Enter the following, then click **OK**:

   Network Address/Netmask: 1.2.3.0/24
   Gateway: (select)
   Interface: ethernet2
   Gateway IP Address: 1.2.2.250

   Network > Routing > Routing Entries > trust-vr New: Enter the following, then click **OK**:

   Network Address/Netmask: 0.0.0.0/0
   Gateway: (select)
   Interface: ethernet3
   Gateway IP Address: 1.1.1.250

3. **IP Spoof Protection**
   Screening > Screen (Zone: Trust): Select **IP Address Spoof Protection**, then click **Apply**.

   Screening > Screen (Zone: DMZ): Select **IP Address Spoof Protection**, then click **Apply**.

   Screening > Screen (Zone: Untrust): Select **IP Address Spoof Protection**, then click **Apply**.

CLI

1. **Interfaces**
   ```
   set interface ethernet1 zone trust
   set interface ethernet1 ip 10.1.1.1/24
   set interface ethernet1 nat
   set interface ethernet2 zone dmz
   set interface ethernet2 ip 1.2.2.1/24
   set interface ethernet3 zone untrust
   set interface ethernet3 ip 1.1.1.1/24
   ```

2. **Routes**
   ```
   set vrouter trust-vr route 10.1.2.0/24 interface ethernet1 gateway 10.1.1.250
   set vrouter trust-vr route 1.2.3.0/24 interface ethernet2 gateway 1.2.2.250
   set vrouter trust-vr route 0.0.0.0/0 interface ethernet3 gateway 1.1.1.250
   ```

3. **IP Spoof Protection**
   ```
   set zone trust screen ip-spoofing
   set zone dmz screen ip-spoofing
   set zone untrust screen ip-spoofing
   save
   ```

Example: L2 IP Spoof Protection

In this example, you protect the V1-DMZ zone from IP spoofing on traffic originating in the V1-Untrust zone. First, you define the following addresses for three webservers in the V1-DMZ zone:

- servA: 1.2.2.10
- servB: 1.2.2.20
- servC: 1.2.2.30

You then enable IP spoofing in the V1-Untrust zone.

If an attacker in the V1-Untrust zone attempts to spoof the source IP address using any of the three addresses in the V1-DMZ zone, the security device checks the address against those in the address books. When it finds that the source IP address on a packet coming from the V1-Untrust zone belongs to a defined address in the V1-DMZ zone, the device rejects the packet.

WebUI

1. **Addresses**
   Objects > Addresses > List > New: Enter the following, then click **OK**:

   Address Name: servA
   IP Address/Domain Name:
   IP/Netmask: (select), 1.2.2.10/32
   Zone: V1-DMZ

   Objects > Addresses > List > New: Enter the following, then click **OK**:

   Address Name: servB
   IP Address/Domain Name:
   IP/Netmask: (select), 1.2.2.20/32
   Zone: V1-DMZ

   Objects > Addresses > List > New: Enter the following, then click **OK**:

   Address Name: servC
   IP Address/Domain Name:
   IP/Netmask: (select), 1.2.2.30/32
   Zone: V1-DMZ

2. **IP Spoof Protection**
   Screening > Screen (Zone: V1-Trust): Select **IP Address Spoof Protection**, then click **Apply**.

CLI

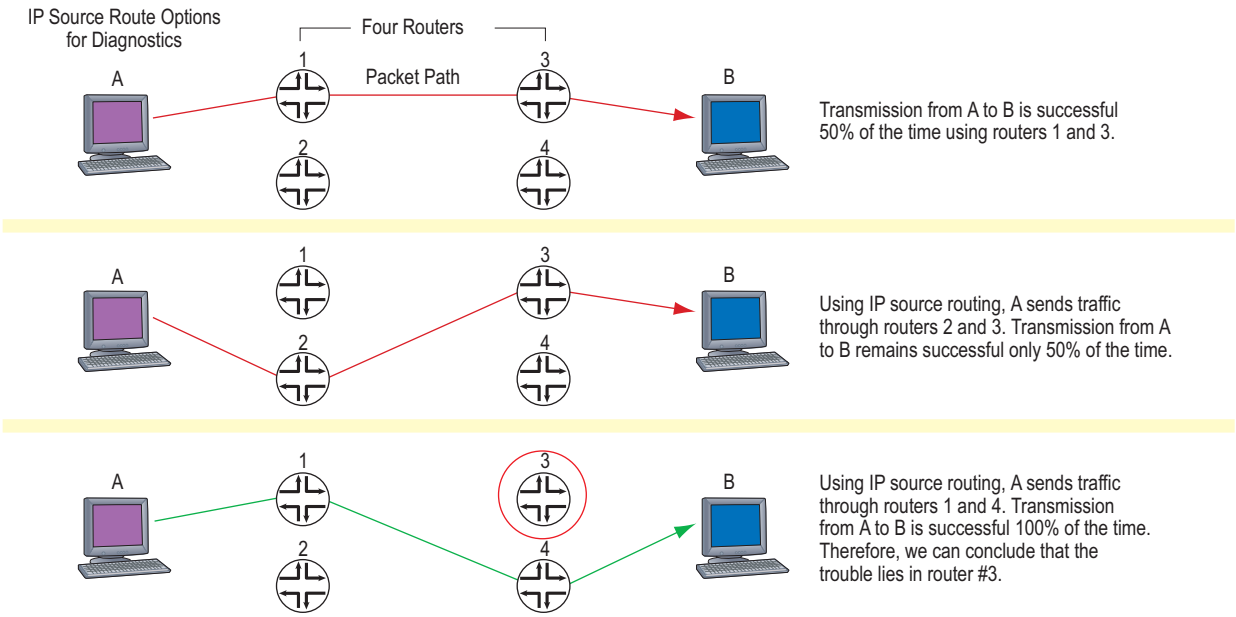
1. **Addresses**
   set address v1-dmz servA 1.2.2.10/32
   set address v1-dmz servB 1.2.2.20/32
   set address v1-dmz servC 1.2.2.30/32

2. **IP Spoof Protection**
   set zone v1-untrust screen ip-spoofing
   save

IP Source Route Options

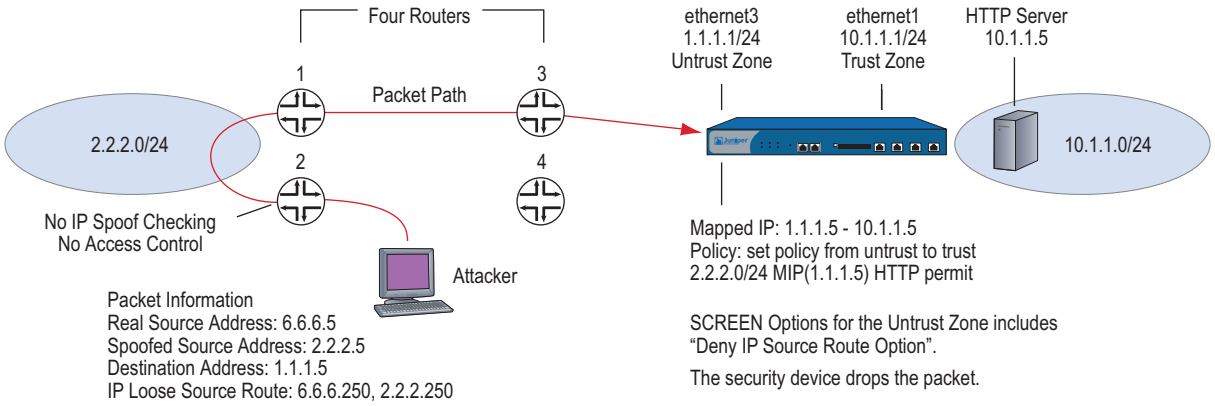
Source routing was designed to allow the user at the source of an IP packet transmission to specify the IP addresses of the routers (also referred to as "hops") along the path that he or she wants an IP packet to take on its way to its destination. The original intent of the IP source route options was to provide routing control tools to aid diagnostic analysis. If, for example, the transmission of a packet to a particular destination meets with irregular success, you might first use either the record route or timestamp IP option to discover the addresses of routers along the path or paths that the packet takes. You can then use either the loose or strict source route option to direct traffic along a specific path, using the addresses you learned from the results that the record route or timestamp options produced. By changing router addresses to alter the path and sending several packets along different paths, you can note changes that either improve or lessen the success rate. Through analysis and the process of elimination, you might be able to deduce where the trouble lies.

Figure 11: IP Source Routing



Although the uses of IP source route options were originally benign, attackers have learned to put them to more devious uses. They can use IP source route options to hide their true address and access restricted areas of a network by specifying a different path. For an example showing how an attacker can put both deceptions to use, consider the following scenario as illustrated in Figure 12.

Figure 12: Loose IP Source Route Option for Deception



The Juniper Networks security device only allows traffic 2.2.2.0/24 if it comes through ethernet1, an interface bound to the Untrust zone. Routers 3 and 4 enforce access controls but routers 1 and 2 do not. Furthermore, router 2 does not check for IP spoofing. The attacker spoofs the source address, and by using the loose source route option, directs the packet through router 2 to the 2.2.2.0/24 network and from there out router 1. Router 1 forwards it to router 3, which forwards it to the security device. Because the packet came from the 2.2.2.0/24 subnet and has a source address from that subnet, it seems to be valid. However, one remnant of the earlier chicanery remains: the loose source route option. In this example, you have enabled the "Deny IP Source Route Option" SCREEN option for the Untrust zone. When the packet arrives at ethernet3, the device rejects it.

You can enable the security device to either block any packets with loose or strict source route options set or detect such packets and then record the event in the counters list for the ingress interface. The SCREEN options are as follows:

- **Deny IP Source Route Option:** Enable this option to block all IP traffic that employs the loose or strict source route option. Source route options can allow an attacker to enter a network with a false IP address.

- **Detect IP Loose Source Route Option:** The security device detects packets where the IP option is 3 (Loose Source Routing) and records the event in the SCREEN counters list for the ingress interface. This option specifies a partial route list for a packet to take on its journey from source to destination. The packet must proceed in the order of addresses specified, but it is allowed to pass through other routers in between those specified.

- **Detect IP Strict Source Route Option:** The security device detects packets where the IP option is 9 (Strict Source Routing) and records the event in the SCREEN counters list for the ingress interface. This option specifies the complete route list for a packet to take on its journey from source to destination. The last address in the list replaces the address in the destination field.

  (For more information about all the IP options, see "Network Reconnaissance Using IP Options" on page 10.)

To block packets with either a loose or strict source route option set, do either of the following, where the specified security zone is the one from which the packets originate:

WebUI

Screening > Screen (Zone: select a zone name): Select **IP Source Route Option Filter**, then click **Apply**.

CLI

set zone *zone* screen ip-filter-src

To detect and record (but not block) packets with a loose or strict source route option set, do either of the following, where the specified security zone is the one from which the packets originate:

WebUI

Screening > Screen (Zone: select a zone name): Enter the following, then click **Apply**:

IP Loose Source Route Option Detection: (select)
IP Strict Source Route Option Detection: (select)

CLI

set zone *zone* screen ip-loose-src-route
set zone *zone* screen ip-strict-src-route

Chapter 3

Denial-of-Service (DoS) Attack Defenses

The intent of a denial-of-service (DoS) attack is to overwhelm the targeted victim with a tremendous amount of bogus traffic so that the victim becomes so preoccupied processing the bogus traffic that it is unable to process legitimate traffic. The target can be the Juniper Networks firewall, the network resources to which the firewall controls access, or the specific hardware platform or operating system (OS) of an individual host.

If a DoS attack originates from multiple source addresses, it is known as a distributed denial-of-service (DDoS) attack. Typically, the source address of a DoS attack is spoofed. The source addresses in a DDoS attack might be spoofed or the actual addresses of hosts that the attacker has previously compromised and which he or she is now using as "zombie agents" from which to launch the attack.

The security device can defend itself and the resources it protects from DoS and DDoS attacks. The following sections describe the various defense options available:

- "Firewall DoS Attacks" on page 28
  - "Session Table Flood" on page 28
  - "SYN-ACK-ACK Proxy Flood" on page 32
- "Network DoS Attacks" on page 34
  - "SYN Flood" on page 34
  - "SYN Cookie" on page 44
  - "ICMP Flood" on page 46
  - "UDP Flood" on page 47
  - "Land Attack" on page 48
- "OS-Specific DoS Attacks" on page 49
  - "Ping of Death" on page 49
  - "Teardrop Attack" on page 50
  - "WinNuke" on page 51

Firewall DoS Attacks

If an attacker discovers the presence of the Juniper Networks firewall, he or she might launch a denial-of-service (DoS) attack against it instead of the network behind it. A successful DoS attack against a firewall amounts to a successful DoS attack against the protected network in that it thwarts attempts of legitimate traffic to traverse the firewall. This section explains two methods that an attacker might use to fill up the session table of a Juniper Networks security device and thereby produce a DoS: Session Table Flood and SYN-ACK-ACK Proxy Flood.

Session Table Flood

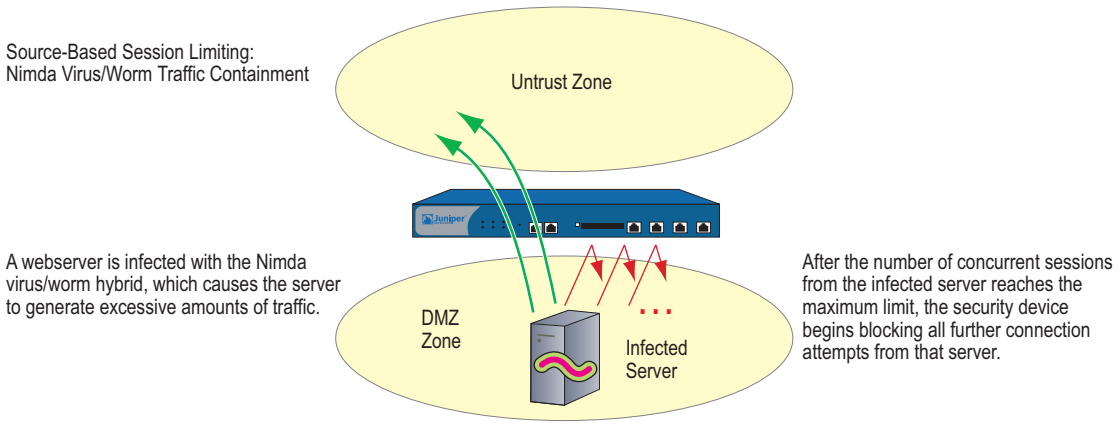
A successful DoS attack overwhelms its victim with such a massive barrage of false simulated traffic that it becomes unable to process legitimate connection requests. DoS attacks can take many forms—SYN flood, SYN-ACK-ACK flood, UDP flood, ICMP flood, and so on—but they all seek the same objective: to fill up their victim's session table. When the session table is full, that host cannot create any new sessions and begins rejecting new connection requests. The following SCREEN options help mitigate such attacks:

- Source-Based and Destination-Based Session Limits
- Aggressive Aging

Source-Based and Destination-Based Session Limits

In addition to limiting the number of concurrent sessions from the same source IP address, you can also limit the number of concurrent sessions to the same destination IP address. One benefit of setting a source-based session limit is that it can stem an attack such as the Nimda virus (which is actually both a virus and a worm) that infects a server and then begins generating massive amounts of traffic from that server. Because all the virus-generated traffic originates from the same IP address, a source-based session limit ensures that the firewall can curb such excessive amounts of traffic.

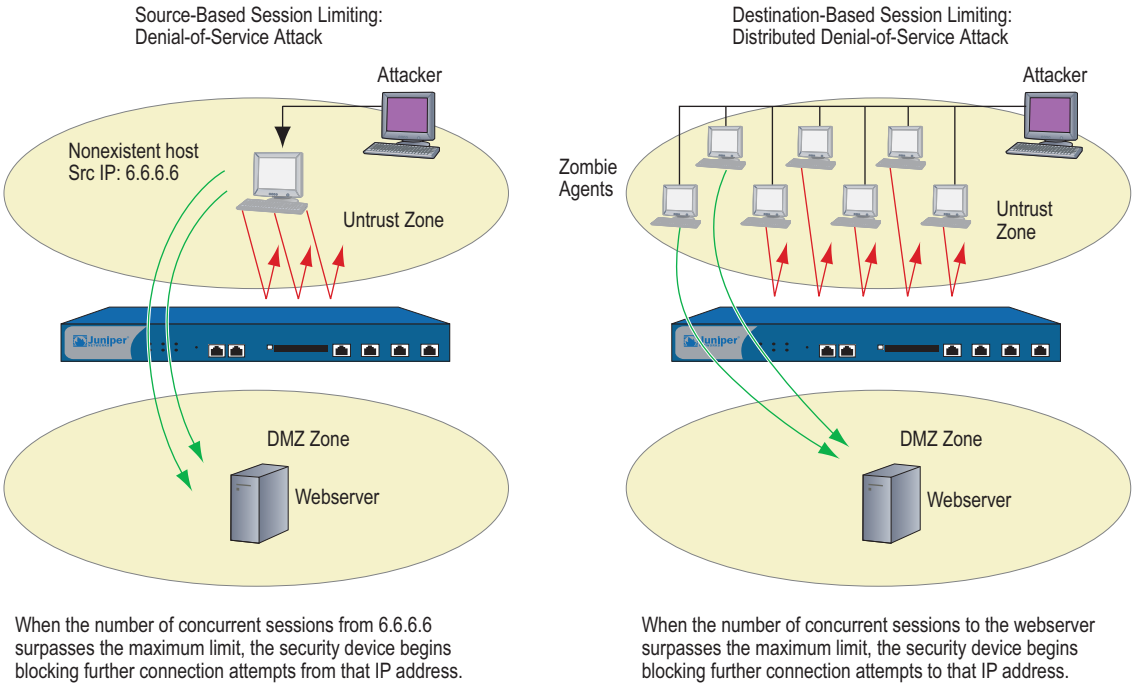
Figure 13: Limiting Sessions Based on Source IP Address



Another benefit of source-based session limiting is that it can mitigate attempts to fill up the ScreenOS session table—if all the connection attempts originate from the same source IP address. However, a wily attacker can launch a distributed denial-of-service (DDoS) attack. In a DDoS attack, the malicious traffic can come

from hundreds of hosts, known as *zombie agents*, that are surreptitiously under the control of an attacker. In addition to the SYN, UDP, and ICMP flood detection and prevention SCREEN options, setting a destination-based session limit can ensure that the security device allows only an acceptable number of concurrent connection requests—no matter what the source—to reach any one host.

Figure 14: Distributed DOS Attack



Determining what constitutes an acceptable number of connection requests requires a period of observation and analysis to establish a baseline for typical traffic flows. You also need to consider the maximum number of concurrent sessions required to fill up the session table of the particular Juniper Networks platform you are using. To see the maximum number of sessions that your session table supports, use the CLI command **get session**, and then look at the first line in the output, which lists the number of current (allocated) sessions, the maximum number of sessions, and the number of failed session allocations:

```
alloc 420/max 128000, alloc failed 0
```

The default maximum for both source- and destination-based session limits is 128 concurrent sessions, a value that might need adjustment to suit the needs of your network environment and the platform.

Example: Source-Based Session Limiting

In this example, you want to limit the amount of sessions that any one server in the DMZ and Trust zones can initiate. Because the DMZ zone only contains webservers, none of which should initiate traffic, you set the source-session limit at the lowest possible value: 1 session. On the other hand, the Trust zone contains personal computers, servers, printers, and so on, many of which do initiate traffic. For the Trust zone, you set the source-session limit maximum to 80 concurrent sessions.

WebUI

Screening > Screen (Zone: DMZ): Enter the following, then click **OK**:

Source IP Based Session Limit: (select)
Threshold: 1 Sessions

Screening > Screen (Zone: Trust): Enter the following, then click **OK**:

Source IP Based Session Limit: (select)
Threshold: 80 Sessions

CLI

```
set zone dmz screen limit-session source-ip-based 1
set zone dmz screen limit-session source-ip-based
set zone trust screen limit-session source-ip-based 80
set zone trust screen limit-session source-ip-based
save
```

Example: Destination-Based Session Limiting

In this example, you want to limit the amount of traffic to a webserver at 1.2.2.5. The server is in the DMZ zone. After observing the traffic flow from the Untrust zone to this server for a month, you have determined that the average number of concurrent sessions it receives is 2000. Based on this information, you decide to set the new session limit at 4000 concurrent sessions. Although your observations show that traffic spikes sometimes exceed that limit, you opt for firewall security over occasional server inaccessibility.

WebUI

Screening > Screen (Zone: Untrust): Enter the following, then click **OK**:

Destination IP Based Session Limit: (select)
Threshold: 4000 Sessions

CLI

```
set zone untrust screen limit-session destination-ip-based 4000
set zone untrust screen limit-session destination-ip-based
save
```

Aggressive Aging

By default, an initial TCP session 3-way handshake takes 20 seconds to time out (that is, to expire because of inactivity). After a TCP session has been established, the timeout value changes to 30 minutes. For HTTP and UDP sessions, the session timeouts are 5 minutes and 1 minute, respectively. The session timeout counter begins when a session starts and is refreshed every 10 seconds if the session is active. If a session becomes idle for more than 10 seconds, the timeout counter begins to decrement.

On certain hardware platforms, ScreenOS provides a mechanism for accelerating the timeout process when the number of sessions in the session table surpasses a specified high-watermark threshold.

NOTE: This feature is not available on the ISG series or NetScreen-5000 series systems.

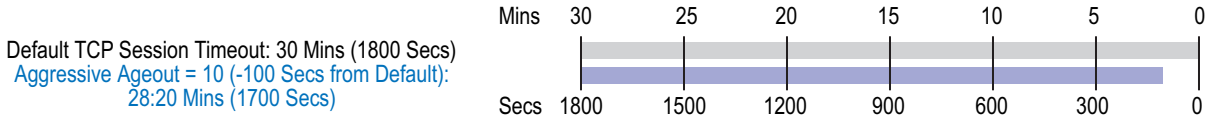
When the number of sessions dips below a specified low-watermark threshold, the timeout process returns to normal. During the period when the aggressive aging out process is in effect, a security device ages out the oldest sessions first, using the aging out rate that you specify. These aged-out sessions are tagged as invalid and are removed in the next "garbage sweep," which occurs every 2 seconds.

The aggressive ageout option shortens default session timeouts by the amount you enter. When you set and enable the aggressive ageout option, the normal session timeout value displayed in the configuration remains unchanged—1800 seconds for TCP, 300 seconds for HTTP, and 60 seconds for UDP sessions. However, when the aggressive ageout period is in effect, these sessions time out earlier—by the amount you specify for early ageout—instead of counting down all the way to zero.

The aggressive ageout value can be between 2 and 10 units, where each unit represents a 10-second interval (that is, the aggressive ageout setting can be between 20 and 100 seconds). The default setting is 2 units, or 20 seconds. If you define the aggressive ageout setting at 100 seconds, for example, you shorten the TCP and HTTP session timeouts as follows:

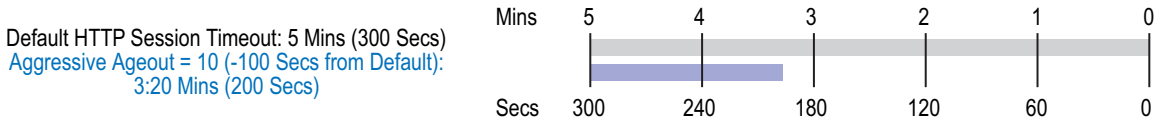
- TCP: The session timeout value shortens from 1800 seconds (30 minutes) to 1700 seconds (28:20 minutes) during the time when the aggressive aging process is in effect. During that period, the security device automatically deletes all TCP sessions whose timeout value has passed 1700 seconds, beginning with the oldest sessions first.

Figure 15: TCP Session Timeout



- HTTP: The session timeout value shortens from 300 seconds (5 minutes) to 200 seconds (3:20 minutes) during the time when the aggressive aging process is in effect. During that period, the security device automatically deletes all HTTP sessions whose timeout value has passed 200 seconds, beginning with the oldest sessions first.

Figure 16: HTTP Session Timeout

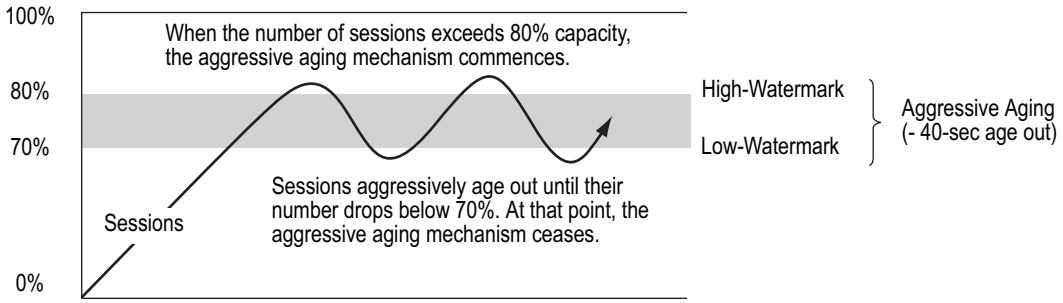


- UDP: Because the default UDP session timeout is 60 seconds, defining an early ageout setting at 100 seconds causes all UDP sessions to ageout and be marked for deletion in the next garbage sweep.

Example: Aggressively Aging Out Sessions

In this example, you set the aggressive aging out process to commence when traffic exceeds a high-watermark of 80 percent and cease when it retreats below a low-watermark of 70 percent. You specify 40 seconds for the aggressive age-out interval. When the session table is more than 80 percent full (the high-mark threshold), the security device decreases the timeout for all sessions by 40 seconds and begins aggressively aging out the oldest sessions until the number of sessions in the table is under 70 percent (the low-mark threshold).

Figure 17: Aging Out Sessions Aggressively



WebUI

NOTE: You must use the CLI to configure the aggressive age-out settings.

CLI

```
set flow aging low-watermark 70
set flow aging high-watermark 80
set flow aging early-ageout 4
save
```

SYN-ACK-ACK Proxy Flood

When an authentication user initiates a Telnet or FTP connection, the user sends a SYN segment to the Telnet or FTP server. The security device intercepts the SYN segment, creates an entry in its session table, and proxies a SYN-ACK segment to the user. The user then replies with an ACK segment. At that point, the initial three-way handshake is complete. The device sends a login prompt to the user. If the user, with malicious intent, does not log in, but instead continues initiating SYN-ACK-ACK sessions, the ScreenOS session table can fill up to the point where the device begins rejecting legitimate connection requests.

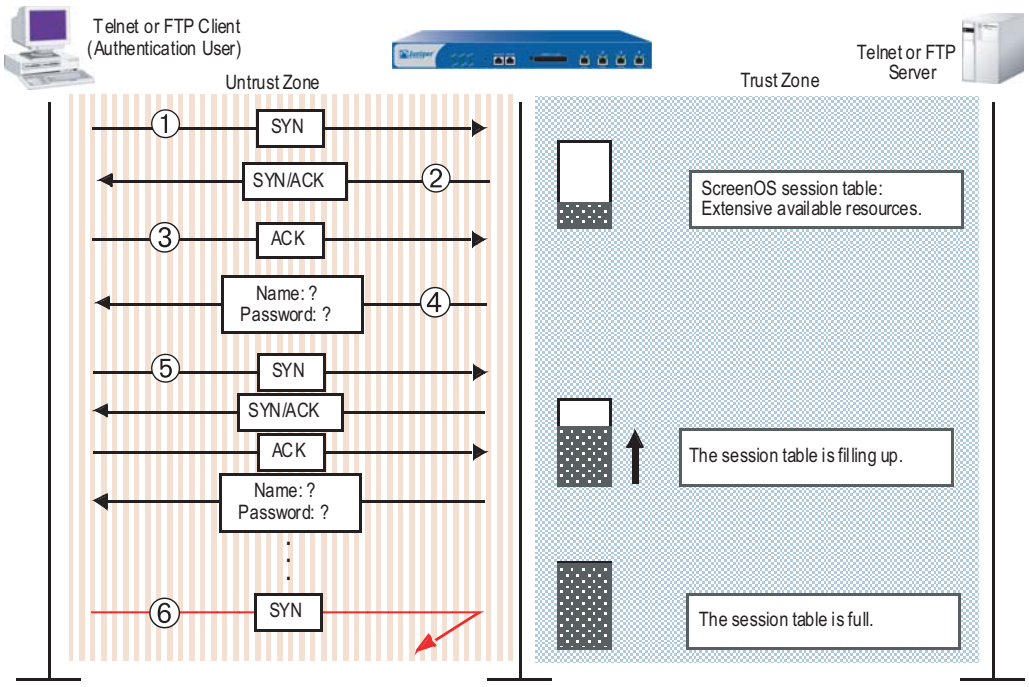
Refer to Figure 18 for a step-by-step process:

1. The client sends a SYN segment to the server.
2. The security device proxies a SYN/ACK segment.
3. The client responds with an ACK segment.
4. The security device prompts the client (auth user) to log in.

5. The client ignores the login prompt and keeps repeating steps 1—4 until the session table is full.

6. Because the session table is full, the security device must reject all further connection requests.

Figure 18: SYN-ACK-ACK Proxy Flood



To prevent such an attack, you can enable the SYN-ACK-ACK proxy protection SCREEN option. After the number of connections from the same IP address reaches the SYN-ACK-ACK proxy threshold, the security device rejects further connection requests from that IP address. By default, the threshold is 512 connections from any single IP address. You can change this threshold (to any number between 1 and 250,000) to better suit the requirements of your network environment.

To enable protection against a SYN-ACK-ACK proxy flood, do either of the following, where the specified zone is that in which the attack originates:

WebUI

Screening > Screen (Zone: select a zone name): Enter the following, then click **Apply**:

SYN-ACK-ACK Proxy Protection: (select)
Threshold: (enter a value to trigger SYN-ACK-ACK proxy flood protection)

NOTE: The value unit is connections per source address. The default value is 512 connections from any single address.

CLI

set zone *zone* screen syn-ack-ack-proxy threshold *number*
set zone *zone* screen syn-ack-ack-proxy

Network DoS Attacks

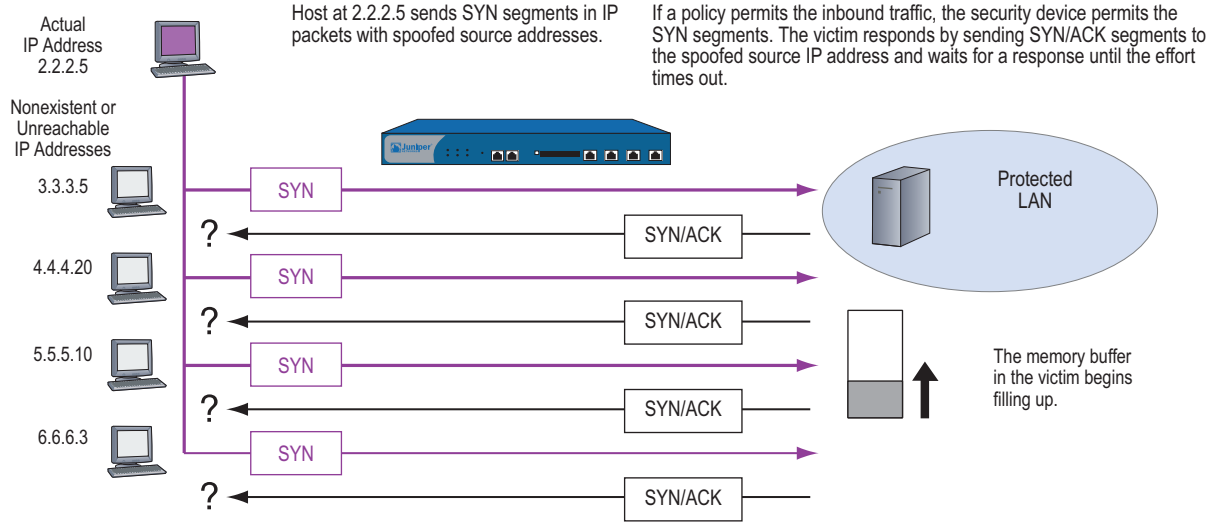
A denial-of-service (DoS) attack directed against one or more network resources floods the target with an overwhelming number of SYN, ICMP, or UDP packets, or with an overwhelming number of SYN fragments. Depending on the attacker's purpose and the extent and success of previous intelligence gathering efforts, the attacker might single out a specific host, such as a router or server; or he or she might aim at random hosts across the targeted network. Either approach has the potential of upsetting service to a single host or to the entire network, depending on how critical the role of the victim is to the rest of the network.

SYN Flood

A SYN flood occurs when a host becomes so overwhelmed by SYN segments initiating incomplete connection requests that it can no longer process legitimate connection requests.

Two hosts establish a TCP connection with a triple exchange of packets known as a *three-way handshake*: A sends a SYN segment to B; B responds with a SYN/ACK segment; and A responds with an ACK segment. A SYN flood attack inundates a site with SYN segments containing forged (spoofed) IP source addresses with nonexistent or unreachable addresses. B responds with SYN/ACK segments to these addresses and then waits for responding ACK segments. Because the SYN/ACK segments are sent to nonexistent or unreachable IP addresses, they never elicit responses and eventually time out.

Figure 19: SYN Flood Attack

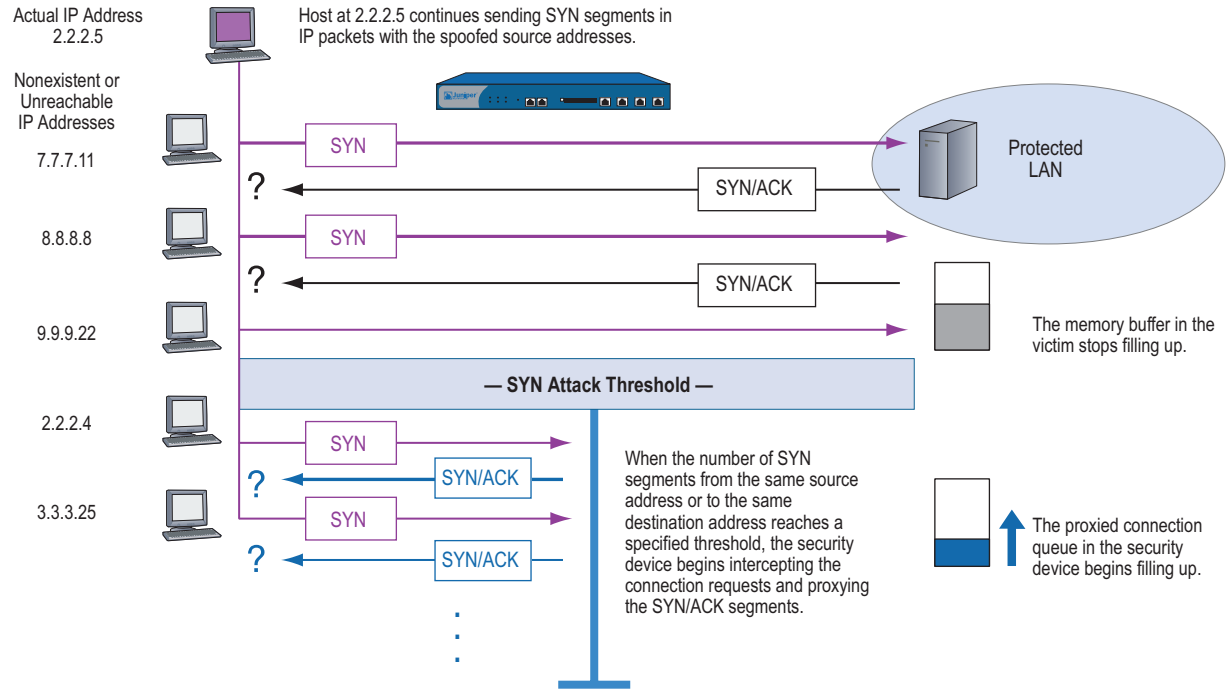


By flooding a host with incomplete TCP connections, the attacker eventually fills the memory buffer of the victim. Once this buffer is full, the host can no longer process new TCP connection requests. The flood might even damage the victim's operating system. Either way, the attack disables the victim and its normal operations.

SYN Flood Protection

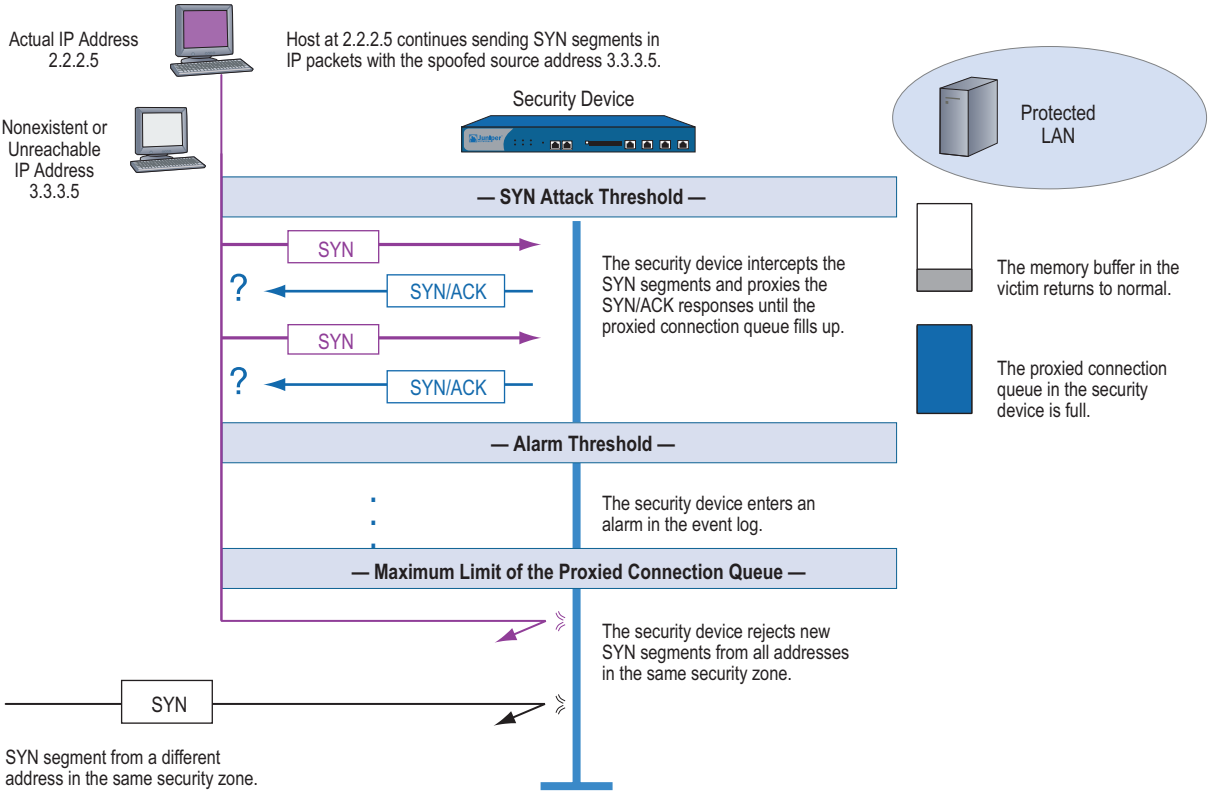
Juniper Networks security devices can impose a limit on the number of SYN segments permitted to pass through the firewall per second. You can base the attack threshold on the destination address and port, the destination address only, or the source address only. When the number of SYN segments per second exceeds one of these thresholds, the security device starts proxying incoming SYN segments, replying with SYN/ACK segments and storing the incomplete connection requests in a connection queue. The incomplete connection requests remain in the queue until the connection is completed or the request times out. In Figure 20, the SYN attack threshold has been passed, and the device has started proxying SYN segments.

Figure 20: Proxying SYN Segments



In Figure 21, the proxied connection queue has completely filled up, and the security device is rejecting new incoming SYN segments. This action shields hosts on the protected network from the bombardment of incomplete three-way handshakes.

Figure 21: Rejecting New SYN Segments



The security device starts receiving new SYN packets when the proxy queue drops below the maximum limit.

NOTE: The procedure of proxying incomplete SYN connections above a set threshold pertains only to traffic permitted by existing policies. Any traffic for which a policy does not exist is automatically dropped.

To enable the SYN flood protection SCREEN option and define its parameters, do either of the following, where the specified zone is that in which a flood might originate:

WebUI

Screening > Screen (Zone: select a zone name): Enter the following, then click **Apply**:

SYN Flood Protection: (select to enable)

Threshold: (enter the number of SYN packets—that is, TCP segments with the SYN flag set—per second required to activate the SYN proxying mechanism)

Alarm Threshold: (enter the number of proxied TCP connection requests required to write an alarm in the event log)

Source Threshold: (enter the number SYN packets per second from a single IP address required for the security device to begin rejecting new connection requests from that source)

Destination Threshold: (enter the number SYN packets per second to a single IP address required for the security device to begin rejecting new connection requests to that destination)

Timeout Value: (enter the length of time in seconds that the security device holds an incomplete TCP connection attempt in the proxied connection queue)

Queue Size: (enter the number of proxied TCP connection requests held in the proxied connection queue before the security device starts rejecting new connection requests)

NOTE: For more details about each of these parameters, see the descriptions in the following CLI section.

CLI

To enable SYN flood protection:

set zone *zone* **screen syn-flood**

You can set the following parameters for proxying uncompleted TCP connection requests:

- **Attack Threshold:** The number of SYN segments (that is, TCP segments with the SYN flag set) to the same destination address and port number per second required to activate the SYN proxying mechanism. Although you can set the threshold at any number, you need to know the normal traffic patterns at your site to set an appropriate threshold for it. For example, if it is an e-business site that normally gets 20,000 SYN segments per second, you might want to set the threshold at 30,000/second. If a smaller site normally gets 20 SYN segments/second, you might consider setting the threshold at 40.

  set zone *zone* **screen syn-flood attack-threshold** *number*

- **Alarm Threshold:** The number of proxied, half-complete TCP connection requests per second after which the security device enters an alarm in the event log. The value you set for an alarm threshold triggers an alarm when the number of proxied, half-completed connection requests to the same destination address and port number per second exceeds that value. For example, if you set the SYN attack threshold at 2000 SYN segments per second and the alarm at 1000, then a total of 3001 SYN segments to the same destination address and port number per second is required to trigger an alarm entry in the log. More precisely:

  1. The firewall passes the first 2000 SYN segments per second that meet policy requirements.

  2. The firewall proxies the next 1000 SYN segments in the same second.

  3. The 1001st proxied connection request (or 3001st connection request in that second) triggers the alarm.

  set zone *zone* **screen syn-flood alarm-threshold** *number*

  For each SYN segment to the same destination address and port number in excess of the alarm threshold, the attack detection module generates a message. At the end of the second, the logging module compresses all similar messages into a single log entry that indicates how many SYN segments to the same destination address and port number arrived after exceeding the alarm threshold. If the attack persists beyond the first second, the event log enters an alarm every second until the attack stops.

- **Source Threshold:** This option allows you to specify the number of SYN segments received per second from a single source IP address—regardless of the destination IP address and port number—before the security device begins dropping connection requests from that source.

  set zone *zone* **screen syn-flood source-threshold** *number*

  Tracking a SYN flood by source address uses different detection parameters from tracking a SYN flood by destination address and destination port number. When you set a SYN attack threshold and a source threshold, you put both the basic SYN flood protection mechanism and the source-based SYN flood tracking mechanism in effect.

- **Destination Threshold:** This option allows you to specify the number of SYN segments received per second for a single destination IP address before the security device begins dropping connection requests to that destination. If a protected host runs multiple services, you might want to set a threshold based on destination IP address only—regardless of the destination port number.

  set zone *zone* **screen syn-flood destination-threshold** *number*

  When you set a SYN attack threshold and a destination threshold, you put both the basic SYN flood protection mechanism and the destination-based SYN flood tracking mechanism in effect.

Tracking a SYN flood by destination address uses different detection parameters from tracking a SYN flood by destination address and destination port number. Consider the following case where the security device has policies permitting FTP requests (port 21) and HTTP requests (port 80) to the same server. If the SYN flood attack threshold is 1000 packets per second (pps) and an attacker sends 999 FTP packets and 999 HTTP packets per second, neither set of packets (where a set is defined as having the same destination address and port number) activates the SYN proxying mechanism. The basic SYN flood attack mechanism tracks destination address and port number, and neither set exceeds the attack threshold of 1000 pps. However, if the destination threshold is 1000 pps, the device treats both FTP and HTTP packets with the same destination address as members of a single set and rejects the 1001st packet—FTP or HTTP—to that destination.

- **Timeout:** The maximum length of time before a half-completed connection is dropped from the queue. The default is 20 seconds, and you can set the timeout from 0–50 seconds. You might try decreasing the timeout value to a shorter length until you begin to see any dropped connections during normal traffic conditions. Twenty seconds is a very conservative timeout for a three-way handshake ACK response.

  set zone *zone* **screen syn-flood timeout** *number*

- **Queue size:** The number of proxied connection requests held in the proxied connection queue before the security device starts rejecting new connection requests. The longer the queue size, the longer the device needs to scan the queue to match a valid ACK response to a proxied connection request. This can slightly slow the initial connection establishment; however, because the time to begin data transfer is normally far greater than any minor delays in initial connection setup, users would not see a noticeable difference.

  set zone *zone* **screen syn-flood queue-size** *number*

- **Drop Unknown MAC:** When a security device detects a SYN attack, it proxies all TCP connection requests. However, a device in Transparent mode cannot proxy a TCP connection request if the destination MAC address is not in its MAC learning table. By default, a device in Transparent mode that has detected a SYN attack passes SYN packets containing unknown MAC addresses. You can use this option to instruct the device to drop SYN packets containing unknown destination MAC addresses instead of letting them pass.

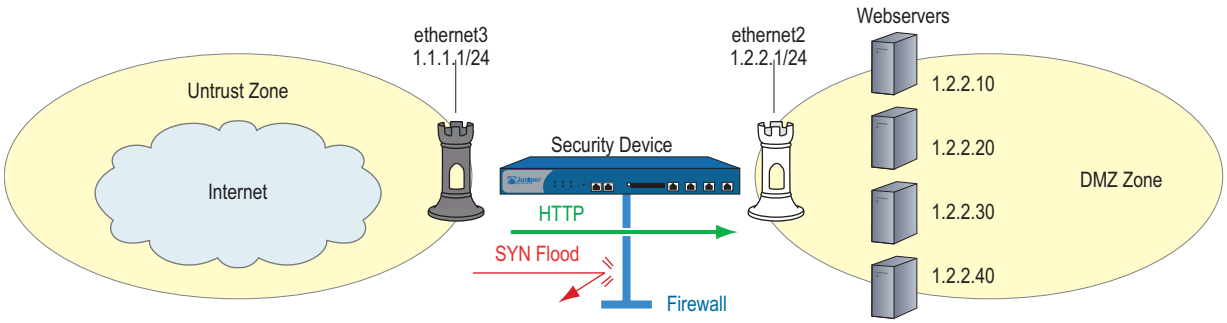
  set zone *zone* **screen syn-flood drop-unknown-mac**

Example: SYN Flood Protection

In this example, you protect four webservers in the DMZ zone from SYN flood attacks originating in the Untrust zone by enabling the SYN flood protection SCREEN option for the Untrust zone.

NOTE: We recommend that you augment the SYN flood protection that the security device provides with device-level SYN flood protection on each of the webservers. In this example, the webservers are running UNIX, which also provides some SYN flood defenses, such as adjusting the length of the connection request queue and changing the timeout period for incomplete connection requests.

Figure 22: Device-Level SYN Flood Protection



To configure the SYN flood protection parameters with appropriate values for your network, you must first establish a baseline of typical traffic flows. For one week, you run a sniffer on ethernet3—the interface bound to the Untrust zone—to monitor the number of new TCP connection requests arriving every second for the four webservers in the DMZ. Your analysis of the data accumulated from one week of monitoring produces the following statistics:

- Average number of new connection requests per server: 250/second
- Average peak number of new connection requests per server: 500/second

NOTE: A sniffer is a network-analyzing device that captures packets on the network segment to which you attach it. Most sniffers allow you to define filters to collect only the type of traffic that interests you. Later, you can view and evaluate the accumulated information. In this example, you want the sniffer to collect all TCP packets with the SYN flag set arriving at ethernet3 and destined for one of the four webservers in the DMZ.

You might want to continue running the sniffer at regular intervals to see if there are traffic patterns based on the time of day, days of the week, the time of month, or the season of the year. For example, in some organizations, traffic might increase dramatically during a critical event. Significant changes probably warrant adjusting the various thresholds.

Based on this information, you set the following SYN flood protection parameters for the Untrust zone, as shown in Table 2.

Table 2: SYN Flood Protection Parameters

Parameter	Value	Reason for Each Value
Attack Threshold	625 packets per second (pps)	This is 25% higher than the average peak number of new connection requests per second per server, which is unusual for this network environment. When the number of SYN packets per second for any one of the four webservers exceeds this number, the device begins proxying new connection requests to that server. (In other words, beginning with the 626th SYN packet to the same destination address and port number in one second, the device begins proxying connection requests to that address and port number.)
Alarm Threshold	250 pps	250 pps is 1/4 of the queue size (1000 proxied, half-completed connection requests[1]). When the device proxies 251 new connection requests in one second, it makes an alarm entry in the event log. By setting the alarm threshold somewhat higher than the attack threshold, you can avoid alarm entries for traffic spikes that only slightly exceed the attack threshold.
Source Threshold	25 pps	When you set a source threshold, the device tracks the source IP address of SYN packets, regardless of the destination address and port number. (Note that this source-based tracking is separate from the tracking of SYN packets based on destination address and destination port number that constitutes the basic SYN flood protection mechanism.) In the one week of monitoring activity, you observed that no more than 1/25 of new connection requests for all servers came from any one source within a one-second interval. Therefore, connection requests exceeding this threshold are unusual and provide sufficient cause for the device to execute its proxying mechanism. (25 pps is 1/25 of the attack threshold, which is 625 pps.) If the device tracks 25 SYN packets from the same source IP address, beginning with the 26th packet, it rejects all further SYN packets from that source for the remainder of that second and the next second as well.
Destination Threshold	0 pps	When you set a destination threshold, the device runs a separate tracking of only the destination IP address, regardless of the destination port number. Because the four webservers only receive HTTP traffic (destination port 80)—no traffic to any other destination port number reaches them—setting a separate destination threshold offers no additional advantage.
Timeout	20 seconds	Because the queue size is relatively short (1000 proxied connection requests), the default value of 20 seconds is a reasonable length of time to hold incomplete connection requests in the queue for this configuration.
Queue Size	1000 proxied, half-completed connections	1000 proxied, half-completed connection requests is twice the average peak number of new connection requests (500 pps). The device proxies up to 1000 requests per second before dropping new requests. Proxying twice the average peak number of new connection requests provides a conservative buffer for legitimate connection requests to get through.

1. Half-completed connection requests are incomplete three-way handshakes. A three-way handshake is the initial phase of a TCP connection. It consists of a TCP segment with the SYN flag set, a response with the SYN and ACK flags set, and a response to that with the ACK flag set.

WebUI

1. **Interfaces**
   Network > Interfaces > Edit (for ethernet2): Enter the following, then click **OK**:

   Zone Name: DMZ
   Static IP: (select this option when present)
   IP Address/Netmask: 1.2.2.1/24

   Network > Interfaces > Edit (for ethernet3): Enter the following, then click **OK**:

   Zone Name: Untrust
   Static IP: (select this option when present)
   IP Address/Netmask: 1.1.1.1/24

2. **Addresses**
   Objects > Addresses > List > New: Enter the following, then click **OK**:

   Address Name: ws1
   IP Address/Domain Name:
   IP/Netmask: (select), 1.2.2.10/32
   Zone: DMZ

   Objects > Addresses > List > New: Enter the following, then click **OK**:

   Address Name: ws2
   IP Address/Domain Name:
   IP/Netmask: (select), 1.2.2.20/32
   Zone: DMZ

   Objects > Addresses > List > New: Enter the following, then click **OK**:

   Address Name: ws3
   IP Address/Domain Name:
   IP/Netmask: (select), 1.2.2.30/32
   Zone: DMZ

   Objects > Addresses > List > New: Enter the following, then click **OK**:

   Address Name: ws4
   IP Address/Domain Name:
   IP/Netmask: (select), 1.2.2.40/32
   Zone: DMZ

   Objects > Addresses > Groups > (for Zone: DMZ) New: Enter the following group name, move the following addresses, then click **OK**:

   Group Name: web_servers

   Select **ws1** and use the < < button to move the address from the Available Members column to the Group Members column.

   Select **ws2** and use the < < button to move the address from the Available Members column to the Group Members column.

Select **ws3** and use the < < button to move the address from the Available Members column to the Group Members column.

Select **ws4** and use the < < button to move the address from the Available Members column to the Group Members column.

3. Policy

Policies > (From: Untrust, To: DMZ) New: Enter the following, then click **OK**:

Source Address:
    Address Book Entry: (select), Any
Destination Address:
    Address Book Entry: (select), web_servers
Service: HTTP
Action: Permit

4. SCREEN

Screening > Screen (Zone: Untrust): Enter the following, then click **Apply**:

SYN Flood Protection: (select)
    Threshold: 625
    Alarm Threshold: 250
    Source Threshold: 25
    Destination Threshold: 0
    Timeout Value: 20
    Queue Size: 1000

NOTE: Because 20 seconds is the default setting, you do not have to set the timeout to 20 seconds unless you have previously set it to another value.

CLI

1. Interfaces

```
set interface ethernet2 zone dmz
set interface ethernet2 ip 1.2.2.1/24
set interface ethernet3 zone untrust
set interface ethernet3 ip 1.1.1.1/24
```

2. Addresses

```
set address dmz ws1 1.2.2.10/32
set address dmz ws2 1.2.2.20/32
set address dmz ws3 1.2.2.30/32
set address dmz ws4 1.2.2.40/32
set group address dmz web_servers add ws1
set group address dmz web_servers add ws2
set group address dmz web_servers add ws3
set group address dmz web_servers add ws4
```

3. Policy

```
set policy from untrust to dmz any web_servers HTTP permit
```

4. **SCREEN**

```
set zone untrust screen syn-flood attack-threshold 625
set zone untrust screen syn-flood alarm-threshold 250
set zone untrust screen syn-flood source-threshold 25
set zone untrust screen syn-flood timeout 20
set zone untrust screen syn-flood queue-size 1000
set zone untrust screen syn-flood
save
```

NOTE: Because 20 seconds is the default setting, you do not have to set the timeout to 20 seconds unless you have previously set it to another value.

SYN Cookie

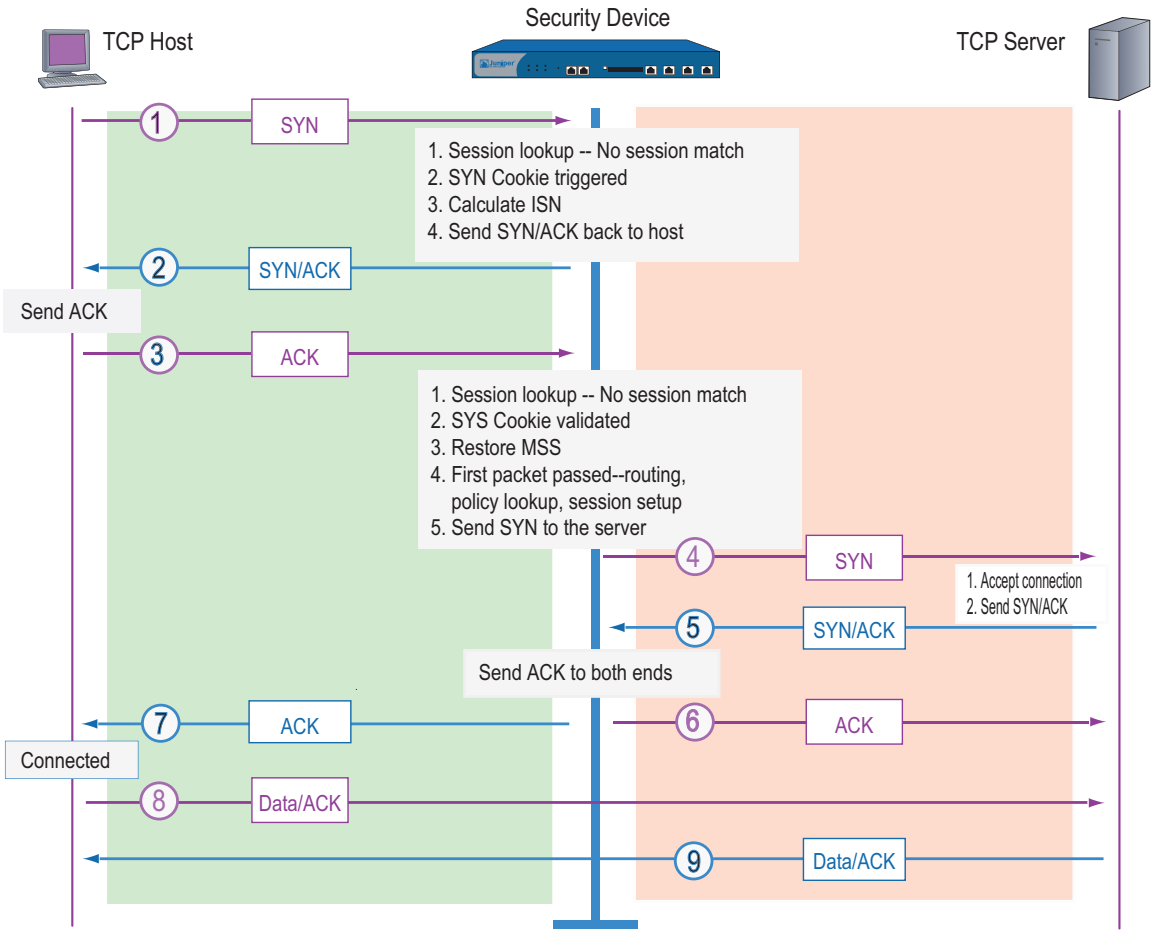
SYN Cookie is a stateless SYN proxy mechanism you can use in conjunction with the defenses against a SYN flood attack described in “SYN Flood” on page 34. Like traditional SYN proxying, SYN Cookie is activated when the SYN flood attack threshold is exceeded, but because SYN Cookie is stateless, it does not set up a session or do policy and route lookups upon receipt of a SYN segment, and maintains no connection request queues. This dramatically reduces CPU and memory usage and is the primary advantage of using SYN Cookie over the traditional SYN proxying mechanism.

When SYN Cookie is enabled on the security device and becomes the TCP-negotiating proxy for the destination server, it replies to each incoming SYN segment with a SYN/ACK containing an encrypted cookie as its Initial Sequence Number (ISN). The cookie is a MD5 hash of the original source address and port number, destination address and port number, and ISN from the original SYN packet. After sending the cookie, the device drops the original SYN packet and deletes the calculated cookie from memory. If there is no response to the packet containing the cookie, the attack is noted as an active SYN attack and is effectively stopped.

If the initiating host responds with a TCP packet containing the cookie + 1 in the TCP ACK field, the device extracts the cookie, subtracts 1 from the value, and recomputes the cookie to validate that it is a legitimate ACK. If it is legitimate, the device starts the TCP proxy process by setting up a session and sending a SYN to the server containing the source information from the original SYN. When the device receives a SYN/ACK from the server, it sends ACKs to the sever and to the initiation host. At this point the connection is established and the host and server are able to communicate directly.

Figure 23 shows how a connection is established between an initiating host and a server when SYN Cookie is active on the security device.

Figure 23: Establishing a Connection with SYN Cookie Active



To enable SYN Cookie, set a SYN flood attack threshold (as described in "SYN Flood" on page 34), and do one of the following:

WebUI

Configuration > Advanced > Flow: Enter the following and click **Apply**:

TCP SYN-proxy SYN-cookie: (select)

CLI

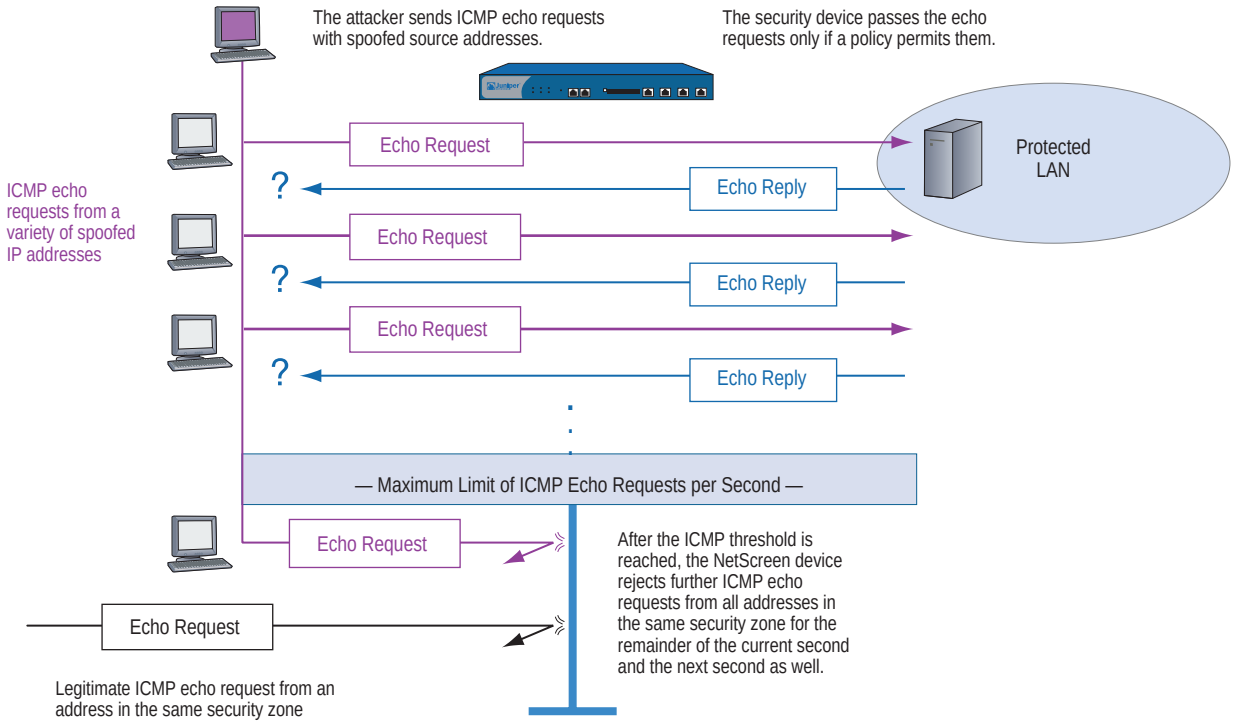
set flow syn-proxy syn-cookie

ICMP Flood

An ICMP flood typically occurs when ICMP echo requests overload its victim with so many requests that it expends all its resources responding until it can no longer process valid network traffic. When enabling the ICMP flood protection feature, you can set a threshold that once exceeded invokes the ICMP flood attack protection feature. (The default threshold value is 1000 packets per second.) If the threshold is exceeded, the security device ignores further ICMP echo requests for the remainder of that second plus the next second as well.

NOTE: An ICMP flood can consist of any type of ICMP message. Therefore, a Juniper Networks security device monitors all ICMP message types, not just echo requests.

Figure 24: ICMP Flooding



To enable ICMP flood protection, do either of the following, where the specified zone is that in which a flood might originate:

WebUI

Screening > Screen (Zone: select a zone name): Enter the following, then click **Apply**:

ICMP Flood Protection: (select)
Threshold: (enter a value to trigger ICMP flood protection)

NOTE: The value unit is ICMP packets per second. The default value is 1000 packets per second.

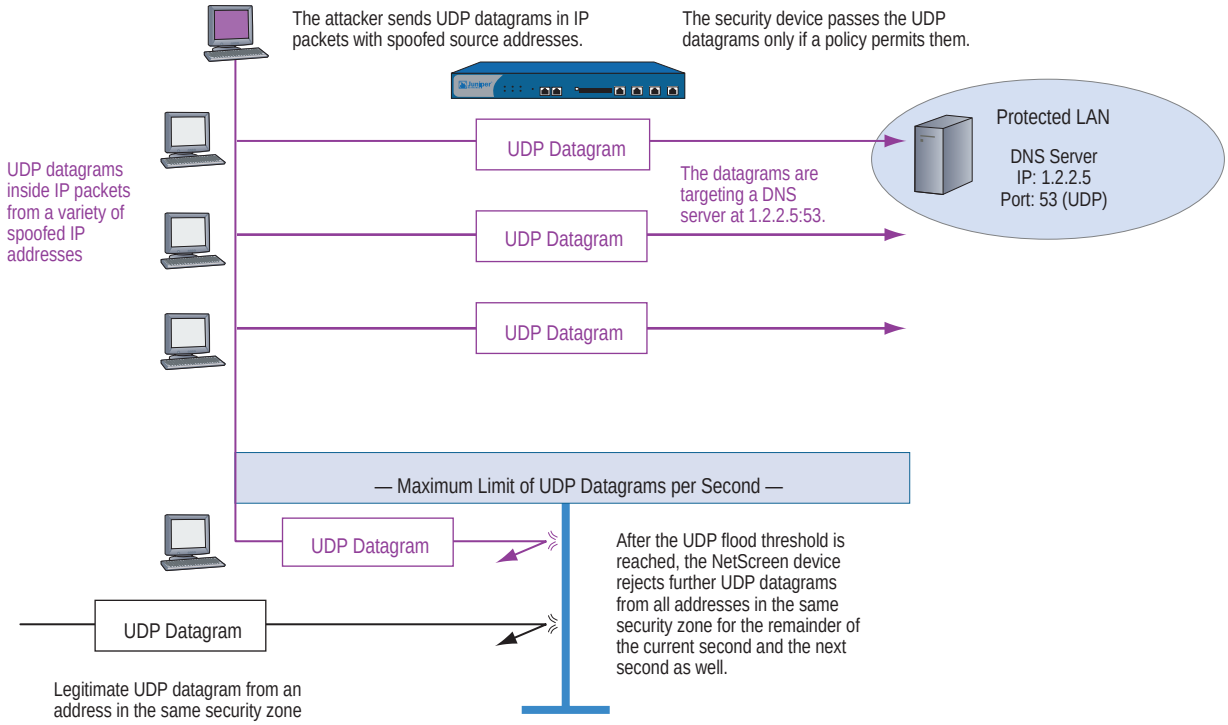
CLI

set zone *zone* screen icmp-flood threshold *number*
set zone *zone* screen icmp-flood

UDP Flood

Similar to the ICMP flood, UDP flooding occurs when an attacker sends IP packets containing UDP datagrams with the purpose of slowing down the victim to the point that it can no longer handle valid connections. After enabling the UDP flood protection feature, you can set a threshold that, once exceeded, invokes the UDP flood attack protection feature. (The default threshold value is 1000 packets per second.) If the number of UDP datagrams from one or more sources to a single destination exceeds this threshold, the security device ignores further UDP datagrams to that destination for the remainder of that second plus the next second as well.

Figure 25: UDP Flooding



To enable UDP flood protection, do either of the following, where the specified zone is that in which a flood might originate:

WebUI

Screening > Screen (Zone: select a zone name): Enter the following, then click **Apply**:

UDP Flood Protection: (select)
Threshold: (enter a value to trigger UDP flood protection)

NOTE: The value unit is UDP packets per second. The default value is 1000 packets per second.

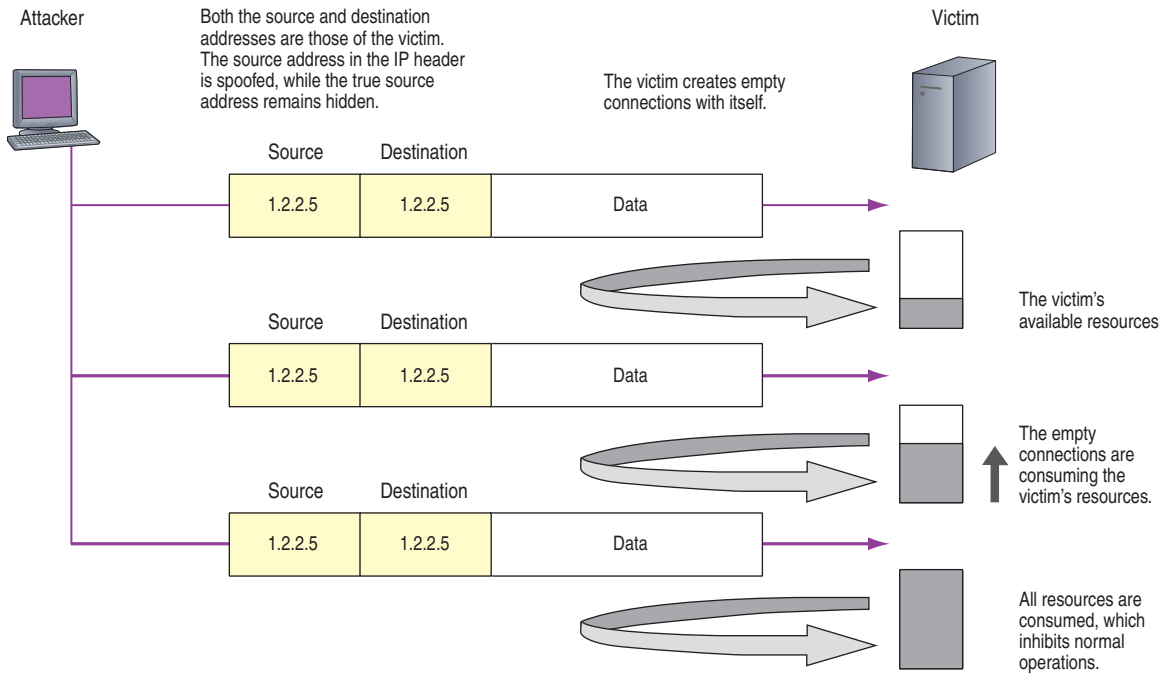
CLI

set zone *zone* screen udp-flood threshold *number*
set zone *zone* screen udp-flood

Land Attack

Combining a SYN attack with IP spoofing, a Land attack occurs when an attacker sends spoofed SYN packets containing the IP address of the victim as both the destination and source IP address. The receiving system responds by sending the SYN-ACK packet to itself, creating an empty connection that lasts until the idle timeout value is reached. Flooding a system with such empty connections can overwhelm the system, causing a denial of service.

Figure 26: Land Attack



When you enable the SCREEN option to block Land attacks, the security device combines elements of the SYN flood defense and IP spoofing protection to detect and block any attempts of this nature.

To enable protection against a Land attack, do either of the following, where the specified zone is that in which the attack originates:

WebUI

Screening > Screen (Zone: select a zone name): Select **Land Attack Protection**, then click **Apply**.

CLI

set zone *zone* screen land

OS-Specific DoS Attacks

If an attacker not only identifies the IP address and responsive port numbers of an active host but also its operating system (OS), instead of resorting to brute-force attacks, he or she can launch more elegant attacks that can produce one- or two-packet "kills." The attacks presented in this section can cripple a system with minimum effort. If your Juniper Networks security device is protecting hosts susceptible to these attacks, you can enable the security device to detect these attacks and block them before they reach their target.

Ping of Death

The maximum allowable IP packet size is 65,535 bytes, including the packet header, which is typically 20 bytes long. An ICMP echo request is an IP packet with a pseudo header, which is 8 bytes long. Therefore, the maximum allowable size of the data area of an ICMP echo request is 65,507 bytes (65,535 - 20 - 8 = 65,507).

However, many ping implementations allow the user to specify a packet size larger than 65,507 bytes. A grossly oversized ICMP packet can trigger a range of adverse system reactions such as denial of service (DoS), crashing, freezing, and rebooting.

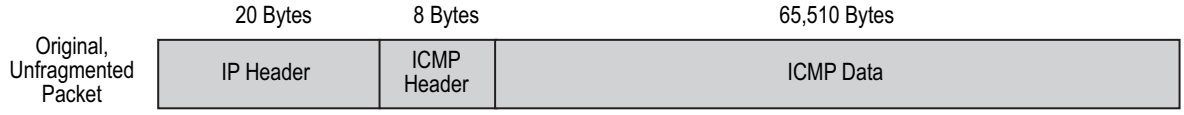
When you enable the Ping of Death SCREEN option, the security device detects and rejects such oversized and irregular packet sizes even when the attacker hides the total packet size by purposefully fragmenting it.

NOTE: For information about IP specifications, see RFC 791, *Internet Protocol*.

For information about ICMP specifications, see RFC 792, *Internet Control Message Protocol*.

For information about Ping of Death, see http://www.insecure.org/sploits/ping-o-death.html.

Figure 27: Ping of Death



The size of this packet is 65,538 bytes. It exceeds the size limit prescribed by RFC 791, Internet Protocol, which is 65,535 bytes. As the packet is transmitted, it becomes broken into numerous fragments. The reassembly process might cause the receiving system to crash.

To enable protection against a Ping of Death attack, do either of the following, where the specified zone is that in which the attack originates:

WebUI

Screening > Screen (Zone: select a zone name): Select **Ping of Death Attack Protection**, then click **Apply**.

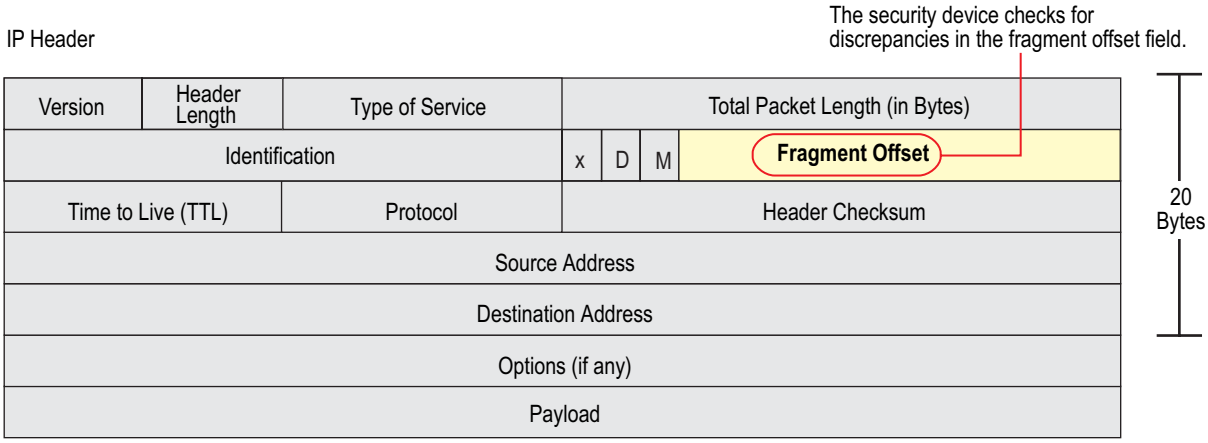
CLI

set zone *zone* screen ping-death

Teardrop Attack

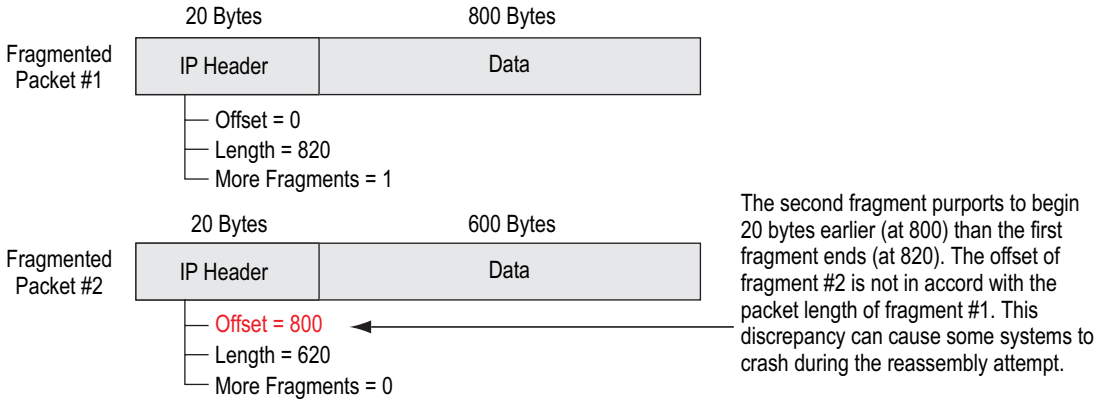
Teardrop attacks exploit the reassembly of fragmented IP packets. In the IP header, one of the fields is the fragment offset field, which indicates the starting position, or offset, of the data contained in a fragmented packet relative to the data of the original unfragmented packet.

Figure 28: Teardrop Attacks



When the sum of the offset and size of one fragmented packet differ from that of the next fragmented packet, the packets overlap, and the server attempting to reassemble the packet can crash, especially if it is running an older operating system that has this vulnerability.

Figure 29: Fragment Discrepancy



After you enable the Teardrop Attack SCREEN option, whenever the device detects this discrepancy in a fragmented packet, it drops it.

To enable protection against a Teardrop attack, do either of the following, where the specified zone is that in which the attack originates:

WebUI

Screening > Screen (Zone: select a zone name): Select **Teardrop Attack Protection**, then click **Apply**.

CLI

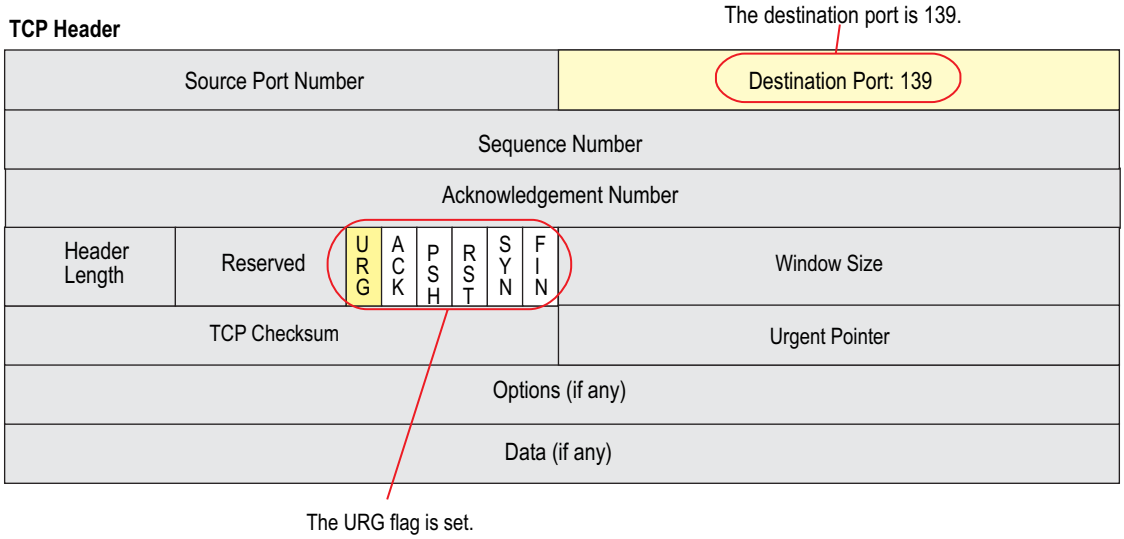
set zone *zone* screen tear-drop

WinNuke

WinNuke is a DoS attack targeting any computer on the Internet running Windows. The attacker sends a TCP segment—usually to NetBIOS port 139 with the urgent (URG) flag set—to a host with an established connection. This introduces a NetBIOS fragment overlap, which causes many machines running Windows to crash. After rebooting the attacked machine, the following message appears, indicating that an attack has occurred:

> An exception 0E has occurred at 0028:[address] in VxD MSTCP(01) + 000041AE. This was called from 0028:[address] in VxD NDIS(01) + 00008660. It may be possible to continue normally.
>
> Press any key to attempt to continue.
>
> Press CTRL+ALT+DEL to restart your computer. You will lose any unsaved information in all applications.
>
> Press any key to continue.

Figure 30: WinNuke Attack Indicators



If you enable the WinNuke attack defense SCREEN option, the security device scans any incoming Microsoft NetBIOS session service (port 139) packets. If the device observes that the URG flag is set in one of those packets, it unsets the URG flag, clears the URG pointer, forwards the modified packet, and makes an entry in the event log noting that it has blocked an attempted WinNuke attack.

To enable protection against a WinNuke attack, do either of the following, where the specified zone is that in which the attack originates:

WebUI

Screening > Screen (Zone: select a zone name): Select **WinNuke Attack Protection**, then click **Apply**.

CLI

set zone *zone* screen winnuke

Chapter 4

Content Monitoring and Filtering

Juniper Networks provides broad protection and control of network activity through ScreenOS features and the pairing of ScreenOS with Websense, SurfControl, Kaspersky Lab, and Trend Micro products.

Juniper Networks provides some content monitoring and filtering capabilities within ScreenOS in its malicious URL protection SCREEN option. Furthermore, through the fragment reassembly feature, a Juniper Networks security device can detect URLs even among fragmented TCP segments and fragmented IP packets.

For antivirus (AV) protection, you have a choice on some security devices to obtain an advanced license key and an AV subscription key and use an internal AV scanning feature. For web filtering, you can configure a device to work with an internal web-filtering engine or with one or more external web-filtering servers.

This chapter describes how to configure the device to perform segment and packet reassembly, monitor HTTP traffic for malicious URLs, and communicate with other devices to perform AV scanning and web filtering. The chapter is organized into the following sections:

- "Fragment Reassembly" on page 54
  - "Malicious URL Protection" on page 54
  - "Application Layer Gateway" on page 55
- "Antivirus Scanning" on page 57
  - "Scanning FTP Traffic" on page 58
  - "Scanning HTTP Traffic" on page 59
  - "Scanning IMAP and POP3 Traffic" on page 61
  - "Scanning SMTP Traffic" on page 63
  - "Updating the AV Pattern File" on page 64
  - "Spyware and Phishing Protection" on page 67

- "Policy-Based AV Scanning" on page 68
- "AV Scanner Global Settings" on page 69
- "AV Scanner Profile Settings" on page 72

- "Anti-Spam Filtering" on page 77
- "Web Filtering" on page 80
  - "Integrated Web Filtering" on page 81
  - "Redirect Web Filtering" on page 89

Fragment Reassembly

Typically, a network forwarding device such as a router or switch does not reassemble fragmented packets that it receives. It is the responsibility of the destination host to reconstruct the fragmented packets when they all arrive. Because the purpose of forwarding devices is the efficient delivery of traffic, queuing fragmented packets, reassembling them, then refragmenting them, and forwarding them is unnecessary and inefficient. However, passing fragmented packets through a firewall is insecure. An attacker can intentionally break up packets to conceal traffic strings that the firewall otherwise would detect and block.

ScreenOS allows you to enable fragment reassembly on a per zone basis. Doing so allows the security device to expand its ability to detect and block malicious URL strings, and to improve its ability to provide an Application Layer Gateway (ALG) to check the data portions of packets.

Malicious URL Protection

In addition to the web-filtering feature, explained later in this chapter (see "Redirect Web Filtering" on page 89), you can define up to 48 malicious URL string patterns per zone, each of which can be up to 64 characters long, for malicious URL protection at the zone level. With the Malicious URL blocking feature enabled, the security device examines the data payload of all HTTP packets. If it locates a URL and detects that the beginning of its string—up to a specified number of characters—matches the pattern you defined, the device blocks that packet from passing the firewall.

A resourceful attacker, realizing that the string is known and might be guarded against, can deliberately fragment the IP packets or TCP segments to make the pattern unrecognizable during a packet-by-packet inspection. For example, if the malicious URL string is **120.3.4.5/level/50/exec**, IP fragmentation might break up the string into the following sections:

- First packet: **120**
- Second packet: **3.4.5/level/50**
- Third packet: **/exec**

Individually, the fragmented strings can pass undetected through the security device, even if you have the string defined as **120.3.4.5/level/50/exec** with a length of 20 characters. The string in the first packet—"120."— matches the first part of the defined pattern, but it is shorter than the required length of 20 matching characters. The strings in the second and third packets do not match the beginning of the defined pattern, and also pass without impedance.

However, if the packets are reassembled, the fragments combine to form a recognizable string that the device can block. Using the Fragment Reassembly feature, the device can buffer fragments in a queue, reassemble them into a complete packet, and then inspect that packet for a malicious URL. Depending on the results of this reassembly process and subsequent inspection, the device performs one of the following actions:

- If the device discovers a malicious URL, it drops the packet and enters the event in the log.
- If the device cannot complete the reassembly process, a time limit is imposed to age out and discard fragments.
- If the device determines that the URL is not malicious but the reassembled packet is too big to forward, the device fragments that packet into multiple packets and forwards them.
- If the device determines that the URL is not malicious and does not need to fragment it, it forwards the packet.

Application Layer Gateway

ScreenOS provides an Application Layer Gateway (ALG) for a number of protocols, such as DNS, FTP, H.323, and HTTP. Of these, fragment reassembly can be an important component in the enforcement of policies involving FTP and HTTP services. The ability of the Juniper Networks firewall to screen packets for protocols such as FTP-Get and FTP-Put requires it to examine not only the packet header but also the data in the payload. For example, there might be two policies, one denying FTP-put from the Untrust to DMZ zones, and another permitting FTP-get from the Untrust to the DMZ zones:

```
set policy from untrust to dmz any any ftp-put deny
set policy from untrust to dmz any any ftp-get permit
```

To distinguish the two types of traffic, the firewall examines the payload. If it reads **RETR** *filename*, the FTP client has sent a request to get (or "retrieve") the specified file from the FTP server, and the security device allows the packet to pass. If the security device finds **STOR** *filename*, the client has sent a request to put (or "store") the specified file on the server, and the device blocks the packet.

To get around this defense, an attacker can deliberately fragment a single FTP-put packet into two packets that contain the following text in their respective payloads: packet 1: **ST**; packet 2: **OR** *filename*. When the security device inspects each packet individually, it does not find the string **STOR** *filename*, and consequently allows them both to pass.

However, if the packets are reassembled, the fragments combine to form a recognizable string upon which the security device can act. Using the Fragment Reassembly feature, the device buffers the FTP fragments in a queue, reassembles them into a complete packet, and then inspects that packet for the complete FTP request. Depending on the results of this reassembly process and subsequent inspection, the device performs one of the following actions:

- If the device discovers an FTP-put request, it drops the packet and enters the event in the log.
- If the device cannot complete the reassembly process, a time limit is imposed to age out and discard fragments.
- If the device discovers an FTP-get request but the reassembled packet is too big to forward, the device fragments that packet into multiple packets and forwards them.
- If the device discovers an FTP-get request and does not need to fragment it, it then forwards the packet.

Example: Blocking Malicious URLs in Packet Fragments

In this example, you define the following three malicious URL strings and enable the malicious URL blocking option:

- Malicious URL #1
  - ID: Perl
  - Pattern: scripts/perl.exe
  - Length: 14
- Malicious URL #2
  - ID: CMF
  - Pattern: cgi-bin/phf
  - Length: 11
- Malicious URL #3
  - ID: DLL
  - Pattern: 210.1.1.5/msadcs.dll
  - Length: 18

The values for “length” indicate the number of characters in the pattern that must be present in a URL—starting from the first character—for a positive match. Note that for #1 and #3, not every character is required.

You then enable fragment reassembly for the detection of the URLs in fragmented HTTP traffic arriving at an Untrust zone interface.

WebUI

Screening > Mal-URL (Zone: Untrust): Enter the following, then click **OK**:

ID: perl
Pattern: /scripts/perl.exe
Length: 14

Screening > Mal-URL (Zone: Untrust): Enter the following, then click **OK**:

ID: cmf
Pattern: cgi-bin/phf
Length: 11

Screening > Mal-URL (Zone: Untrust): Enter the following, then click **OK**:

ID: dll
Pattern: 210.1.1.5/msadcs.dll
Length: 18

Network > Zones > Edit (for Untrust): Select the **TCP/IP Reassembly for ALG** checkbox, then click **OK**.

CLI

```
set zone untrust screen mal-url perl “scripts/perl.exe“ 14
set zone untrust screen mal-url cmf “cgi-bin/phf” 11
set zone untrust screen mal-url dll “210.1.1.5/msadcs.dll” 18
set zone untrust reassembly-for-alg
save
```

Antivirus Scanning

A virus is executable code that infects or attaches itself to other executable code so that it can reproduce itself. Some viruses are malicious and erase files or lock up systems, while other viruses act by infecting files and can overwhelm the target host or network with bogus data.

Select Juniper Networks security devices support an internal antivirus (AV) scan engine (AV scanner) that provides AV scanning for specific Application Layer transactions. You can configure the scanner to examine network traffic that uses the following protocols:

Protocols	See
File Transfer Protocol (FTP)	“Scanning FTP Traffic” on page 58
HyperText Transfer Protocol (HTTP)	“Scanning HTTP Traffic” on page 59
Internet Mail Access Protocol (IMAP)	“Scanning IMAP and POP3 Traffic” on page 61
Post Office Protocol, version 3 (POP3)	“Scanning IMAP and POP3 Traffic” on page 61
Simple Mail Transfer Protocol (SMTP)	“Scanning SMTP Traffic” on page 63

To apply AV protection, you reference the internal scanner in a security policy. When the security device receives traffic to which a policy requiring AV scanning applies, it directs the content it receives to its internal scanner. After verifying that it has received the entire content of an FTP, an HTTP, an IMAP, a POP3, or an SMTP

object, the scanner examines the data for viruses. It does this by referencing a virus pattern file in order to identify virus signatures. When the scanner detects a virus, the device drops the content and sends a message to inform the client that the content is infected. If the scanner does not detect a virus, the device forwards the content to its intended destination.

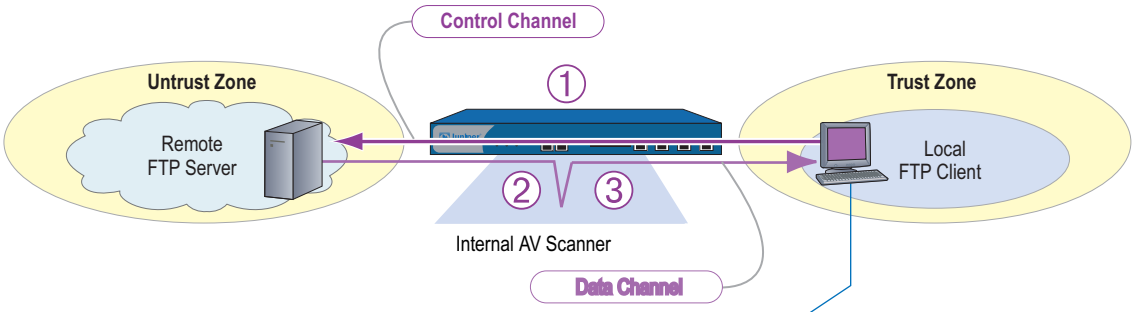
NOTE: To see information on saving an AV pattern file to the Juniper Networks security device and then periodically updating it, see "Updating the AV Pattern File" on page 64.

Scanning FTP Traffic

For File Transfer Protocol (FTP) traffic, the security device monitors the control channel and, when it detects one of the FTP commands for transferring data (RETR, STOR, STOU, APPE, or NLST), it scans the data sent over the data channel. Depending on the results of the scan and how you have configured the **fail-mode** behavior, the device takes one of the following actions:

If the Data	And	The Security Device
is uncontaminated		passes the data to the FTP client through the data channel
contains a virus		drops data from the data channel and sends a virus notification message to the FTP client through the control channel
exceeds the maximum content size	**drop** is set	drops data from the data channel and sends a "file too large" message to the FTP client through the control channel
exceeds the maximum content size	**drop** is unset	passes the unexamined data to the FTP client through the data channel
cannot successfully be scanned	**fail mode** is unset	drops data from the data channel and sends a "scan error" message to the FTP client through the control channel
cannot successfully be scanned	**traffic permit** is set	passes the data to the FTP client through the data channel
exceeds the maximum concurrent messages	**drop** is set	drops data from the data channel and sends an "exceeding maximum message setting" message to the FTP client through the control channel
exceeds the maximum concurrent messages	**drop** is unset	passes the data to the FTP client through the data channel

Figure 31: Antivirus Scanning for FTP Traffic



1. A local FTP client opens an FTP control channel to an FTP server and requests the transfer of some data.
2. The FTP client and server negotiate a data channel over which the server sends the requested data. The security device intercepts the data and passes it to its internal AV scan engine, which scans it for viruses.
3. After completing the scan, the device follows one of two courses*:
   - If there is no virus, it forwards the data to the client.
   - If there is a virus, it replaces the data with a drop message in the data channel and sends a message reporting the infection in the control channel.

* If the scanned data exceeds the maximum content setting, or, if the scan cannot be successfully completed, the device follows a different course of action depending on the fail-mode setting.

Scanning HTTP Traffic

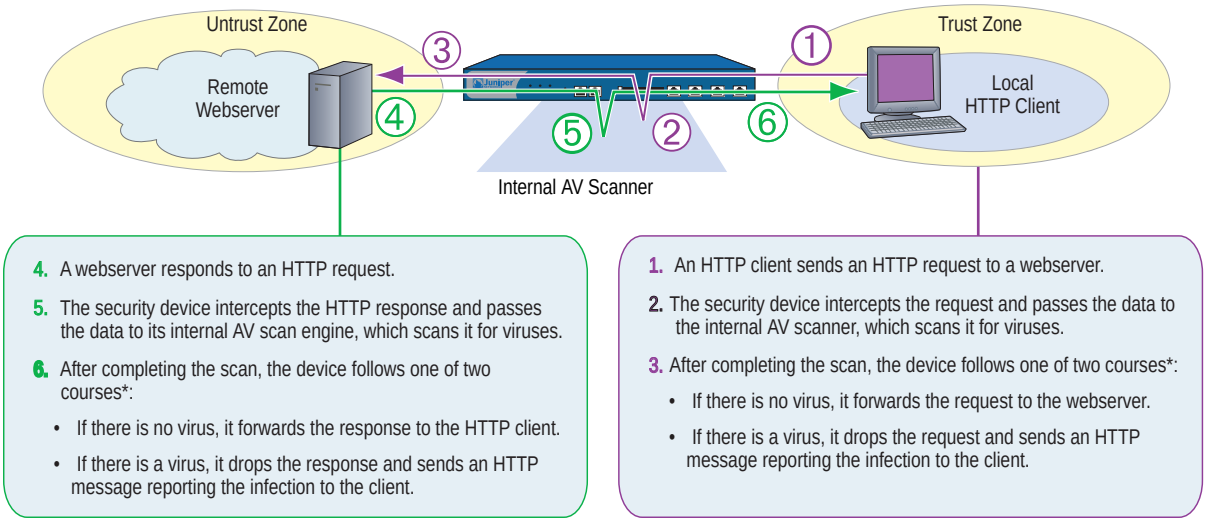
For HTTP traffic scanning, the security device scans both HTTP responses and HTTP requests (**get**, **post**, and **put** commands). The internal AV scanner examines HTTP downloads; that is, HTTP data contained in responses from a webserver to HTTP requests from a client. The internal AV scanner also scans uploads, such as when an HTTP client completes a questionnaire on a webserver or when a client writes a message in an email originating on a webserver.

Depending on the results of the scan and how you have configured the fail-mode behavior, the security device takes one of the following actions:

If the Data	And	The Security Device
is uncontaminated		passes the data to the HTTP client
contains a virus		drops the data and sends a virus notification message to the HTTP client
exceeds the maximum content size	**drop** is set	drops the data and sends a “file too large” message to the HTTP client
exceeds the maximum content size	**drop** is unset	passes the data to the HTTP client
cannot successfully be scanned	**fail mode** is unset	drops the data and sends a “scan error” message to the HTTP client
cannot successfully be scanned	**traffic permit** is set	passes the data to the HTTP client

If the Data	And	The Security Device
exceeds the maximum concurrent messages	**drop** is set	drops data from the data channel and sends an "exceeding maximum message setting" message to the HTTP client through the control channel
exceeds the maximum concurrent messages	**drop** is unset	passes the data to the HTTP client through the data channel

Figure 32: Antivirus Scanning for HTTP Traffic



* If the scanned data exceeds the maximum content setting, or, if the scan cannot be successfully completed, the security device follows a different course of action depending on the fail-mode setting.

HTTP MIME Extensions

By default, HTTP scanning does not scan HTTP entities composed of any of the following Multipurpose Internet Mail Extensions (MIME) content types and—when present following a slash—subtypes:

- Application/x-director
- Application/pdf
- Image/
- Video/
- Audio/
- Text/css
- Text/html

To improve performance, Juniper Networks security devices do not scan the above MIME content types. Because most HTTP entities are made up of the above content types, HTTP scanning only applies to a small subset of HTTP entities, such as application/zip and application/exe content types, where viruses are most likely to be hiding.

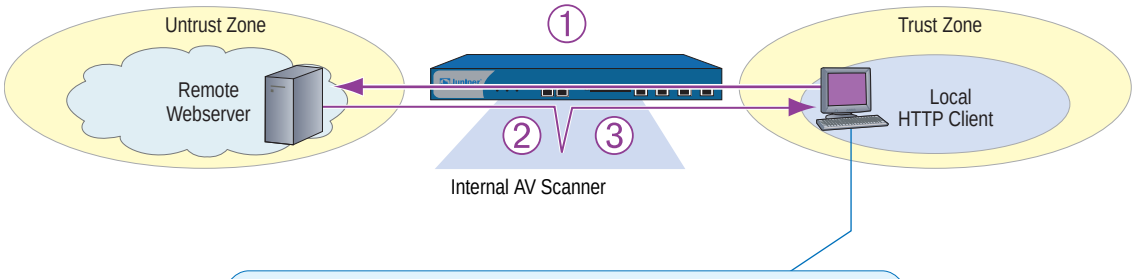
To change HTTP scanning behavior so that the security device scans all kinds of HTTP traffic regardless of MIME content types, enter the following command:

```
set av profile jnpr-profile
(av:jnpr-profile)-> unset av http skipmime
(av:jnpr-profile)-> exit
save
```

HTTP Webmail

For HTTP webmail traffic scanning, the security device redirects the webserver replies (responding to a client's HTTP webmail requests) to the internal AV scanner before it forwards the traffic to the client.

Figure 33: Antivirus Scanning for HTTP Webmail Traffic



1. A local HTTP client sends an HTTP web-mail request to a remote webserver, which the security device permits.
2. The device intercepts the inbound HTTP reply and passes the HTTP data to its internal AV scan engine, which scans it for viruses.
3. After completing the scan, the device follows one of two courses*:
   - If there is no virus, it forwards the message to the client.
   - If there is a virus, it drops the message and sends an HTTP message reporting the infection to the client.

* If the scanned data exceeds the maximum content setting, or, if the scan cannot be successfully completed, the security device follows a different course of action depending on the fail-mode setting. If a virus is found in an element of the HTML page, the element content is replaced by white space.

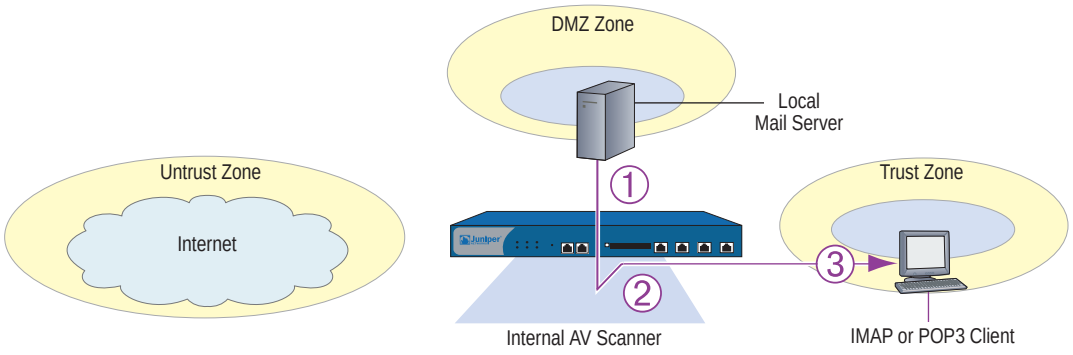
Scanning IMAP and POP3 Traffic

For IMAP and POP3 traffic scanning, the Juniper Networks security device redirects traffic from a local mail server to the internal AV scanner before sending it to the local IMAP or POP3 client. Depending on the results of the scan and how you have configured the fail-mode behavior, the device takes one of the following actions:

If the Data	And	The Security Device
is uncontaminated		passes the message to the IMAP or POP3 client.
contains a virus	**email notification** is set	changes the content type to "text/plain," replaces the body of the message with the following notice, sends it to the IMAP or POP3 client, and notifies sender: VIRUS WARNING. Contaminated File: *filename* Virus Name: *virus_name*

If the Data	And	The Security Device
exceeds the maximum content size or cannot successfully be scanned or exceeds the maximum concurrent messages	**drop** is set **fail mode** is unset **email notification** is set to sender/recipient	changes the content type to "text/plain," replaces the body of the message with the following notice, and sends it to the IMAP or POP3 client: Content was not scanned for viruses because *reason_text_str* (code *number*), and it was dropped. The *reason_text_str* can be one of the following: ■ The file was too large. ■ An error or a constraint was found. ■ The max. content size was exceeded. ■ The max. number of messages was exceeded. Notifies scan error problems to sender/recipient.
exceeds the maximum content size or cannot successfully be scanned or exceeds the maximum concurrent messages	**drop** is unset **traffic permit** is set **drop** is unset **email notification** is set to sender/recipient	passes the original message to the IMAP or POP3 client with the original subject line modified as follows: *original_subject_text_str* (No virus check because *reason_text_str*, code *number*) Notifies scan error problems to sender/recipient.

Figure 34: Antivirus Scanning for IMAP and POP3 Traffic



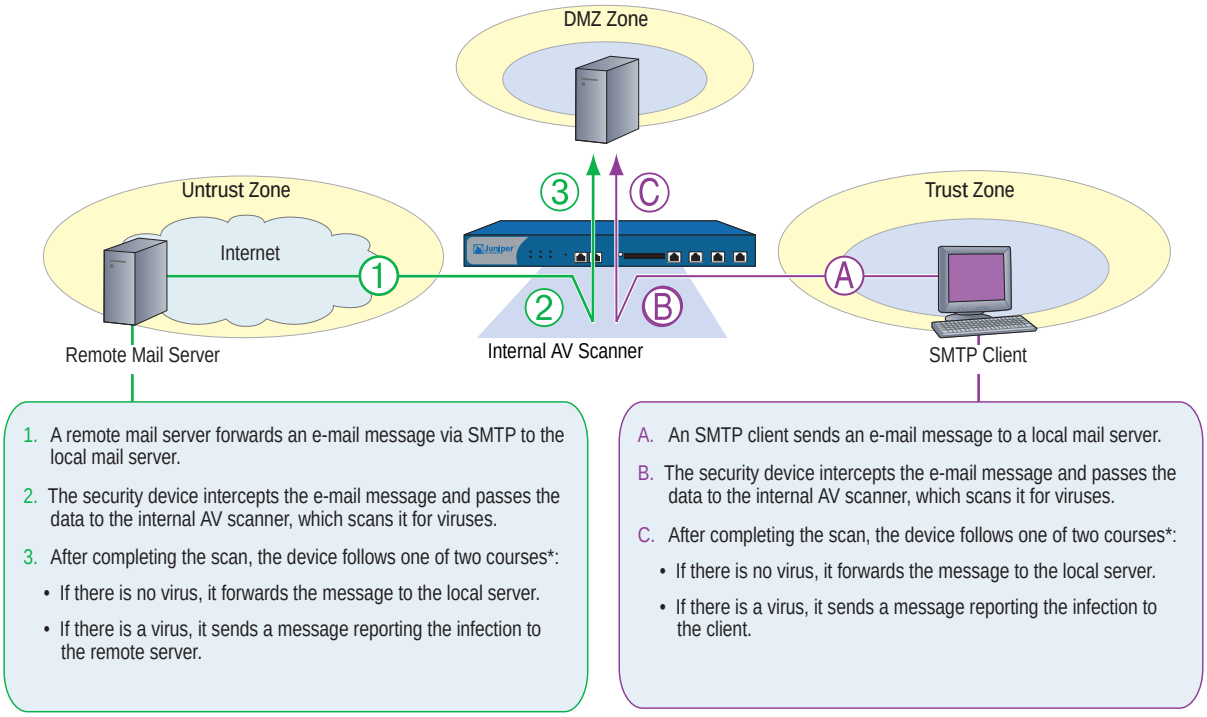
Scanning SMTP Traffic

For SMTP traffic scanning, the Juniper Networks security device redirects traffic from local SMTP clients to the internal AV scanner before sending it to the local mail server. Depending on the results of the scan and how you have configured the fail-mode behavior, the device takes one of the following actions:

If the Data	And	The Security Device
is uncontaminated		passes the message to the SMTP recipient.
contains a virus	**email notification** is set	changes the content type to "text/plain," replaces the body of the message with the following notice, sends it to the SMTP recipient, and notifies sender: VIRUS WARNING. Contaminated File: *filename* Virus Name: *virus_name*
exceeds the maximum content size or cannot successfully be scanned or exceeds the maximum concurrent messages	**drop** is set **fail mode** is unset **drop** is set **email notification** is set to sender/recipient	changes the content type to "text/plain," replaces the body of the message with the following notice, and sends it to the SMTP recipient: Content was not scanned for viruses because *reason_text_str* (code *number*), and it was dropped. The *reason_text_str* can be one of the following: ■ The file was too large. ■ An error or constraint was found. ■ The max. content size was exceeded. ■ The max. number of messages was exceeded Notifies scan error problems to sender/recipient.
exceeds the maximum content level or cannot successfully be scanned or exceeds the maximum concurrent messages	**drop** is disabled **traffic permit** is set **drop** is unset **email notification** is set to sender/recipient	passes the original message to the SMTP recipient with the original subject line modified as follows: *original_subject_text_str* (No virus check because *reason_text_str*, code *number*) Notifies scan error problems to sender/recipient.

NOTE: Because an SMTP *client* refers to the entity that sends email, a client might, in fact, be another SMTP server.

Figure 35: Antivirus Scanning for SMTP Traffic



* If the scanned message exceeds the maximum content setting, or, if the scan cannot be successfully completed, the security device follows a different course of action depending on the fail-mode setting.

Updating the AV Pattern File

Internal AV scanning requires that you load a database of AV patterns onto the Juniper Networks security device and update the pattern file periodically. To do so, you must register the device and purchase a subscription for the AV signature service. The subscription allows you to load the current version of the database and update it as newer versions become available for the life of the subscription. The procedure for initiating the AV signature service varies:

- If you purchased a security device with AV functionality, you can load an AV pattern file for a short period after the initial purchase. You must, however, register the device and purchase a subscription for AV signatures in order to continue receiving pattern updates.
- If you are upgrading a current security device to use internal AV scanning, you must register the device and purchase a subscription for AV signatures before you can begin loading the AV pattern file. After completing the registration process, you must wait up to four hours before initiating the AV pattern file download.

NOTE: For more information about the AV signature service, see “Registration and Activation of Subscription Services” on page **2-**253.

Figure 36 and Figure 37 illustrate how the pattern file is updated. The process of updating the AV pattern file is as follows:

1. On the security device, specify the URL address of the pattern-update server.

   Depending on your AV scan engine type, use one of the following two default pattern-update URLs:

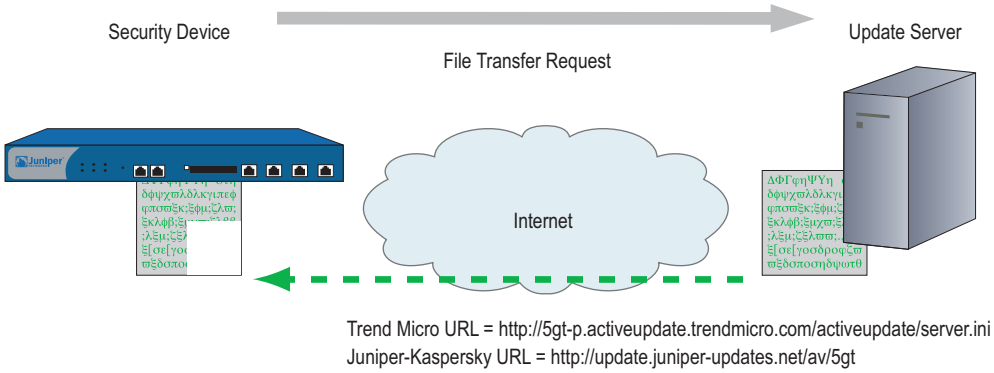
   - Juniper-Kaspersky antivirus scanner

     http://update.juniper-updates.net/av/5gt

   - Trend Micro antivirus scanner

     http://5gt-p.activeupdate.trendmicro.com/activeupdate/server.ini.

Figure 36: Updating Pattern Files—Step 1

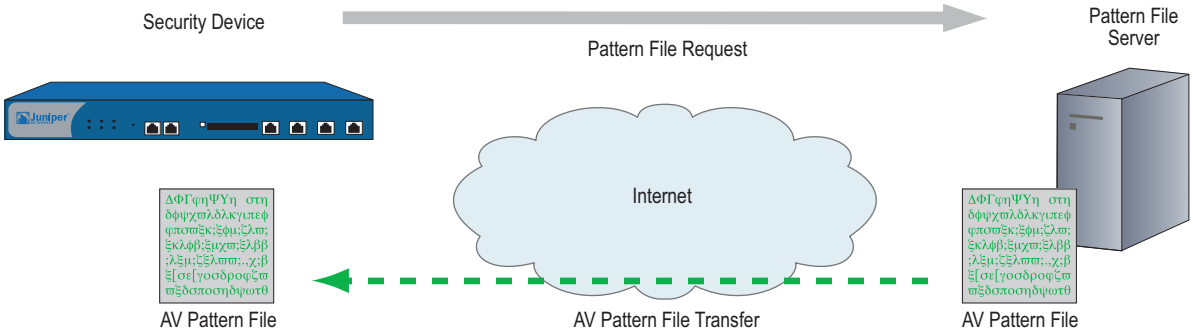


2. After the security device downloads the server initialization file, the device checks that the pattern file is valid. It then parses it to obtain information about the updated pattern file, including the pattern file version and size and the location of the pattern fileserver.

NOTE: ScreenOS contains a CA certificate for authenticating communication with the pattern update fileserver.

3. If the pattern file on the security device is out of date (or nonexistent because this is the first time you are loading it), and, if the AV pattern update service subscription is still valid, the device automatically retrieves an updated pattern file from the pattern fileserver.

Figure 37: Updating Pattern Files—Step 2



4. The device saves the new pattern file to flash and RAM memory and overwrites the existing file, if there was one.

Updates to the pattern file are added as new viruses propagate. You can configure the device to regularly update the pattern file automatically, or you can update the file manually.

NOTE: Once your subscription expires, the update server no longer permits you to update the AV pattern file.

Example: Automatic Update

In this example, you configure the security device to update the pattern file automatically every 15 minutes. (The default AV pattern-update interval is 60 minutes.) For example, if the pattern-update server is located at the URL address: http://5gt-p.activeupdate.trendmicro.com/activeupdate/server.ini, you configure automatic update as follows:

WebUI

Screening > Antivirus > Scan Manager: Enter the following, then click **Apply**:

Pattern Update Server: http://update.juniper-updates.net/av/5gt
Auto Pattern Update: (select), Interval: 120 minutes (10~10080)

CLI

```
set av scan-mgr pattern-update-url http://update.juniper-updates.net/av/5gt
    interval 120
save
```

Example: Manual Update

In this example, you update the pattern file manually. The pattern update server is located at the following URL address: http://update.juniper-updates.net/av/5gt

WebUI

Screening > Antivirus > Scan Manager: Enter the following, then click **Apply**:

Pattern Update Server: http://update.juniper-updates.net/av/5gt
Update Now: (select)

CLI

exec av scan-mgr pattern-update

The **set** command is not required because the URL address is the default.

Spyware and Phishing Protection

The Juniper-Kaspersky scan engine, by default, provides the highest level of security. In addition to stopping all viruses (including polymorphic and other advanced viruses), the new scan engine also provides inbound spyware and phishing protection.

Spyware protection. The spyware-protection feature adds another layer of protection to Juniper Networks anti-spyware and anti-adware solutions by letting you block incoming spyware, adware, keyloggers, and related malware to prevent it from penetrating your enterprise.

This solution complements Juniper Networks IDP products, which provide spyware phone-home protection (that is, stopping spyware from sending sensitive data if your laptops/desktops/servers are infected).

Phishing protection. The "phishing" protection allows you to block emails that try to entice users to fake (phishing) sites that steal sensitive data from them.

You may choose to change the default security level of scanning with the following two options:

- **Basic in-the-wild scanning.** This level of scanning administers a lower degree of security by scanning the most prevalent viruses, although it provides increased performance.
- **Extended scanning.** This level of scanning includes traditionally more noisy pieces of spyware/adware to the standard scan. It provides more spyware coverage but potentially can return more false positives.

WebUI

NOTE: You must use the CLI to configure this option.

CLI

set av scan-mgr pattern-type standard

The standard option is the default.

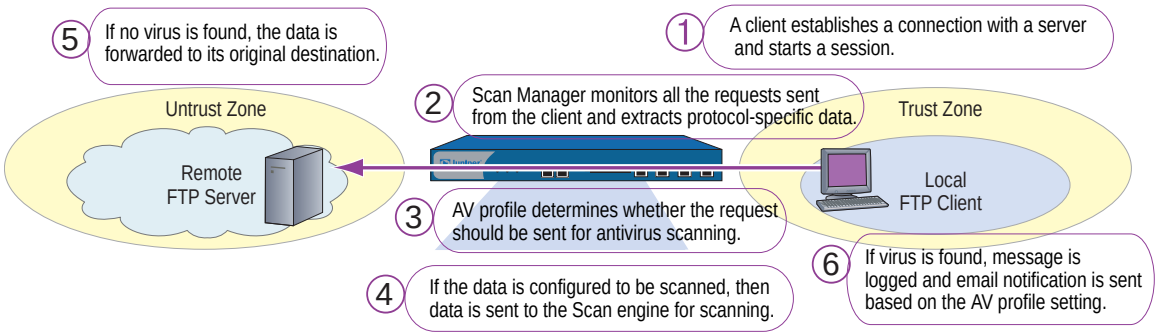
Policy-Based AV Scanning

To enhance performance and control the AV scan engine, use policy-based scanning to allow you to select specific data traffic for AV scanning. To configure policy-based scanning, you must configure AV profiles for use in policies.

1. Initiate an AV profile context.
2. Configure the profile (the predefined profile is ns-profile).
   - Select data for AV scanning based on one or more of the following:
     - Application protocols (FTP, HTTP, IMAP, POP3, or SMTP) with timeout values
     - File extensions
     - Content type
   - Disable skipmime list to allow the security device to scan all kinds of HTTP traffic regardless of MIME content types.
   - Notify virus or scanning errors (for IMAP, POP3, and SMTP traffic only) to sender or recipients
3. Exit the AV profile context.
4. Assign the AV profile to an existing policy. (Only one AV profile can be linked to a firewall policy.)
5. Save your profile.

Figure 38 shows how the AV profile works with the AV scanner.

Figure 38: How the AV Profile Works with the AV Scanner



AV Scanner Global Settings

Modify the AV scanner settings to serve the needs of your network environment. The following sections explain the global settings on your AV scanner:

- AV Resource Allotment
- Fail-Mode Behavior
- Maximum Content Size and Maximum Messages
- HTTP Keep-Alive
- HTTP Trickling

AV Resource Allotment

A malicious user might simultaneously generate a large amount of traffic in an attempt to consume all available resources and thereby hinder the ability of the AV scanner to scan other traffic. To prevent such activity from succeeding, the Juniper Networks security device can impose a maximum percentage of AV resources that traffic from a single source can consume at any one time. The default maximum percentage is 70 percent. You can change this setting to any value between 1 and 100 percent, where 100 percent does not impose any restriction on the amount of AV resources that traffic from a single source can consume.

WebUI

NOTE: You must use the CLI to configure this option.

CLI

```
set av all resources number
unset av all resources
```

The above **unset av** command returns the maximum percentage of AV resources per source to the default (70 percent).

Fail-Mode Behavior

Fail-mode is the behavior that the security device applies when it cannot complete a scan operation—either to permit the unexamined traffic or to block it. By default, if a device cannot complete a scan, it blocks the traffic that a policy with antivirus checking enabled permits. You can change the default behavior from block to permit.

WebUI

Screening > Antivirus > Global: Select **Fail Mode Traffic Permit** to permit unexamined traffic, or clear it to block unexamined traffic, then click **Apply**.

CLI

```
set av all fail-mode traffic permit
unset av all fail-mode traffic
```

The above **unset av** command returns the fail mode behavior to the default (block unexamined traffic).

Maximum Content Size and Maximum Messages

The AV scanner examines a maximum of 16 messages and 10 megabytes of decompressed file content at any time. If the total number of messages or the size of the content received concurrently exceeds these limits, by default the scanner drops the content without checking for viruses.

For example, the scanner can receive and examine four 4-megabyte messages concurrently. If the scanner receives nine 2-megabyte messages concurrently, it passes the content without scanning it. You can change this default behavior so that the scanner passes traffic instead of dropping it.

NOTE: The default for Maximum Content Size is 10 MB. However, if DI is enabled, Juniper Networks recommends configuring a value of 6 MB.

WebUI

Screening > Antivirus > Scan Manager: Select **pass** if file size exceeds 10,000KB

Or

Select **pass** if number of files exceeds 16, then click **Apply**.

CLI

```
set av scan-mgr max-content-size drop
set av scan-mgr max-msgs drop
```

HTTP Keep-Alive

By default, the security device uses the HTTP "close" connection option for indicating the end of data transmission. (If necessary, the device changes the token in the connection header field from "keep-alive" to "close.") In this method, when the HTTP server completes its data transmission, it sends a TCP FIN to close the TCP connection and thereby indicate that it has finished sending data. When the device receives a TCP FIN, it has all the HTTP data from the server and can instruct the AV scanner to begin scanning.

You can change the default behavior of the security device to use the HTTP "keep-alive" connection option, which does not send a TCP FIN to indicate the termination of data transmission. The HTTP server must indicate that it has sent all the data in another way, such as by sending the content length in the HTTP header or by some form of encoding. (The method that a server uses varies by server type.) This method keeps the TCP connection open while the antivirus examination occurs, which decreases latency and improves processor performance. However, it is not as secure as the "close" connection method. You can change this behavior if you find that HTTP connections are timing out during the antivirus examination.

WebUI

Screening > Antivirus > Global: Select **Keep Alive** to use the "keep-alive" connection option, or clear it to use the "close" connection option, then click **Apply**.

CLI

set av http keep-alive
unset av http keep-alive

HTTP Trickling

HTTP trickling is the forwarding of specified amounts of unscanned HTTP traffic to the requesting HTTP client to prevent the browser window from timing out while the scan manager examines downloaded HTTP files. (The security device forwards small amounts of data in advance of transferring an entire scanned file.) By default, HTTP trickling is disabled. To enable it and use the default HTTP trickling parameters, do either of the following:

WebUI

Screening > Antivirus > Global: Select the Trickling Default checkbox, then click **Apply**.

CLI

set av http trickling default

With the default parameters, the security device employs trickling if the size of an HTTP file is 3 megabytes or larger. Then it forwards 500 bytes of content for every 1 megabyte sent for scanning.

To change the parameters for HTTP trickling, do either of the following:

WebUI

Screening > Antivirus > Global: Enter the following, then click **Apply**:

Trickling:
Custom: (select)
Minimum Length to Start Trickling: Enter ***number1***.
Trickle Size: Enter ***number2***.
Trickle for Every MB Sent for Scanning: Enter ***number3***.

CLI

set av http trickling *number1 number3 number2*

The three *number* variables have the following meanings:

- *number1:* The minimum size (in megabytes) of an HTTP file to trigger trickling
- *number2:* The size (in bytes) of unscanned traffic that the security device forwards
- *number3*: The size (in megabytes) of a block of traffic to which the security device applies trickling

---

NOTE: Data trickled to the client's hard drive appears as a small, unusable file. Because trickling works by forwarding a small amount of data to a client without scanning it, virus code might be among the data that the security device has trickled to the client. We advise users to delete such files.

You can disable HTTP trickling in the WebUI (Screening > Antivirus: Click **Disable** in the Trickling section) or with the CLI command **unset av http trickling enable**. However, if a file being downloaded is larger than 8 MB and HTTP trickling is disabled, the browser window will most likely time out.

AV Scanner Profile Settings

Policies use AV profiles to determine which traffic undergoes AV examination and the actions to take as a result of this examination.

NOTE: A predefined AV profile named **ns-profile** exists on your Juniper Networks security device.

You must do the following to link the AV profile to a firewall policy. Only one AV profile can be linked to a firewall policy.

WebUI

Policies: Click **Edit** on the policy to which you want to link the AV profile and select the profile under Antivirus Profile. Click **OK**.

CLI

ns5gt1-> **set policy id** *policy_num* **av ns-profile**

The following sections explain how to initiate an AV profile and configure the profile settings:

- Initiating an AV Profile
- Example: Scanning All Traffic Types
- Example: AV Scanning for SMTP and HTTP Traffic Only
- AV Profile Settings

Initiating an AV Profile

The following commands initiate a custom AV profile named *jnpr-profile,* which by default is configured to scan FTP, HTTP, IMAP, POP3, and SMTP traffic.

WebUI

Screening > Antivirus > Profile: Select **New** and enter the profile name, *jnpr-profile,* then click **OK**.

CLI

set av profile *jnpr-profile*
ns5gt(av:jnpr-profile)->

ns5gt1-> **set av profile jnpr-profile**
ns5gt(av:jnpr-profile)->

After you enter an AV profile context, all subsequent command executions modify the specified AV profile (*jnpr-profile*).

Example: Scanning All Traffic Types

In this example, you configure the AV scanner to examine FTP, HTTP, IMAP, POP3, and SMTP traffic. Because you anticipate that the scanner will be processing a lot of traffic, you also increase the timeout from 180 seconds (the default setting) to 300 seconds.

WebUI

Screening > Antivirus > Profile: Enter *profile_name*, then click **OK**.

By default, traffic for all five protocols, FTP, HTTP, IMAP, POP3, and SMTP, is scanned.

NOTE: To change the timeout value, you must use the CLI.

CLI

```
set av profile jnpr-profile
(av:jnpr-profile)-> set http enable
(av:jnpr-profile)-> set http timeout 300
(av:jnpr-profile)-> set ftp enable
(av:jnpr-profile)-> set ftp timeout 300
(av:jnpr-profile)-> set imap enable
(av:jnpr-profile)-> set imap timeout 300
(av:jnpr-profile)-> set pop3 enable
(av:jnpr-profile)-> set pop3 timeout 300
(av:jnpr-profile)-> set smtp enable
(av:jnpr-profile)-> set smtp timeout 300
(av:jnpr-profile)-> exit
save
```

Example: AV Scanning for SMTP and HTTP Traffic Only

By default, the AV scanner examines FTP, HTTP, IMAP, POP3, and SMTP traffic. You can change the default behavior so that the scanner examines specific types of network traffic only.

You can also change the timeout value for each protocol. By default, an AV scan operation times out after 180 seconds if the security device does not start scanning after it receives all the data. The range is 1 to 1800 seconds.

In this example, you configure the AV scanner to examine all SMTP and HTTP traffic. You return the timeout value for both protocols to their defaults: 180 seconds.

NOTE: The internal AV scanner examines specific HTTP webmail patterns only. The patterns for Yahoo!, Hotmail, and AOL mail services are predefined.

WebUI

Screening > Antivirus > Select **New** and enter the profile name *jnpr-profile.*

Enter the following, then click **OK**.

Protocols to be scanned:
HTTP: (select)
SMTP: (select)
POP3: (clear)
FTP: (clear)
IMAP: (clear)

NOTE: To change the timeout value, you must use the CLI.

CLI

```
set av profile jnpr-profile
(av:jnpr-profile)-> set smtp timeout 180
(av:jnpr-profile)-> set http timeout 180
(av:jnpr-profile)-> unset pop3 enable
(av:jnpr-profile)-> unset ftp enable
(av:jnpr-profile)-> unset imap enable
(av:jnpr-profile)-> exit
unset av http webmail enable
save
```

AV Profile Settings

The following scanning options are configured for each application protocol:

- "Decompressing File Attachments" on page 74
- "AV Scanning Based on File Extensions" on page 75
- "AV Scanning Based on HTTP Content Type" on page 75
- "Notifying Sender and Recipient via Email" on page 76
- "Example: Dropping Large Files" on page 76

Decompressing File Attachments

When the device receives content, the internal AV scanner decompresses any compressed files. It decompresses up to two layers of compressed files by default. For example, if the scanner receives a file with an attachment and the attachment is a compressed file layered within another compressed file, the scanner may decompress both layers in order to detect any viruses. You can configure the internal AV scanner to decompress up to four compressed files layered within one another.

WebUI

Screening > Antivirus > Profile: Select **New** or **Edit** to edit an existing profile. Update the Decompress Layer to 2, then click **Apply**.

CLI

```
set av http keep-alive
unset av http keep-alive
```

AV Scanning Based on File Extensions

File-extension lists are used to make decisions on which files undergo AV scanning for a specific protocol. There is one inclusion file-extension list and one exclusion file-extension list for each protocol.

A message is scanned when the file extension of a message is in the inclusion file-extension list. A message is not scanned if the file extension is in the exclusion file-extension list. If the file extension is neither in the inclusion nor the exclusion file-extension list, then the scanning decision depends on the default file-extension-scan setting. The default file extension is in the scan engine database, so it is read-only. There is no predefined file extension list for each protocol.

Configure the AV scanner to scan IMAP traffic by extensions and exclude files with the following extensions: .ace, .arj, and .chm.

WebUI

Screening > Antivirus > Ext-list > New > enter an extension-list name (elist1), and enter the list of extensions (ace;arj;chm). Click **OK**.

Antivirus > Profile > Select the Profile to **Edit** > Select **IMAP** > Select the following options, then click **OK**:

Enable
Scan Mode: Scan by Extension
Exclude Extension List: elist1

CLI

```
set av extension-list elist1 ace;arj;chm
set av profile test1
(av:test1)-> set imap scan-mode scan-ext
(av:test1)-> set imap extension-list exclude elist1
```

AV Scanning Based on HTTP Content Type

Use this option to decide which HTTP traffic must undergo AV scanning. The HTTP traffic is categorized into default predefined Multipurpose Internet Mail Extensions (MIME) types such as application/x-director, application/pdf, image, and so on. You can configure the AV profile to skip MIME lists containing specific MIME types. The default predefined MIME list is ns-skip-mime-list.

In this example, you configure the security device to scan all kinds of HTTP traffic regardless of MIME content type:

WebUI

Screening > Antivirus > Profile > Select the Profile to **Edit** > Select HTTP and clear the Skipmime Enable option. Click **OK**.

CLI

```
set av profile jnpr-profile
(av:jnpr-profile)-> unset av http skipmime
(av:jnpr-profile)-> exit
save
```

For more information on MIME types, refer to *ScreenOS CLI Reference Guide IPv4 Command Descriptions*.

Notifying Sender and Recipient via Email

The email-notification option applies to IMAP, POP3, and SMTP protocols only. You can configure the AV profile to notify scanning errors or virus information to senders or recipients.

When a virus is found in an email message, the content of the warning message (virus name, source/destination IP) is included in a notification-level message. The warning level message is sent via an email through the SMTP protocol.

When a scanning error occurs in a message, the content of the scanning error message should be included in a warning-level message. This message is sent via an email through the SMTP protocol.

In this example, you configure the security device to do the following:

- Notify the sender when a virus is detected
- Notify the sender and recipients if scanning errors occur

WebUI

Screening > Antivirus > Profile > Select the Profile to **Edit** > Select IMAP, then click **OK**.

Enter the following, then click **OK**.

Protocols to be scanned:
Email Notify > Select Virus Sender
Email Notify > Select Scan-error Sender
Email Notify > Select Scan-error Recipient

CLI

```
set av profile jnpr-profile
(av:jnpr-profile)-> set imap email-notify virus sender
(av:jnpr-profile)-> set imap email-notify scan-error sender
(av:jnpr-profile)-> set imap email-notify scan-error recipient
(av:jnpr-profile)-> exit
save
```

Example: Dropping Large Files

In this example, you configure the AV scanner to decompress HTTP traffic of up to three files layered within one another. You also configure the scanner to drop content either if the total number of messages received concurrently exceeds four messages or if the total decompressed size of the content exceeds 12 MB.

WebUI

Screening > Antivirus > Scan Manager: Enter the following, then click **OK**:

Drop: (select) file if it exceeds 3000 KB (20~10000)
Drop: (select) file if the number of files exceeds 4 files (1~16)

Screening > Antivirus > Profile: Select Edit > HTTP: Enter the following, then click **OK**:

File decompression: 3 layers (1~4)

CLI

```
set av scan-mgr max-msgs 4
set av scan-mgr max-content-size 3000
set av scan-mgr max-content-size drop
set av profile jnpr-profile
(av:jnpr-profile)-> set http decompress-layer 3
(av:jnpr-profile)-> exit
save
```

Anti-Spam Filtering

The anti-spam feature examines transmitted messages and decides which are spam and which are not. When the device detects a message deemed to be spam, it either tags the message field with a preprogrammed string, or it drops the message. Anti-spam uses a constantly-updated IP-based spam blocking service that uses information gathered worldwide. Because this service is robust and yields very few false positives, it is not mandatory to tune or configure black lists. However, the administrator has the option of adding specific domains and IPs to local white lists or black lists, which the device can enforce locally. Note: This release supports anti-spam for SMTP protocol only.

Spam consists of unwanted email messages, usually sent by commercial, malicious, or fraudulent entities. To prevent or reduce the volume of received spam messages, you can configure an anti-spam profile. You can use the profile in policies to detect and filter out suspected spam messages. An anti-spam profile allows you to designate lists of IP addresses, emails, hostnames or domain names, designated as malicious (spam) or benign (non-spam). The profile can include lists of the following types:

- Public-based black lists or white lists. If the connection is from a mail forwarding agent, the device can filter the connection's source IP address using lists of devices deemed to be benign (white list) or malicious (black list).
- Domain-name-based white lists or black lists. The device can use such lists to filter connections that use domain names deemed to be benign or malicious.
- Address-book-based white lists or black lists. The device can use such lists to base filtering on the sender’s email address or domain. By default, any email server should accept its own user’s email.

Black Lists and White Lists

The anti-spam feature requires that the firewall have internet connectivity with the Spam Block List server (SBL). The firewall performs reverse DNS lookups on the source of the SMTP sender (or relaying agent), adding the name of the SBL server (such as sbl-server) as the authoritative domain. The DNS server then forwards each request to the SBL server, which returns a value to the firewall.

Alternatively, the user can configure local white and black lists (described above and below). In this case, by default the system checks first against the local database of white/black lists. If it does not find the name, the firewall proceeds to query the SBL server located in the Internet.

Basic Configuration

The following commands provide an example of basic anti-spam configuration.

```
set anti-spam profile ns-profile
set policy from untrust to trust any mail-server SMTP permit log anti-spam ns-profile
```

In the following example, the firewall tests spammer.org to see if it resides on the white list or the black list.

```
exec anti-spam testscan spammer.org
```

If the black list contains spammer.org, the device may produce the following output:

AS: anti spam result: action Tag email subject, reason: Match local blacklist

Alternatively, if the white list contains spammer.org, the device may produce the following output:

AS: anti spam result: action Pass, reason: Match local whitelist

For information on creating black lists or white lists, see “Defining a Black List” on page 79 and “Defining a White List” on page 79.

Filtering Spam Traffic

In the following examples, SMTP traffic with spam traverses the system. However, ScreenOS checks for spam by either DNS name or IP address.

```
ns5gt1-> exec anti-spam test 2.2.2.2
```

AS: anti spam result: action Tag email subject, reason: Match local black list

```
exec anti-spam testscan spammer.org
```

AS: anti spam result: action Tag email subject, reason: Match local black list

Dropping Spam Messages

Executing the **set anti-spam profile** *name_str* command without specifying further options places the CLI within the context of a new or existing anti-spam profile. For example, the following commands define a profile named ns-profile, and then enter the ns-profile context to instruct the device to drop suspected spam messages:

```
ns-> set anti-spam profile ns-profile
ns(ns-profile)-> set default action drop
```

After you enter an anti-spam context, all subsequent command executions modify the specified anti-spam profile (ns-profile in this example). To save your changes, you must first exit the anti-spam context, and then enter the save command:

```
ns(ns-profile)-> exit
ns-> save
```

Defining a Black List

Use the black list commands to add or remove an IP address, email, hostname or domain name from the local anti-spam black list. Each entry in a black list can identify a possible spammer.

Example: These commands perform the following tasks:

1. Initiate a profile context (ns-profile).
2. Give the profile a black list entry that prevents connections with the hostname www.wibwaller.com.
3. Exit the spam context and apply the profile to an existing policy (id 2).

```
ns5gt1-> set anti-spam profile ns-profile
ns5gt1(anti-spam:ns-profile)-> set blacklist www.wibwaller.com
ns5gt1(anti-spam:ns-profile)-> exit
ns5gt1-> set policy id 2 anti-spam ns-profile
```

Defining a White List

Use the white list commands to add or remove an IP address, email, hostname or domain name from the local white list. Each entry in a white list can identify an entity that is not a suspected spammer. The following table shows some possible entries.

Example: These commands perform the following tasks:

1. Initiate a profile context (ns-profile).
2. Give the profile a white list entry that allows connections with the hostname www.fiddwicket.com.
3. Exit the spam context and apply the profile to an existing policy (id 2).

```
ns5gt1-> set anti-spam profile ns-profile
ns5gt1(anti-spam:ns-profile)-> set whitelist www.fiddwicket.com
ns5gt1(anti-spam:ns-profile)-> exit
ns5gt1-> set policy id 2 anti-spam ns-profile
```

Defining a Default Action

Use the default commands to specify how the device handles messages deemed to be spam. The device can either drop a spam message, or identify it as spam by tagging it.

You can place the tag in either of two email message areas.

- In the header of message
- In the subject of message

Example: These commands perform the following tasks:

1. Initiate a profile context (ns-profile).

2. Specify that email messages deemed to be spam have the string “This is spam” in the message header.

3. Exit the spam context and apply the profile to an existing policy (id 2).

```
ns5gt1-> set anti-spam profile ns-profile
ns5gt1(anti-spam:ns-profile)-> set default action tag header "This is spam"
ns5gt1(anti-spam:ns-profile)-> exit
ns5gt1-> set policy id 2 anti-spam ns-profile
```

Defining a Spam-Blocking List

Use the **sbl** command to enable use of the external spam-blocking SBL service, which uses a black list to identify known spam sources. The service replies to queries from the device about whether an IP address, email, hostname or domain name belongs to a known spammer.

Example: These commands perform the following tasks:

1. Initiate a profile context (ns-profile).

2. Enable use of the default anti-spam service.

3. Exit the spam context and apply the profile to an existing policy (id 2).

```
ns5gt1-> set anti-spam profile ns-profile
ns5gt1(anti-spam:ns-profile)-> set sbl default-server-enable
ns5gt1(anti-spam:ns-profile)-> exit
ns5gt1-> set policy id 2 anti-spam ns-profile
```

Web Filtering

Web filtering enables you to manage Internet access and prevent access to inappropriate web content. ScreenOS provides two web filtering solutions:

- Integrated web filtering allows you to permit or block access to a requested site by binding a web-filtering profile to a firewall policy. A web-filtering profile specifies URL categories and the action the security device takes (permit or block) when it receives a request to access a URL in each category. URL categories are either pre-defined and maintained by SurfControl or are user-defined. For information on configuring the integrated web filtering feature, see “Integrated Web Filtering” on page 81.

- Redirect web filtering redirects the security device to send the first HTTP request in a TCP connection to either a Websense server or a SurfControl server, enabling you to block or permit access to different sites based on their URLs, domain names, and IP addresses. For information on configuring the redirect web filtering feature, see “Redirect Web Filtering” on page 89.

Integrated Web Filtering

To enable web filtering, you first bind a web-filtering profile to a policy. With integrated web filtering, the Juniper Networks security device intercepts each HTTP request, determines whether to permit or block access to a requested site by categorizing its URL, then matches the URL category to a web-filtering profile. A web-filtering profile defines the action the security device takes (permit or block) when it receives a request to access a URL.

A URL category is a list of URLs organized by content. Security devices use the SurfControl pre-defined URL categories to determine the category of a URL. SurfControl Content Portal Authority (CPA) servers maintain the largest database of all types of web content classified into about 40 categories. A partial list of the URL categories is shown in "Define URL Categories (Optional)" on page 82 and for a complete list of URL categories developed by SurfControl, visit the SurfControl web site at http://www.surfcontrol.com. In addition to the SurfControl pre-defined URL categories, you can also group URLs and create categories based on your needs. For information on creating user-defined categories, see "Define URL Categories (Optional)" on page 82.

Following is the basic sequence of events when a host in the Trust zone tries an HTTP connection to a server in the Untrust zone.

1. The security device checks for a firewall policy that applies to the traffic.
   - If there is no firewall policy for the traffic, it drops the traffic.
   - If there is a firewall policy and if web filtering is enabled on that policy, the device intercepts all HTTP requests.
2. The device checks if there is a user-defined profile bound to the firewall policy. If there is none, then it uses the default profile, **ns-profile**.
3. The device checks whether the category of the requested URL is already cached. If it is not, the device sends the URL to the SurfControl CPA server for categorization and caches the result.
4. Once the device determines the category of the URL, it checks whether the category is in the web-filtering profile bound to the firewall policy.
   - If the category is in the profile, the device blocks or permits access to the URL as defined in the profile.
   - If the category is not in the profile, it performs the configured default action.

To configure a security device for web perform the steps as described in the following sections:

1. Set a Domain Name System Server on page 82
2. Enable Web-Filtering Context on page 82
3. Define URL Categories (Optional) on page 82
4. Define Web-Filtering Profiles (Optional) on page 84

5. Enable URL Profiles and Policies on page 86

1. Set a Domain Name System Server

The Juniper Networks security device incorporates Domain Name System (DNS) support, allowing you to use domain names as well as IP addresses for identifying locations. You must configure at least one DNS server so the security device can resolve the CPA server name to an address. For information on DNS, see "Domain Name System Support" on page **2-**221.

2. Enable Web-Filtering Context

You can use the WebUI or CLI commands to enable integrated web filtering on a security device. If you use the CLI, you must enter the web-filtering context before entering the commands specific to integrated web filtering. To enter the web-filtering context, use the following CLI command:

```
set url protocol sc-cpa
```

After you enter the previous command, the prompt changes.

```
ns(url:sc-cpa)->
```

This change indicates that you have entered the web-filtering context and can now configure integrated web-filtering parameters.

Example: Enable Web Filtering

In this example, you enable integrated web filtering on a Juniper Networks security device.

WebUI

Screening > Web Filtering > Protocol Selection: Select **Integrated (SurfControl)**, then click **Apply**. Then select **Enable Web Filtering via CPA Server**, and click **Apply** again.

CLI

```
ns-> set url protocol sc-cpa
ns(url:sc-cpa)-> set enable
ns(url:sc-cpa)-> exit
ns-> save
```

3. Define URL Categories (Optional)

A category is a list of URLs grouped by content. There are two types of categories: predefined and user-defined. SurfControl maintains about 40 pre-defined categories. A partial list of the URL categories is shown below. For a complete list and description of each URL category developed by SurfControl, visit the SurfControl web site at http://www.surfcontrol.com.

To view the list of SurfControl pre-defined URL categories, execute the following command:

WebUI

Screening > Web Filtering > Profile > Predefine Category

CLI

```
ns-> set url protocol sc-cpa
ns(url:sc-cpa)-> get category pre
```

The category list displayed is similar to the following:

```
    Type          code    Category name
-------------------------------------------------
    PreDefine     76      Advertisements
    PreDefine     50      Arts & Entertainment
    PreDefine     3001    Chat
    PreDefine     75      Computing & Internet
    .
    .
    .
```

The pre-defined categories list displays the categories and their SurfControl internal codes. Though you cannot list the URLs within a category, you can determine the category of a web site by using the "Test A Site" feature on the SurfControl web site at www.surfcontrol.com.

In addition to the SurfControl pre-defined URL categories, you can group URLs and create categories specific to your needs. Each category can have a maximum of 20 URLs. When you create a category, you can add either the URL or the IP address of a site. When you add a URL to a user-defined category, the device performs DNS lookup, resolves the host name into IP addresses, and caches this information. When a user tries to access a site with the IP address of the site, the device checks the cached list of IP addresses and tries to resolve the host name.

Many sites have dynamic IP addresses, which means that their IP addresses change occasionally. A user attempting to access a site can type an IP address that is not in the cached list on the device. Therefore, if you know the IP addresses of sites you are adding to a category, enter both the URL and the IP address(es) of the site.

NOTE: If a URL is in both a user-defined category and a pre-defined category, the device matches the URL to the user-defined category.

Example: URL Category

In this example, you create a category named **Competitors**, and add the URLs: **www.games1.com** and **www.games2.com**.

WebUI

Screening > Web Filtering > Profile > Custom List > New: Enter the following, then click **Apply**:

```
Category Name: Competitors
URL: www.games1.com
```

Enter the following, then click **OK**:

```
URL: www.games2.com
```

CLI

```
ns-> set url protocol sc-cpa
ns(url:sc-cpa)-> set category competitors url www.games1.com
ns(url:sc-cpa)-> set category competitors url www.games2.com
ns(url:sc-cpa)-> exit
ns-> save
```

4. Define Web-Filtering Profiles (Optional)

A web-filtering profile consists of a group of URL categories and their corresponding actions:

- Permit - The security device allows access to the site.
- Block - The security device does not allow access to the site. When the device blocks access to a site, it displays a message indicating the category of the URL.
- Black List - The security device always blocks access to the sites in the black list. You can create a user-defined category or use a pre-defined category.
- White List - The security device always allows access to the sites in the white list. You can create a user-defined category or use a pre-defined category.

Juniper Networks security devices provide a default profile called **ns-profile**. This profile lists the SurfControl pre-defined URL categories and their corresponding actions. You cannot edit the default profile or add a black or white list. To view the pre-defined profile, use the following command:

WebUI

Screening > Web Filtering > Profile > Predefined Profile

CLI

```
ns-> set url protocol sc-cpa
ns(url:sc-cpa)-> get profile ns-profile
```

The security device displays the pre-defined profile as illustrated below:

```
Web filtering profile name: ns-profile
black-list category: none
white-list category: none

Category            Action
 Advertisements          block
 Arts & Entertainment    permit
 Chat                    permit
 Computing & Internet    permit
 .
 .
 .
 Violence                block
 Weapons                 block
 Web-based Email         permit
 other                   permit
```

If the URL in an HTTP request is not in any of the categories listed in the default profile, the default action of the security device is to permit access to the site.

You can create a profile that is similar to ns-profile, by cloning ns-profile and editing the new profile. Perform the following step in the WebUI to clone ns-profile.

WebUI

Screening > Web Filtering > Profile > Custom Profile: ns-profile: Select **Clone**.

NOTE: You must use the WebUI to clone the predefined profile, ns-profile.

You can also create your own web-filtering profile. When you create a web-filtering profile, you can:

- Add both user-defined and SurfControl pre-defined URL categories.
- Specify a category for the black list and/or the white list.
- Change the default action.

Example: Web-Filtering Profile

In this example, you create a custom profile called **my-profile** with a default action of "permit." Then, you take the category you created in the previous example and add it to my-profile with an action of "block." Note that when you configure the default action using the CLI, you specify the action for the "Other" category.

WebUI

Screening > Web Filtering > Profile > Custom Profile > New: Enter the following, then click **Apply**:

Profile Name: my-profile
Default Action: Permit

Select the following, then click **OK**:

Subscribers Identified by:
Category Name: Competitors (select)
Action: Block (select)
Configure: Add (select)

CLI

```
ns-> set url protocol sc-cpa
ns(url:sc-cpa)-> set profile my-profile other permit
ns(url:sc-cpa)-> set profile my-profile competitors block
ns(url:sc-cpa)-> exit
ns-> save
```

5. Enable URL Profiles and Policies

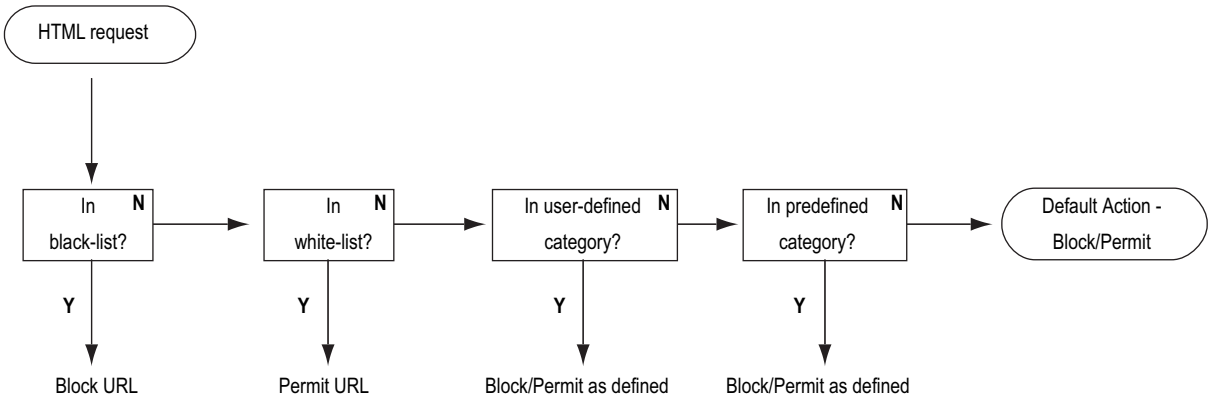
Firewall policies permit or deny specified types of traffic unidirectional between two points. (For information on firewall policies, see “Policies” on page **2-**163.) You can enable both antivirus (AV) scanning and integrated web filtering in a policy. (For information on AV scanning, see “Antivirus Scanning” on page 57.)

When you enable integrated web filtering in a policy, the security device intercepts all HTTP requests. If there is a web-filtering profile bound to the policy, the device matches the URL in the incoming HTTP request to the categories in the profile in the following sequence:

1. Black list
2. White list
3. User-defined categories
4. SurfControl pre-defined URL categories

If the device does not find the category of the requested URL, then it blocks or permits access to the URL, based on the default action that is configured.

Figure 39: URL Profiles and Policies Flow Chart



If the URL is allowed, the security device performs AV scanning on the content of the transaction, if AV scanning is enabled and configured. If the URL is blocked, the device closes the TCP connection, sends a message to the user, and does not check for AV scanning.

Example: Integrated Web Filtering

In this example, you perform the following steps to enable integrated web filtering on the security device and block access to the competitors sites.

1. Create a category called **Competitors**.
2. Add the following URLS to the category: **www.comp1.com** and **www.comp2.com**.
3. Create a profile called **my-profile** and add the **Competitors** category to my-profile.
4. Apply **my-profile** to a firewall policy.

WebUI

1. **Web Filtering**
   Screening > Web Filtering > Protocol Selection: Select **Integrated (SurfControl)**, then click **Apply**. Then select **Enable Web Filtering via CPA Server**, and click **Apply** again.

2. **URL Category**
   Screening > Web Filtering > Profile > Custom List > New: Enter the following, then click **Apply**:

   Category Name: Competitors
   URL: www.comp1.com

   Enter the following, then click **OK**:

   URL: www.comp2.com

3. **Web FIltering Profile**
   Screening > Web Filtering > Profile > Custom Profile > New: Enter the following, then click **Apply**:

   Profile Name: my-profile
   Default Action: Permit

   Category Name: Competitors (select)
   Action: Block (select)
   Configure: Add (select)

4. **Policy**
   Policies > (From: Trust, To: Untrust) New: Enter the following, then click **OK**:

   Source Address:
   Address Book Entry: (select), Any
   Destination Address:
   Address Book Entry: (select), Any
   Service: HTTP
   Web Filtering: (select), my-profile
   Action: Permit

CLI

1. **Web Filtering**
   ns->set url protocol sc-cpa
   ns(url:sc-cpa)-> set enable

2. **URL Category**
   ns(url:sc-cpa)-> set category competitors url www.comp.com
   ns(url:sc-cpa)-> set category competitors url www.comp.com

3. **Web Filtering Profile**
   ns(url:sc-cpa)-> set profile my-profile other permit
   ns(url:sc-cpa)-> set profile my-profile competitors block
   ns(url:sc-cpa)-> exit

4. **Firewall Policy**
   ns-> set policy id 23 from trust to untrust any any http permit url-filter
   ns-> set policy id 23
   ns(policy:23)-> set url protocol sc-cpa profile my-profile
   ns(policy:23)-> exit
   ns-> save

SurfControl Servers

SurfControl has three server locations, each of which serves a specific geographic area: the Americas, Asia Pacific, and Europe/MiddleEast/Africa. The default primary server is the Americas, and the default backup server is Asia Pacific. You can change the primary server, and the security device automatically selects a backup server, based on the primary server. (The Asia Pacific server is the backup for the Americas server, which is also the backup for the other two servers.)

The SurfControl CPA server periodically updates its list of categories. Since the CPA server does not notify its clients when the list is updated, the security device must periodically poll the CPA server. By default, the device queries the CPA server for category updates every two weeks. You can change this default to suit your networking environment. You can also manually update the category list by entering the web-filtering context and executing the **exec cate-list-update** CLI command. To manually update the category list, do the following:

```
ns-> set url protocol sc-cpa
ns(url:sc-cpa)-> exec cate-list-update
```

Web-Filtering Cache

By default, the security device caches the categorization of URLs. This action reduces the overhead of accessing the SurfControl CPA server each time the device receives a new request for previously requested URLs. You can configure the size of the cache and how long the URLs are cached, according to the performance and memory requirements of your networking environment. The default cache size is platform dependent and the default timeout is 24 hours.

Example: Cache Parameters

In this example, you change the cache size to 400 kilobytes (KB) and the timeout value to 18 hours.

WebUI

Screening > Web Filtering > Protocol Selection > SC-CPA: Enter the following, then click **Apply**:

Enable Cache: (select)
Cache Size: 400 (K)
Cache Timeout: 18 (Hours)

CLI

```
ns-> set url protocol sc-cpa
ns(url:sc-cpa)-> set cache size 400
ns(url:sc-cpa)-> set cache timeout 18
ns(url:sc-cpa)-> exit
ns-> save
```

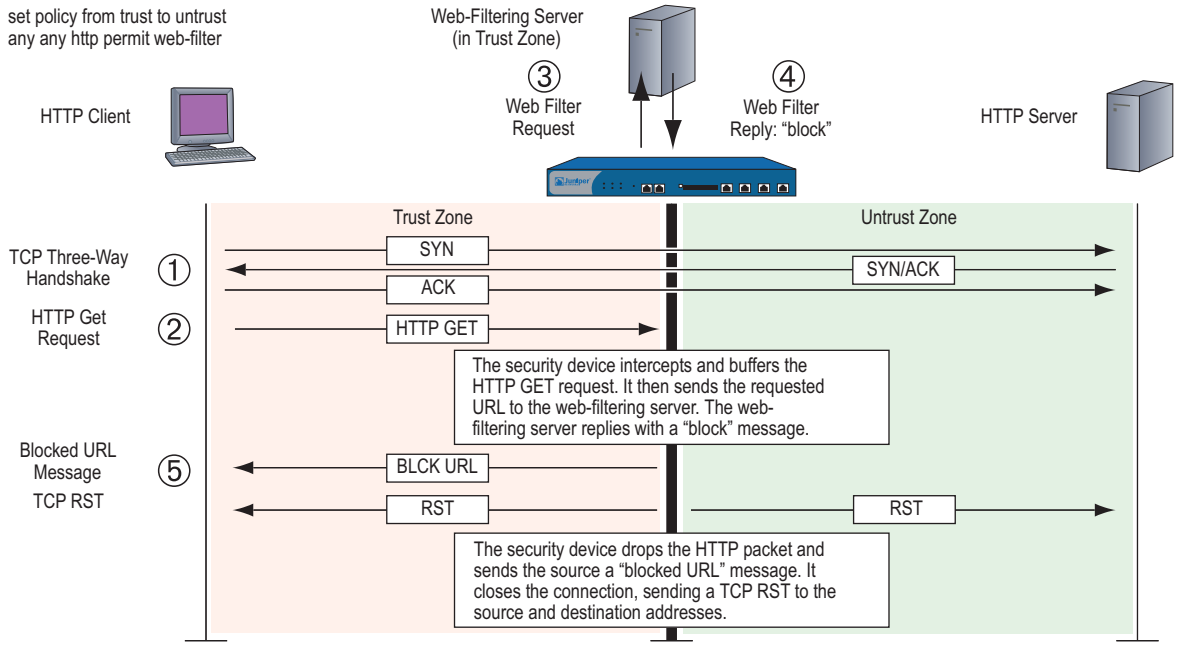
Redirect Web Filtering

Juniper Networks security devices support redirect web filtering using either the Websense Enterprise Engine or the SurfControl Web Filter, both of which enable you to block or permit access to different sites based on their URLs, domain names, and IP addresses. The security device can link directly to a Websense or SurfControl web-filtering server.

NOTE: For additional information about Websense, visit www.websense.com. For additional information about SurfControl, visit www.surfcontrol.com.

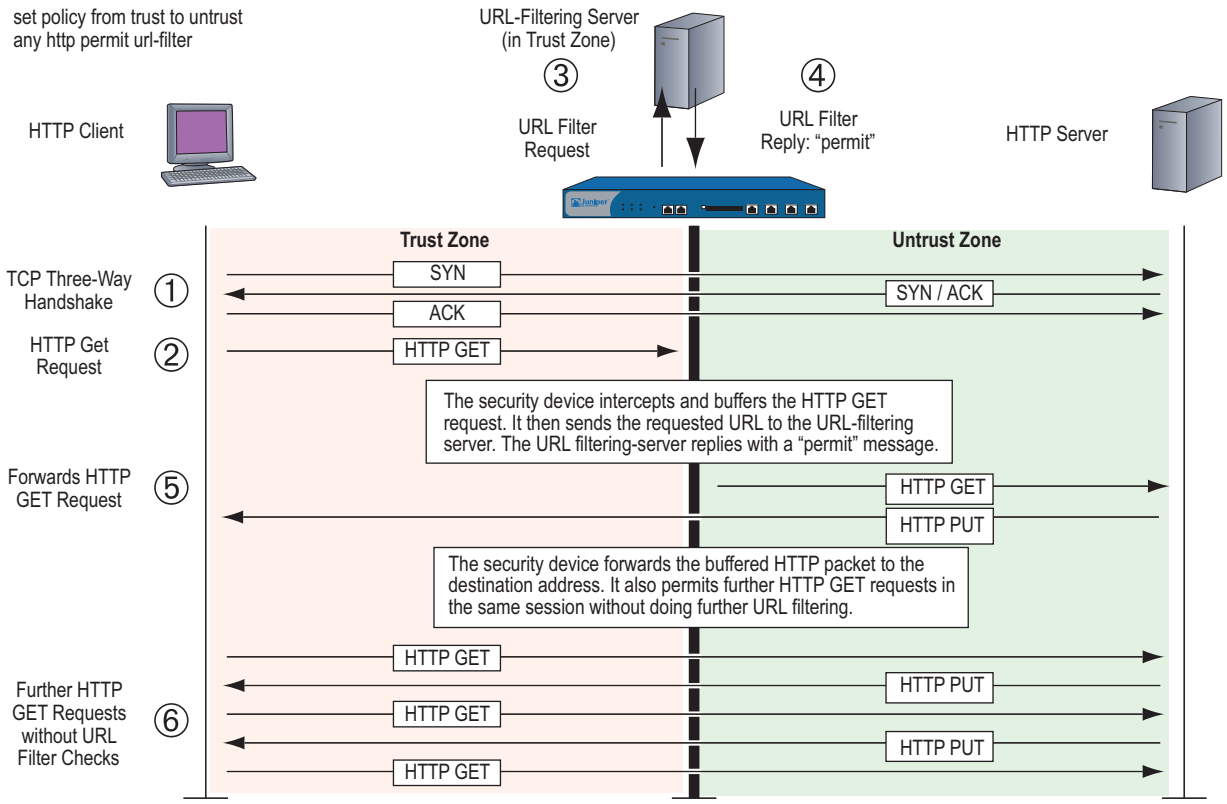
Figure 40 shows the basic sequence of events when a host in the Trust zone attempts an HTTP connection to a server in the Untrust zone. However, web filtering determines that the requested URL is prohibited.

Figure 40: A Blocked URL From Trust Zone to Untrust Zone



If the server permits access to the URL, the sequence of events in the HTTP connection attempt proceeds as shown in Figure 41.

Figure 41: A Permitted URL From Trust Zone to Untrust Zone



Virtual Systems Support

Security devices with virtual systems support up to eight web-filtering servers—one server reserved for the root system, which can be shared with an unrestricted number of virtual systems; and seven web-filtering servers for private use by the virtual systems. A root-level administrator can configure the web-filtering module at the root and virtual system (vsys) levels. A vsys-level administrator can configure the URL module for his or her own vsys if that vsys has its own dedicated web-filtering server. If the vsys-level administrator uses the root web-filtering server settings, that administrator can view—but not edit—the root-level web-filtering settings.

Alternatively, devices with virtual systems that use Websense web-filtering servers can share all eight Websense servers, not just the root server. Each Websense server can support an unrestricted number of virtual systems, allowing you to balance the traffic load among the eight servers.

To configure multiple virtual systems to connect to a Websense web-filtering server, the root-level or vsys administrator must perform the following:

1. Create a account name for each vsys. Use the following CLI command:

   set url account *name*

When a host in a vsys sends out a URL request, it includes the account name. This name enables the Websense server to identify which vsys sent the URL request.

2. Configure the same web-filtering server settings and system-level behavioral parameters for each vsys that shares a Websense web-filtering server. The next section contains information about configuring web-filtering settings and parameters.

Configuring Redirect Web Filtering

To configure a device for redirect web filtering, you must perform the following:

1. Set up communications with up to eight web-filtering servers.

2. Define some system-level behavioral parameters. One set of parameters can apply to the root system and any vsys that shares the web-filtering configuration with the root system. Other sets can apply to virtual systems that have a dedicated web-filtering server.

3. Activate web filtering at the root and vsys levels.

4. Enable web filtering in individual policies.

Details of these steps are provided in the following subsections.

1. Device-to-Device Communications

To configure the security device to communicate with a Websense or SurfControl server, select the server to which you are connecting the device. Select one of the following:

- A Websense server.

- A SurfControl server using the SurfControl Content Filtering Protocol (SCFP).

- A SurfControl server using the Content Portal Authority (CPA) protocol. Use this for the integrated web filtering solution. (For information about integrated web filtering, see "Integrated Web Filtering" on page 81.)

To select the server type, use the either of the following:

WebUI

Screening > Web Filtering > Protocol

CLI

```
set url protocol type { websense | scfp |sc-cpa }
```

If a device connects to a Websense server and the device has multiple virtual systems, the virtual systems can share the server. To configure multiple virtual systems to share a Websense server, use the following CLI command to create a account name for each vsys:

```
set url account name_str
```

Once vsys names are configured, define the settings for the web-filtering server and the parameters for the behavior that you want the security device to take when applying web filtering. If you configure these settings in the root system, they also apply to any vsys that shares the web-filtering configuration with the root system. For a vsys, the root and vsys administrators must configure the settings separately. Virtual systems that share the same Websense web-filtering server must have the same web-filtering settings.

Configure the following web-filtering setting at the system level for device-to-device communications:

- **Server Name:** The IP address or Fully Qualified Domain Name (FQDN) of the computer running the Websense or SurfControl server.
- **Server Port:** If you have changed the default port on the server, you must also change it on the security device. (The default port for Websense is 15868, and the default port for SurfControl is 62252.) Please see your Websense or SurfControl documentation for full details.
- **Source Interface:** The source from which the device initiates web-filter requests to a web-filtering server.
- **Communication Timeout:** The time interval, in seconds, that the device waits for a response from the web-filtering server. If the server does not respond within the time interval, the device either blocks the request or allows it, as you choose. For the time interval, you can enter a number between 10 and 240.

To configure the previously mentioned settings, use either of the following:

WebUI

Screening > Web Filtering > Protocol > Click on the **Websense/SurfControl** hyperlink

CLI

```
set url server { ip_addr | dom_name } port_num timeout_num
```

2. System-Level Behavioral Parameters

After configuring the device communications, define the behavior parameters that you want the system—root or vsys—to take when applying web filtering. The behavior options are as follows:

- **If connectivity to the server is lost:** If the security device loses contact with the web-filtering server, you can specify whether to **Block** or **Permit** all HTTP requests.
- **Blocked URL Message Type:** The source of the message the user receives when Websense or SurfControl blocks a site. If you select **NetPartners Websense/SurfControl**, the security device forwards the message it receives in

the “block” response from the Websense or SurfControl server. When you select **NetScreen**, the device sends the message that you have previously entered in the **NetScreen Blocked URL Message** field.

NOTE: If you select **NetScreen**, some of the functions that Websense provides, such as redirection, are suppressed.

- **NetScreen Blocked URL Message:** This is the message the security device returns to the user after blocking a site. You can use the message sent from the Websense or SurfControl server or you can create a message (up to 500 characters) to be sent from the device.

To configure the previously mentioned settings, use either of the following:

WebUI

Screening > Web Filtering > Protocol > Click on the **Websense/SurfControl** hyperlink

CLI

```
set url fail-mode { block | permit }
set url type { NetScreen | server }
set url message string
```

3. System-Level Activation

Web filtering must be enabled at the system level. For a device that is hosting virtual systems, enable web filtering for each system that you want it applied. For example, if you want the root system and a vsys to apply web filtering, enable web filtering in both the root system and that vsys.

To activate and deactivate web filtering at the system level, use either of the following:

WebUI

Screening > Web Filtering > Protocol >

Click on the **Websense/SurfControl** hyperlink, then clear the **Enable Web Filtering** checkbox.

CLI

```
set url config { disable | enable }
```

When web filtering is enabled at the system level, HTTP requests are redirected to a Websense or SurfControl server. This action allows the device to check all HTTP traffic for policies (defined in that system) that require web filtering. If you disable web filtering at the system level, the device ignores the web-filtering component in policies and treats the policies as “permit” policies.

4. Policy-Level Application

Configure the device to contact the web-filtering server on a per-policy basis.

To enable web filtering in a policy, use the either of the following:

WebUI

In the WebUI, go to Policies > Edit (for the policy you want to apply web filtering) then select the **Web Filter** checkbox.

CLI

set policy from *zone* to *zone src_addr dst_addr service* permit url-filter

NOTE: The device reports the status of the Websense or SurfControl server. To update the status report, click the **Server Status** icon in the WebUI:

Screening > Web Filtering > Protocol > Click on the **Websense/SurfControl** hyperlink

Example: Web Filtering Configuration

In this example, you configure the security device to do the following:

1. Set the interfaces to work with a SurfControl server at IP address 10.1.2.5, with port number 62252 (default), and have the web-filtering server in the Trust security zone.

2. Have web filtering on all outbound HTTP traffic from hosts in the Trust security zone to hosts in the Untrust security zone. If the device loses connectivity with the web-filtering server, the device permits outbound HTTP traffic. When an HTTP client requests access to a prohibited URL, the device sends the following message: "We're sorry, but the requested URL is prohibited. If this prohibition appears to be in error, contact ntwksec@mycompany.com."

3. Set both security zones to be in the trust-vr routing domain with the interface for the Untrust zone as ethernet3 and have IP address of 1.1.1.1/24 and the interface for the Trust zone as ethernet1 and have IP address of 10.1.1.1/24. Because the web-filtering server is not in the immediate subnet of one of the device interfaces, a route is added to it through ethernet1 and the internal router at 10.1.1.250.

4. Configure the policies to enable web filtering so that Trust to Untrust permits HTTP service any source address and any destination address,

WebUI

1. Interfaces

Network > Interfaces > Edit (for ethernet1): Enter the following, then click **Apply**:

Zone Name: Trust
Static IP: (select this option when present)
IP Address/Netmask: 10.1.1.1/24

Enter the following, then click **OK**:

Interface Mode: NAT

Network > Interfaces > Edit (for ethernet3): Enter the following, then click **OK**:

Zone Name: Untrust
Static IP: (select this option when present)
IP Address/Netmask: 1.1.1.1/24

2. **Web-Filtering Server**

Screening > Web Filtering > Protocol: Select **Redirect (SurfControl)**, then click **Apply**. Then enter the following, and click **Apply** again:

Enable Web Filtering: (select)
Server Name: 10.1.2.5
Server Port: 62252
Communication Timeout: 10 (seconds)
If connectivity to the server is lost ... all HTTP requests: Permit
Blocked URL Message Type: NetScreen
NetScreen Blocked URL Message: We're sorry, but the requested URL is prohibited. Contact ntwksec@mycompany.com.

3. **Routes**

Network > Routing > Routing Entries > trust-vr New: Enter the following, then click **OK**:

Network Address/Netmask: 0.0.0.0/0
Gateway: (select)
Interface: ethernet3
Gateway IP Address: 1.1.1.250

Network > Routing > Routing Entries > trust-vr New: Enter the following, then click **OK**:

Network Address/Netmask: 10.1.2.0/24
Gateway: (select)
Interface: ethernet1
Gateway IP Address: 10.1.1.250

4. **Policy**

Policies > (From: Trust, To: Untrust) New: Enter the following, then click **OK**:

Source Address:
Address Book Entry: (select), Any
Destination Address:
Address Book Entry: (select), Any
Service: HTTP
Action: Permit
Web Filtering: (select)

CLI

1. **Interfaces**

   ```
   ns-> set interface ethernet1 zone trust
   ns-> set interface ethernet1 ip 10.1.1.1/24
   ns-> set interface ethernet3 zone untrust
   ns-> set interface ethernet3 ip 1.1.1.1/24
   ```

2. **Web-Filtering Server**

   ```
   ns-> set url protocol type scfp
   ns-> set url server 10.1.2.5 62252 10
   ns-> set url fail-mode permit
   ns-> set url type NetScreen
   ns-> set url message “We’re sorry, but the requested URL is prohibited. Contact
        ntwksec@mycompany.com.”
   ns-> set url config enable
   ```

3. **Routes**

   ```
   ns-> set vrouter trust-vr route 0.0.0.0/0 interface ethernet3 gateway 1.1.1.250
   ns-> set vrouter trust-vr route 10.1.2.0/24 interface ethernet1 gateway
        10.1.1.250
   ```

4. **Policy**

   ```
   ns-> set policy from trust to untrust any any http permit url-filter
   ns-> save
   ```

Chapter 5

Deep Inspection

You can enable Deep Inspection (DI) in policies to examine permitted traffic and take action if the DI module in ScreenOS finds attack signatures or protocol anomalies. The following sections in this chapter present the DI elements that appear in policies and explains how to configure them:

- "Overview" on page 98
- "Attack Object Database Server" on page 102
- "Attack Objects and Groups" on page 110
  - "Supported Protocols" on page 112
  - "Stateful Signatures" on page 115
  - "TCP Stream Signatures" on page 116
  - "Protocol Anomalies" on page 116
  - "Attack Object Groups" on page 117
  - "Disabling Attack Objects" on page 119
- "Attack Actions" on page 120
  - "Brute Force Attack Actions" on page 127
- "Attack Logging" on page 130
- "Mapping Custom Services to Applications" on page 132
- "Customized Attack Objects and Groups" on page 136
  - "User-Defined Stateful Signature Attack Objects" on page 136
  - "TCP Stream Signature Attack Objects" on page 140
  - "Configurable Protocol Anomaly Parameters" on page 142
- "Negation" on page 143

You can also enable DI at the security zone level for HTTP components. These SCREEN options are explained in the final section of this chapter:

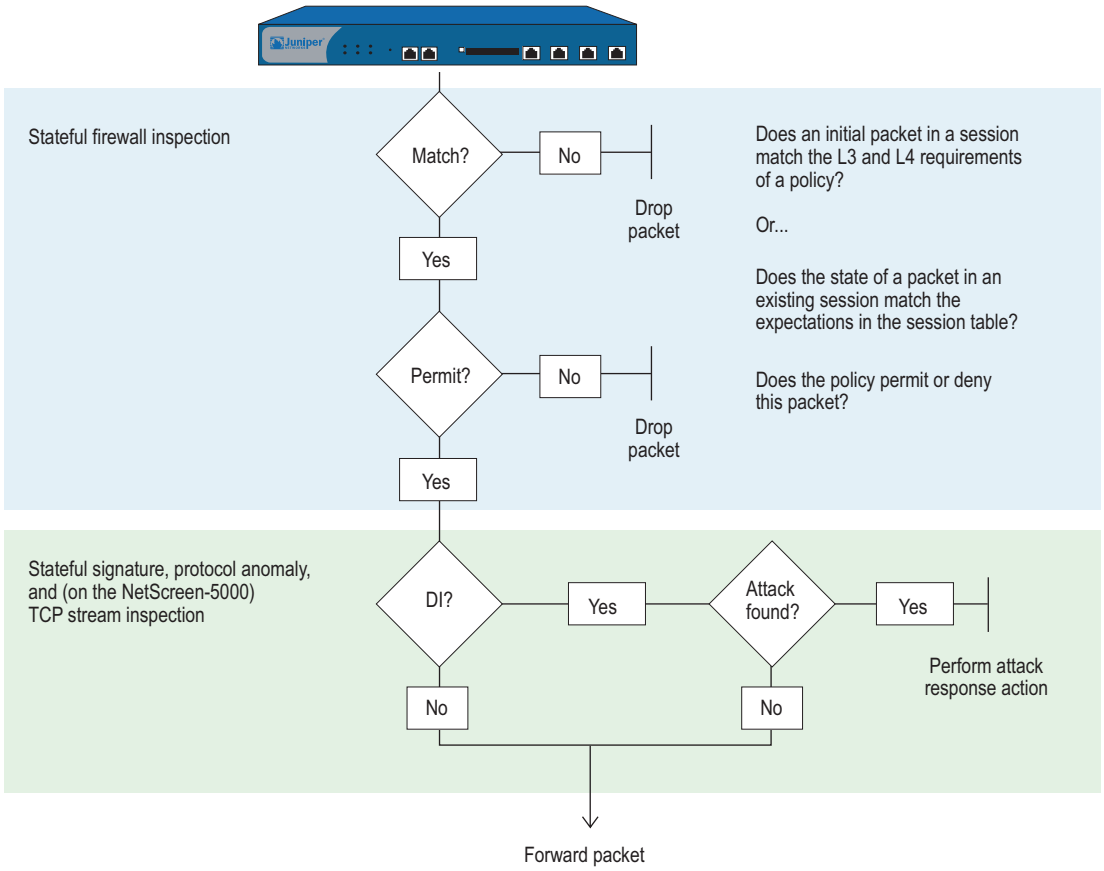
- "Granular Blocking of HTTP Components" on page 147
  - "ActiveX Controls" on page 148
  - "Java Applets" on page 148
  - "EXE Files" on page 148
  - "ZIP Files" on page 148

Overview

Deep Inspection (DI) is a mechanism for filtering the traffic permitted by the Juniper Networks firewall. DI examines Layer 3 and Layer 4 packet headers and Layer 7 application content and protocol characteristics in an effort to detect and prevent any attacks or anomalous behavior that might be present. Figure 42 shows how a packet undergoes Layer 3 inspection.

NOTE: Juniper Networks security devices detect anomalous traffic patterns at Layer 3 and Layer 4 (IP and TCP) via SCREEN options set at the zone level, not the policy level. Examples of IP and TCP traffic-anomaly detection are "IP Address Sweep" on page 8, "Port Scanning" on page 9, and the various flood attacks described in "Network DoS Attacks" on page 34.

Figure 42: Stateful Firewall Inspection



When the security device receives the first packet of a session, it inspects the source and destination IP addresses in the IP packet header (Layer 3 inspection) and the source and destination port numbers and protocol in the TCP segment or UDP datagram header (Layer 4 inspection). If the Layer 3 and 4 components match the criteria specified in a policy, the device then performs the specified action on the packet—permit, deny, or tunnel. When the device receives a packet for an established session, it compares it with the state information maintained in the session table to determine if it belongs to the session.

NOTE: If the specified action is tunnel, the notion of permission is implied. Note that if you enable DI in a policy whose action is tunnel, the security device performs the specified DI operations before encrypting an outbound packet and after decrypting an inbound packet.

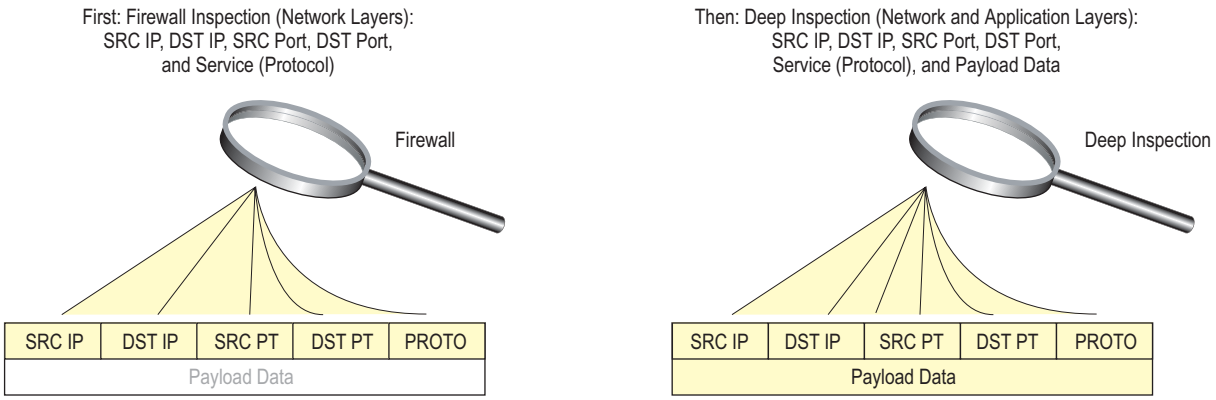
If you have enabled DI in the policy that applies to this packet and the policy action is “permit” or “tunnel,” then the security device further inspects it and its associated data stream for attacks. It scans the packet for patterns that match those defined in one or more groups of attack objects. Attack objects can be attack signatures or protocol anomalies, which you can either define yourself or download to the security device from a database server. (For more information, see “Attack Objects and Groups” on page 110 and “Customized Attack Objects and Groups” on page 136.)

NOTE: The Deep Inspection (DI) feature is available after you have obtained and loaded an advanced mode license key. (If you upgrade from a pre-5.0.0 version of ScreenOS, the mode automatically becomes "advanced." In this case, an advanced-mode license key is not required.)The ability to download signature packs from the database server requires that you first subscribe for the service. For more information, see "Registration and Activation of Subscription Services" on page **2-**253.

Based on the attack objects specified in the policy, the security device might perform the following inspections (see Figure 43):

- Examine header values and payload data for stateful attack signatures
- Compare the format of the transmitted protocol with the standards specified in the RFCs and RFC extensions for that protocol to determine if someone has altered it, possibly for malicious purposes

Figure 43: Firewall Inspection Versus Deep Inspection



If the security device detects an attack, it performs the action specified for the attack object group to which the matching attack object belongs: close, close-client, close-server, drop, drop-packet, ignore, or none. If it does not find an attack, it forwards the packet. (For more information about attack actions, see "Attack Actions" on page 120.)

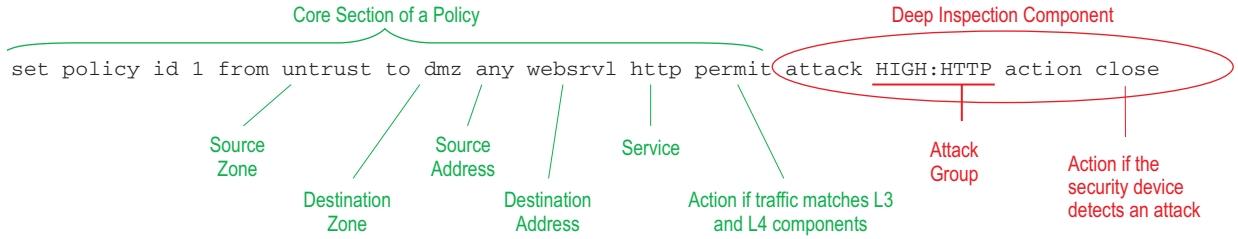
You can conceptually separate a **set policy** command into two parts—the core section and the DI component:

- The core section contains the source and destination zones, source and destination addresses, one or more services, and an action.
- The DI component instructs the security device to inspect traffic permitted by the core section of the policy for patterns matching the attack objects contained in one or more attack object groups. If the security device detects an attack object, it then performs the action defined for the corresponding group.

NOTE: You can optionally add other extensions to the core component of a **set policy** command: VPN and L2TP tunnel references, a schedule reference, address translation specifications, user authentication specifications, antivirus checking, logging, counting, and traffic management settings. Whereas these extensions are optional, the elements that constitute the core of a policy—source and destination zones, source and destination addresses, service (or services), and action—are required. (An exception to this is a global policy, in which no source and destination zones are specified: **set policy global** ***src_addr dst_addr service action***. For more information about global policies, see "Global Policies" on page **2-**166.)

The following **set policy** command includes a DI component:

Figure 44: DI Component in the Set Policy Command



The above command directs the security device to permit HTTP traffic from any address in the Untrust zone to the destination address "websrv1" in the DMZ zone. It also instructs the device to inspect all HTTP traffic permitted by this policy. If any pattern in the traffic matches an attack object defined in the attack object group "HIGH:HTTP:ANOM", the device closes the connection by dropping the packet and sending TCP RST notifications to the hosts at the source and destination addresses.

It is possible to enter the context of an existing policy by using its ID number. For example:

```
ns-> set policy id 1
ns(policy:1)->
```

NOTE: The command prompt changes to signal that the subsequent command will be within a particular policy context.

Entering a policy context is convenient if you want to enter several commands related to a single policy. For example, the following set of commands creates a policy that permits HTTP and HTTPS traffic from the any address in the Untrust to websrv1 and websrv2 in the DMZ zone and looks for high and critical HTTP stateful signature and protocol anomaly attacks:

```
ns-> set policy id 1 from untrust to dmz any websrv1 http permit attack
     CRITICAL:HTTP:ANOM action close
ns-> set policy id 1
ns(policy:1)-> set dst-address websrv2
ns(policy:1)-> set service https
ns(policy:1)-> set attack CRITICAL:HTTP:SIGS action close-server
ns(policy:1)-> set attack HIGH:HTTP:ANOM action drop
ns(policy:1)-> set attack HIGH:HTTP:SIGS action close-server
ns(policy:1)-> exit
ns-> save
```

The above configuration permits both HTTP and HTTPS traffic, but only looks for attacks in HTTP traffic. To be able to add attack object groups within a policy context, you must first specify a DI attack and action in the top-level command. In the above example, you can add CRITICAL:HTTP:SIGS, HIGH:HTTP:ANOM, and HIGH:HTTP:SIGS attack object groups because you first configured the policy for DI with the CRITICAL:HTTP:ANOM group.

NOTE: You can specify a different attack action for each attack object group in a policy. If the security device simultaneously detects multiple attacks, it applies the most severe action, which in the above example is "close." For information about the seven attack actions, including their severity levels, see "Attack Actions" on page 120.

Attack Object Database Server

The attack object database server contains all the predefined attack objects, organized into attack object groups by protocol and severity level. Juniper Networks stores the attack object database on a server at https://services.netscreen.com/restricted/sigupdates.

Predefined Signature Packs

The attack object database is organized into four signature packs, base, server protection, client protection, and worm mitigation. This approach is ideal because of the limited device memory and increased protocol support in the signature packs. Table 3 describes each of the predefined signature packs and the threat coverage.

Table 3: Predefined Signature Packs

Signature Pack	Description	Threat Coverage
Base[1]	A selected set of signatures for client/server and worm protection optimized for remote and branch offices along with small/medium businesses.	Includes a sample of worm, client-to-server, and server-to-client signatures for Internet-facing protocols and services, such as HTTP, DNS, FTP, SMTP, POP3, IMAP, NetBIOS/SMB, MS-RPC, P2P, and IM (AIM, YMSG, MSN, and IRC).
Server protection	For small/medium enterprises and remote and branch offices of large enterprises needing perimeter defense and compliance for server infrastructure, such as IIS, Exchange, and Oracle.	Primarily focuses on protecting a server farm. It includes a comprehensive set of server-oriented protocols, such as HTTP, DNS, FTP, SMTP, IMAP, MS-SQL, Oracle, and LDAP. Also includes worm signatures that target servers.
Client protection	For small/medium enterprises and remote and branch offices of large enterprises needing perimeter defense and compliance for hosts (desktops, laptops, and so on).	Primarily focuses on protecting users from getting malware, Trojans, and so on while surfing the Internet. Includes a comprehensive set of client-oriented protocols, such as HTTP, DNS, FTP, IMAP, POP3, P2P, and IM (AIM, YMSG, MSN, and IRC). Also includes worm signatures that target clients.
Worm Mitigation	For remote and branch offices of large enterprises along with small/medium businesses to provide the most comprehensive defense against worm attacks.	Includes stream signatures and primarily focuses on providing comprehensive worm protection. Detects server-to-client and client-to-server worm attacks for all protocols.

1.Due to memory allocation required for new enhancements, only DI signatures of critical severity are provided for NS-5XT/GT devices.

Updating Signature Packs

Juniper Networks stores the signature packs on a server at https://services.netscreen.com/restricted/sigupdates. To use the predefined attack objects, you must download the signature packs from this server, load it on your security device, and then reference specific attack object groups in policies. To gain access to the attack object database server, you must first subscribe to the DI signature service for your device. (For information on how to do that, see “Registration and Activation of Subscription Services” on page **2-**253.)

There are four ways to update the database:

NOTE: You can also use NetScreen-Security Manager to download the signature packs. For information, see the *NetScreen-Security Manager Administration Guide.*

You can load an authentication certificate (imagekey.cer) for verifying the integrity of the attack object database when downloading it.

- **Immediate Update:** With this option, you update the attack object database on the security device immediately with the database stored on the attack object database server. For this operation to work, you must first configure the attack object database server settings. (For an example, see “Example: Immediate Update” on page 105.)

NOTE: Before performing an immediate database update, you can use the **exec attack-db check** command to check if the attack object database on the server is more recent than the one on the security device.

- **Automatic Update:** With this option, the security device downloads the attack object database at user-scheduled times if the database on the server is a newer version than that previously loaded on the device. Juniper Networks updates the database on a regular basis with newly discovered attack patterns. Therefore, because of its changing nature, you must update the database on your security device regularly too. For this operation to work, you must first configure the attack object database server settings. (For an example, see "Example: Automatic Updates" on page 106.)

- **Automatic Notification and Immediate Update:** With this option, the security device checks at user-scheduled times if the data on the attack object database server is more recent than that on the device. If the data on the server is more recent, a notice appears on the Home page in the WebUI, and in the CLI after you log into the security device. You can then enter the **exec attack-db update** command or click the **Update Now** button on the Configuration > Update > Attack Signature page in the WebUI to save the signature pack from the server to the device. For the server-checking operation semi-automatic procedure to work, you must first configure the attack object database server settings. (For an example, see "Example: Automatic Notification and Immediate Update" on page 107.)

- **Manual Update:** With this option, you first use a browser to download the signature pack to a local directory or TFTP server directory. You can then load the database on the security device using either the WebUI (from the local directory) or CLI (from the TFTP server directory). (For an example, see "Example: Manual Update" on page 108.)

Before You Start Downloading

Before you start downloading a signature pack using any of the four methods described above, you must do the following:

1. Register your security device and obtain an authorization code.
2. Purchase a license key and activate a subscription for Deep Inspection.
3. Verify that the system clock and the Domain Name System (DNS) settings on your device are accurate.

WebUI

Configuration > Date/Time

Network > DNS > Host

4. Click the **Update Now** button.

   Note that this option is only available after you retrieve a Deep Inspection subscription key.

The security device then attempts to contact the server at the default URL: https://services.netscreen.com/restricted/sigupdates; or, if you have entered a different URL in the Database Server field, it attempts to contact the URL that you entered. Table 4 on page 105 lists the predefined signatures packs and the corresponding URLs.

After a few moments, a message appears indicating whether the update was successful. If the update was unsuccessful, then check the event log to determine the cause of the failure.

NOTE: After you download the signature pack the first time, you must reset the security device. Following each download thereafter, resetting the device is unnecessary.

Example: Immediate Update

In this example (see Figure 45), you save a predefined signature pack from the attack object database server to the security device immediately. Table 4 lists the predefined signatures packs and the corresponding URLs.

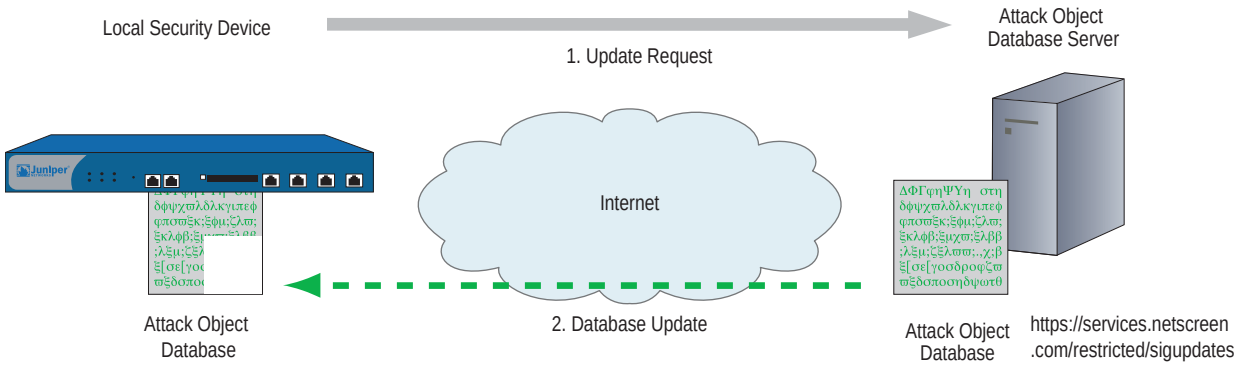
Table 4: URLs for Predefined Signature Packs

To Save the	Specify This URL
Base signature pack (default)	https://services.netscreen.com/restricted/sigupdates The security device uses this URL by default.
Server-protection signature pack	https://services.netscreen.com/restricted/sigupdates/server
Client-protection signature pack	https://services.netscreen.com/restricted/sigupdates/client
Worm-mitigation signature pack	https://services.netscreen.com/restricted/sigupdates/worm

You do not set a schedule for updating the database on the security device. Instead, you save the database from the server to the security device immediately.

NOTE: This example assumes that you have already obtained and activated a subscription for the DI signature service for the security device. (For information about subscriptions, see “Registration and Activation of Subscription Services” on page 2-253.)

Figure 45: Updating DI Signatures Immediately



WebUI

Configuration > Update > Attack Signature: Click the **Update Now** button.

CLI

```
ns-> exec attack-db update
Loading attack database.....
Done.
Done.
Switching attack database...Done
Saving attack database to flash...Done.
ns->
```

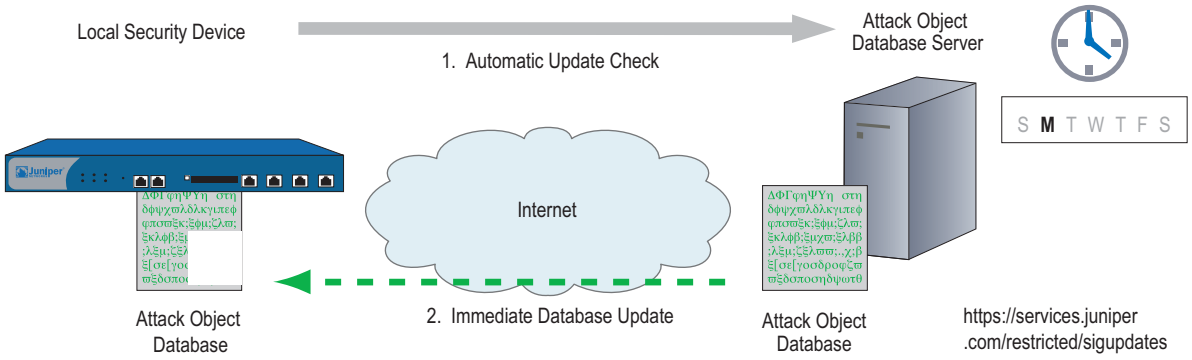
Example: Automatic Updates

In this example (see Figure 46), you set a schedule to update the database on the security device every Monday at 4:00 AM. At that scheduled time, the device compares the version of the database on the server with that on the device. If the version on the server is more recent, the security device automatically replaces its database with the newer version.

NOTE: This example assumes that you have already obtained and activated a subscription for the DI signature service for the security device. (For information about subscriptions, see “Registration and Activation of Subscription Services” on page **2-**253.)

To update the base signature pack, use the default URL: https://services.netscreen.com/restricted/sigupdates.Refer to Table 4 on page 105 for a list of predefined signatures packs and the corresponding URLs.

Figure 46: Updating DI Signatures Automatically



WebUI

Configuration > Update > Attack Signature: Enter the following, then click **OK**:

Database Server: (leave empty)
Update Mode: Automatic Update
Schedule:
Weekly on: Monday
Time (hh:mm): 04:00

NOTE: If you schedule updates on a monthly basis and the date you choose does not occur in a month (for example, 31 does not occur in several months), the security device uses the last possible date of the month in its place.

CLI

```
set attack db mode update
set attack db schedule weekly monday 04:00
save
```

Example: Automatic Notification and Immediate Update

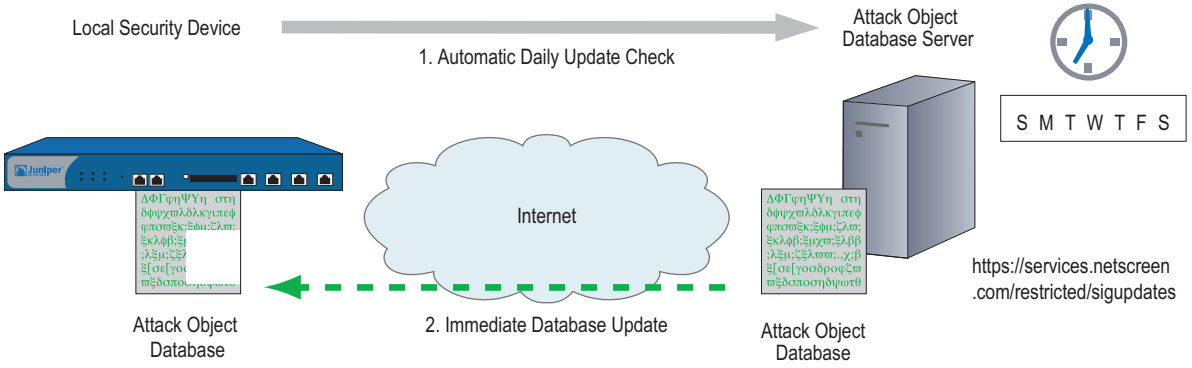
In this example (see Figure 47), you set a schedule to check the database on the security device every day at 07:00 AM.

When you receive a notice that the database on the server has been updated, you click the **Update Now** button on the Configuration > Update > Attack Signature page in the WebUI or enter the **exec attack-db update** command to save the database from the server to the device.

NOTE: This example assumes that you have already obtained and activated a subscription for the DI signature service for the security device. (For information about subscriptions, see “Registration and Activation of Subscription Services” on page 2-253.)

To update the base signature pack, use the default URL: https://services.netscreen.com/restricted/sigupdates. Refer to Table 4 on page 105 for a list of predefined signatures packs and the corresponding URLs.

Figure 47: Notifying Signature Updates



WebUI

1. **Scheduled Database Checking**
   Configuration > Update > Attack Signature: Enter the following, then click **OK**:

   Database Server: (leave empty)
   Update Mode: Automatic Notification
   Schedule:
   Daily
   Time (hh:mm): 07:00

2. **Immediate Database Update**
   When you receive a notice that the attack database on the server is more current than the one on the security device, do the following:

   Configuration > Update > Attack Signature: Click the **Update Now** button.

CLI

1. **Scheduled Database Checking**
   set attack db mode notification
   set attack db schedule daily 07:00

2. **Immediate Database Update**
   When you receive a notice that the attack database on the server is more current than the one on the security device, do the following:

   exec attack-db update

Example: Manual Update

In this example (see Figure 48), you manually save the latest signature pack to the local directory "C:\netscreen\attacks-db" (if you want to use the WebUI to load the database) or C:\Program Files\TFTP Server (if you want to use the CLI to load it). You then load the database on the security device from your local directory.

NOTE: After downloading the signature pack, you can also post it on a local server and set it up for other security devices to access. The admins for the other devices must then change the database server URL to that of the new location. They can either enter the new URL in the Database Server field on the Configuration > Update > Attack Signature page or use the following CLI command: **set attack db server** *url_string*.

For an automatic update, the security device automatically adds the following elements to the URL:

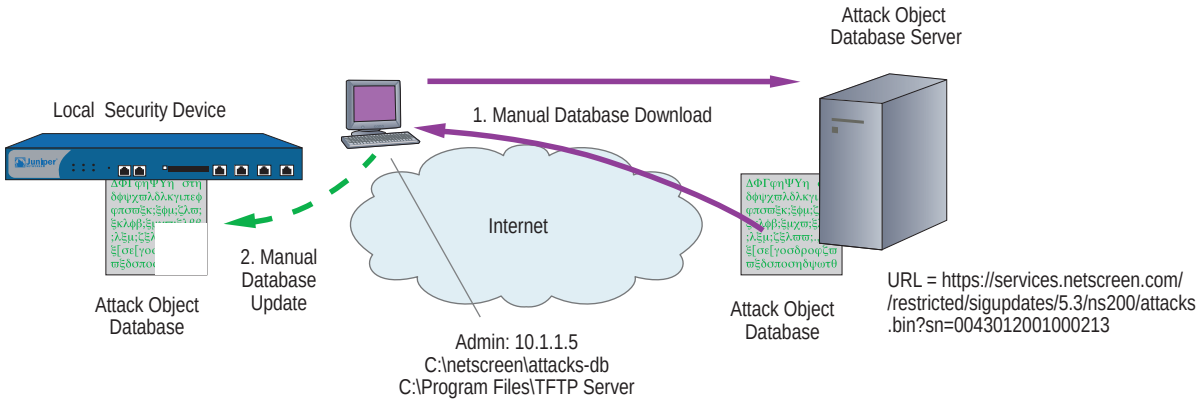
- Serial number of the security device
- Number of the major ScreenOS version running on the device
- Platform type

When you manually update the DI Signatures, you must add these elements yourself. In this example, the serial number is 0043012001000213, the ScreenOS version is 5.3, and the platform is NetScreen-208 (ns200). Consequently, the resulting URL is:

https://services.netscreen.com/restricted/sigupdates/5.3/ns200/attacks.bin?sn=0043012001000213

NOTE: This example assumes that you have already obtained and activated a subscription for the DI signature service for the security device. (For information about subscriptions, see "Registration and Activation of Subscription Services" on page **2**-253.)

Figure 48: Updating DI Signatures Manually



1. Downloading the Signature Pack

To update the base signature pack, use the default URL: https://services.netscreen.com/restricted/sigupdates. Refer to Table 4 on page 105 for a list of predefined signatures packs and the corresponding URLs.

Enter the following URL in the address field of your browser:

```
https://services.netscreen.com/restricted/sigupdates/5.3/ns200/attacks.bin?sn=0043012001000213
```

Save *attacks.bin* to the local directory "C:\netscreen\attacks-db" (for loading via the WebUI) or to your TFTP server directory C:\Program Files\TFTP Server (when you want to use the CLI to load it).

2. Updating the Signature Pack

WebUI

Configuration > Update > Attack Signature: Enter the following, then click **OK**:

Deep Inspection Signature Update:
Load File: Enter **C:\netscreen\attacks-db\attacks.bin**

Or

Click **Browse** and navigate to that directory, select **attacks.bin**, then click **Open**.

If you downloaded the server, client, or worm protection signature packs, then enter the appropriate filename.

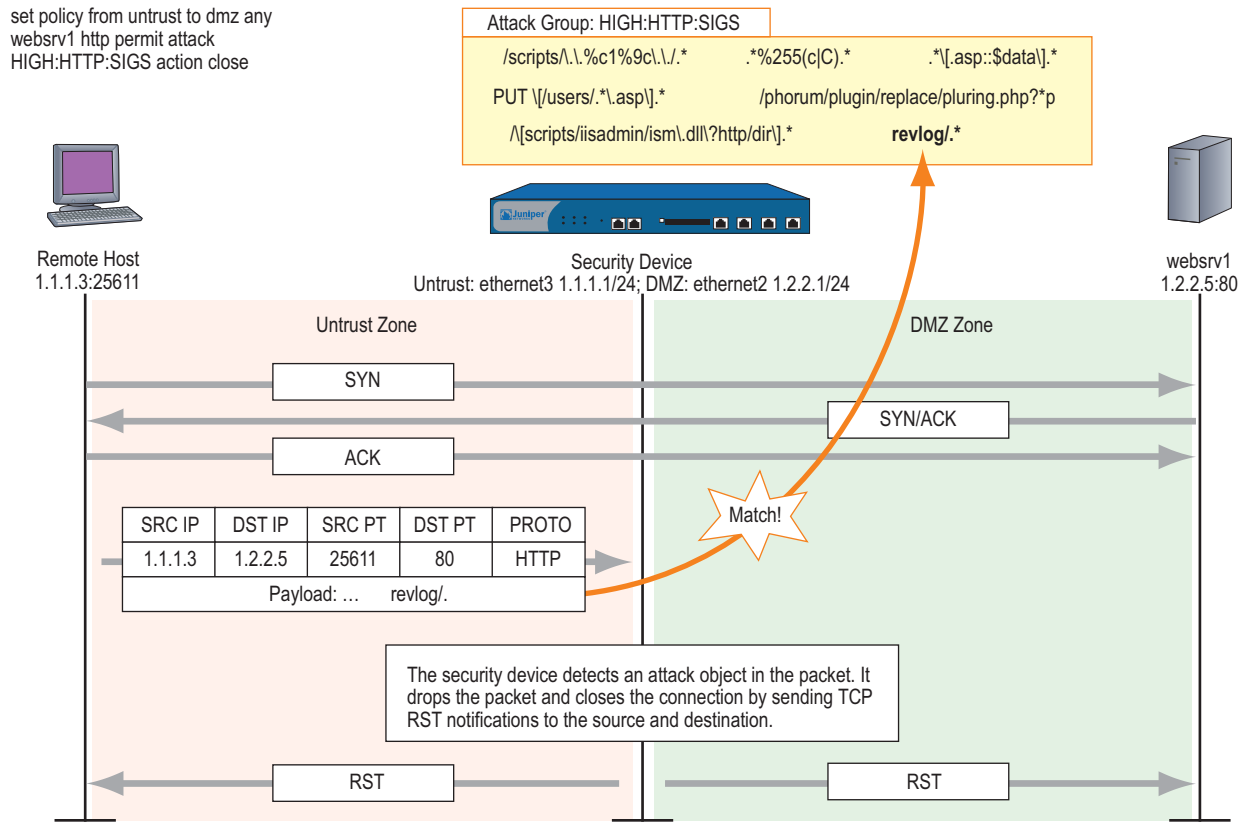
CLI

```
save attack-db from tftp 10.1.1.5 attacks.bin to flash
```

Attack Objects and Groups

Attack objects are stateful signatures, stream signatures (on the NetScreen-5000 series), and protocol anomalies that a security device uses to detect attacks aimed at compromising one or more hosts on a network. Attack objects are in groups organized by protocol type and then by severity. When you add Deep Inspection (DI) to a policy, the device inspects the traffic that the policy permits for any patterns matching those in the referenced attack object group (or groups).

Figure 49: Attack Objects and Groups



The attack object groups that you reference in the DI component of a policy must target the same service type that the policy permits. For example, if the policy permits SMTP traffic, the attack object group must aim at attacks on SMTP traffic. The following policy exemplifies a valid configuration:

set policy id 2 from trust to untrust any any **smtp** permit attack **CRIT:SMTP:SIGS** action close

The next policy is erroneous because the policy permits SMTP traffic, but the attack object group is for POP3 traffic:

set policy id 2 from trust to untrust any any **smtp** permit attack **CRIT:POP3:SIGS** action close

The second policy is configured incorrectly and, if implemented, would cause the security device to expend unnecessary resources inspecting SMTP traffic for POP3 attack objects that it could never find. If policy 2 permits both SMTP and POP3 traffic, you can configure the DI component to check for SMTP attack objects, POP3 attack objects, or for both.

set group service **grp1**
set group service **grp1** add **smtp**
set group service **grp1** add **pop3**
set policy id 2 from trust to untrust any any **grp1** permit attack **CRIT:SMTP:SIGS** action close
set policy id 2 attack **CRIT:POP3:SIGS** action close

Supported Protocols

The Deep Inspection (DI) module supports stateful signature attack objects and protocol anomaly attack objects for the following protocols and applications:

Table 5: Basic Network Protocols

Protocol	Stateful Signature	Protocol Anomaly	Definition
DNS	Yes	Yes	Domain Name System (DNS) is a database system for translating domain names to IP addresses, such as www.juniper.net = 207.17.137.68.
FTP	Yes	Yes	File Transfer Protocol (FTP) is a protocol for exchanging files between computers across a network.
HTTP	Yes	Yes	HyperText Transfer Protocol (HTTP) is a protocol primarily used to transfer information from webservers to web clients.
IMAP	Yes	Yes	Internet Mail Access Protocol (IMAP) is a protocol that provides incoming e-mail storage and retrieval services, with the option that users can either download their e-mail or leave it on the IMAP server.
NetBIOS	Yes	Yes	NetBIOS (Network Basic Input Output System) is an application interface that allows applications on users' workstations to access network services provided by network transports such as NetBEUI, SPX/IPX, and TCP/IP.
POP3	Yes	Yes	Post Office Protocol, version 3 (POP3) is a protocol that provides incoming e-mail storage and retrieval services.
SMTP	Yes	Yes	Simple Mail Transfer Protocol (SMTP) is a protocol for transferring e-mail between mail servers.
Chargen	Yes	Yes	Character generator protocol
DHCP	Yes	Yes	Dynamic Host Configuration Protocol is used to control vital networking parameters of hosts (running clients) with the help of a server. DHCP is backward compatible with BOOTP.
Discard	Yes	Yes	Discard protocol is a useful debugging and measurement tool. A discard service simply throws away any data it receives.
Echo	Yes	Yes	Echo protocol is an internet protocol intended for testing and measurement purposes. A host may connect to a server that supports the ECHO protocol, on either TCP or UDP port 7. The server then sends back any data it receives.
Finger	Yes	Yes	Finger User Information protocol is a simple protocol that provides an interface to a remote user information program.
Gopher	Yes	Yes	Gopher is an internet protocol designed for distributed document search and retrieval.
ICMP	Yes	Yes	Internet Control Message Protocol is a required protocol tightly integrated with IP. ICMP messages, delivered in IP packets, are used for out-of-band messages related to network operation.
IDENT	Yes	Yes	Identification protocol provides a means to determine the identity of a user of a particular TCP connection.
LDAP	Yes	Yes	Lightweight Directory Access Protocol is a set of protocols for accessing information directories.
LPR	Yes	Yes	Line Printer spooler
NFS	Yes	Yes	Network File System (NFS) protocol provides transparent remote access to shared files across networks. The NFS protocol is designed to be portable across different machines, operating systems, network architectures, and transport protocols.

Protocol	Stateful Signature	Protocol Anomaly	Definition
NNTP	Yes	Yes	Network News Transfer Protocol specifies a protocol for the distribution, inquiry, retrieval, and posting of news articles using a reliable stream-based transmission of news.
NTP	Yes	Yes	Network Time Protocol and Simple Network Time Protocol is used to synchronize the time of a computer client or server to another server or reference time source, such as a radio or satellite receiver or modem.
Portmapper	Yes	Yes	Port Mapper Program Protocol maps RPC program and version numbers to transport- specific port numbers.
RADIUS	Yes	Yes	Remote Authentication Dial In User Service, an authentication and accounting system used by many Internet Service Providers (ISPs).
rexec	Yes	Yes	Remote Execution
rlogin	Yes	Yes	Remote Login occurs when a user connects to an Internet host to use its native user interface.
rsh	Yes	Yes	Remote shell
RTSP	Yes	Yes	Real Time Streaming Protocol is a client-server application-level protocol for controlling the delivery of data with real-time properties. It establishes and controls either a single or several time-synchronized streams of continuous media, such as audio and video.
SNMPTRAP	Yes	Yes	Simple Network Management Protocol is an SNMP application that uses the SNMP TRAP operation to send information to a network manager.
SSH	Yes	Yes	Secure Shell Protocol is a protocol for secure remote login and other secure network services over an insecure network.
SSL	Yes	Yes	Secure Sockets Layer is a protocol used for transmitting private documents via the Internet using a cryptographic system.
syslog	Yes	Yes	System Logging Protocol is used for the transmission of event notification messages across networks.
Telnet	Yes	Yes	Telnet protocol is a terminal emulation program for TCP/IP networks. This protocol enables you to communicate with other servers on the network.
TFTP	Yes	Yes	Trivial File Transfer Protocol is a simple protocol used to transfer files. TFTP uses the User Datagram Protocol (UDP) and provides no security features.
VNC	Yes	Yes	Virtual Network Computing is a desktop protocol to remotely control another computer.
Whois	Yes	Yes	Network Directory Service Protocol is a TCP transaction based query/response server that provides network-wide directory service to internet users.

Table 6: Instant Messaging Applications

Protocol	Stateful Signature	Protocol Anomaly	Definition
AIM	Yes	Yes	America Online Instant Messaging (AIM) is the instant messaging application for America Online.
MSN Messenger	Yes	Yes	Microsoft Network Messenger (MSN Messenger) is the instant messaging service provided by Microsoft.
Yahoo! Messenger	Yes	Yes	Yahoo! Messenger is the instant messaging service provided by Yahoo!.
IRC	Yes	Yes	Internet Relay Chat is a text-based protocol, with the simplest client being any socket program capable of connecting to the server.

Table 7: Peer-to-Peer (P2P) Networking Applications

Protocol	Stateful Signature	Protocol Anomaly	Definition
BitTorrent	Yes	No	BitTorrent is a P2P file distribution tool, designed to provide an efficient way to distribute the same file to a large group by having everybody that downloads a file also upload it to others.
DC (Direct Connect)	Yes	No	DC (Direct Connect) is a P2P file-sharing application. A DC network uses hubs to connect groups of users, often with a requirement that they share a certain amount of bytes or files. Many hubs feature special areas of interest, creating small communities for connected users.
eDonkey	Yes	No	eDonkey is a decentralized P2P file-sharing application that uses the Multisource File Transfer Protocol (MFTP). The eDonkey network supports two kinds of applications: clients and servers. Clients connect to the network and share files. Servers act as meeting hubs for the clients.
Gnutella	Yes	Yes	Gnutella is a P2P file-sharing protocol and application without any centralized servers. Some other applications using the Gnutella protocol are BearShare, Limewire, Morpheus, and ToadNode.
KaZaa	Yes	No	KaZaa is a decentralized P2P file-sharing application using the FastTrack protocol. KaZaa is mainly used for sharing MP3 files.
MLdonkey	Yes	No	MLdonkey is a P2P client application that can run on multiple platforms and can access multiple P2P networks, such as BitTorrent, DC, eDonkey, FastTrack (KaZaa and others), and Gnutella and Gnutella2.
Skype	Yes	No	Skype is a free P2P Internet telephony service that allows users to talk with each other over a TCP/IP network such as the Internet.
SMB	Yes	Yes	SMB (Server Message Block) is a protocol for sharing such resources as files and printers among computers. SMB operates on top of the NetBIOS protocol.
WinMX	Yes	No	WinMX is a P2P file-sharing application that allows a client to connect to several servers simultaneously

NOTE: Many of the listed P2P applications use their own proprietary protocols.

Table 8: Application Layer Gateways (ALGs)

Protocol	Stateful Signature	Protocol Anomaly	Definition
MSRPC	Yes	Yes	MSRPC (Microsoft-Remote Procedure Call) is a mechanism for running processes on a remote computer.

If the security device has access to http://help.juniper.net/sigupdates/english, you can see the contents of all the predefined attack object groups and descriptions of the predefined attack objects. Open your browser, and enter one of the following URLs in the Address field:

http://help.juniper.net/sigupdates/english/AIM.html
http://help.juniper.net/sigupdates/english/DNS.html
http://help.juniper.net/sigupdates/english/FTP.html
http://help.juniper.net/sigupdates/english/GNUTELLA.html
http://help.juniper.net/sigupdates/english/HTTP.html
http://help.juniper.net/sigupdates/english/IMAP.html
http://help.juniper.net/sigupdates/english/MSN.html
http://help.juniper.net/sigupdates/english/NBDS.html

http://help.juniper.net/sigupdates/english/NBNAME.html
http://help.juniper.net/sigupdates/english/POP3.html
http://help.juniper.net/sigupdates/english/SMTP.html
http://help.juniper.net/sigupdates/english/MSRPC.html
http://help.juniper.net/sigupdates/english/SMB.html
http://help.juniper.net/sigupdates/english/YMSG.html

Each of the above URLs links to an HTML page containing a list of all the predefined attack objects—organized in groups by severity—for a particular protocol. To see a description of an attack object, click its name.

Stateful Signatures

An attack signature is a pattern that exists when a particular exploit is in progress. The signature can be a Layer 3 or 4 traffic pattern, such as when one address sends lots of packets to different port numbers at another address (see "Port Scanning" on page 9), or a textual pattern, such as when a malicious URL string appears in the data payload of a single HTTP or FTP packet. The string can also be a specific segment of code or a specific value in the packet header. However, when searching for a textual pattern, the Deep Inspection (DI) module in a security device looks for more than just a signature in a packet; it looks for the signature in a particular portion of the packet (even if fragmented or segmented), in packets sent at a particular time in the life of the session, and sent by either the connection initiator or the responder.

NOTE: Because the DI module supports regular expressions, it can use wildcards when searching for patterns. Thus, a single attack signature definition can apply to multiple attack pattern variations. For information about regular expressions, see "Regular Expressions" on page 137.

When the DI module checks for a textual pattern, it considers the roles of the participants as client or server and monitors the state of the session to narrow its search to just those elements relevant to the exploit for which attackers use the pattern. Using contextual information to refine packet examination greatly reduces false alarms—or "false positives"—and avoids unnecessary processing. The term "stateful signatures" conveys this concept of looking for signatures within the context of the participants' roles and session state.

To see the advantage of considering the context in which a signature occurs, note the way the DI module examines packets when enabled to detect the EXPN Root attack. Attackers use the EXPN Root attack to expand and expose mailing lists on a mail server. To detect the EXPN Root attack, the security device searches for the signature "expn root" in the control portion of a Simple Mail Transfer Protocol (SMTP) session. The device examines only the control portion because that is only where the attack can occur. If "expn root" occurs in any other portion of the session, it is not an attack.

Using a simple textual packet signature detection technique, the signature "expn root" triggers an alarm even if it appears in the data portion of the SMTP connection; that is, in the body of an e-mail message. If, for example, you were writing to a colleague about EXPN Root attacks, a simple packet signature detector would regard this as an attack. Using stateful signatures, the DI module can distinguish between text strings that signal attacks and those that are harmless occurrences.

NOTE: For a list of protocols for which there are predefined stateful signature attack objects, see “Supported Protocols” on page 112.

TCP Stream Signatures

Like a stateful signature, a TCP stream signature is a pattern that exists when an exploit is in progress. However, when the DI module examines traffic for stateful signatures, it searches only within specific contexts. When the DI module examines traffic for TCP stream signatures, it does so without regard for contexts. Another distinction between the two types of signatures is that although stateful signatures can be predefined or user-defined, TCP stream signatures must be user-defined. After you add a stream signature attack object to an attack object group, you can then reference that group in a policy that applies DI. (For more about TCP stream signatures, see “TCP Stream Signature Attack Objects” on page 140.)

NOTE: You can define TCP stream signatures on NetScreen-5000 series systems only.

Stream signatures are independent of protocols and are therefore more flexible in matching traffic. Stream signatures can examine traffic where protocols decoders can’t inspect. However, this flexibility affects performance and resource consumption.

Stream signatures consume resources and affect performance, so they must be used sparingly. Stream256 signatures however, operate the same way, but rather than matching over the entire stream, they only match on the first 256 bytes of the stream. Therefore, they consume fewer resources and are less of a performance hit.

Protocol Anomalies

Attack objects that search for protocol anomalies detect traffic that deviates from the standards defined in RFCs and common RFC extensions. With signature attack objects, you must use a predefined pattern or create a new one; therefore, they can only detect known attacks. Protocol anomaly detection is particularly useful for catching new attacks or those attacks that cannot be defined by a textual pattern.

NOTE: For a list of protocols for which there are predefined protocol anomaly attack objects, see “Supported Protocols” on page 112.

Attack Object Groups

Predefined attack object groups contain attack objects for a specific protocol. For each protocol, the groups are separated into protocol anomalies and stateful signatures, and then roughly organized by severity. The three attack object group severity levels are critical, high, and medium:

- **Critical:** Contains attack objects matching exploits that attempt to evade detection, cause a network device to crash, or gain system-level privileges.
- **High:** Contains attack objects matching exploits that attempt to disrupt a service, gain user-level access to a network device, or activate a Trojan horse previously loaded on a device.
- **Medium:** Contains attack objects matching exploits that detect reconnaissance efforts attempting to access vital information through directory traversal or information leaks.
- **Low:** Contains attack objects matching exploits that attempt to obtain non-critical information or scan a network with a scanning tool.
- **Info:** Contains attack objects matching normal, harmless traffic containing URLs, DNS lookup failures, SNMP public community strings, and Peer-to-Peer (P2P) parameters. You can use informational attack objects to obtain information about your network.

Changing Severity Levels

Although attack object groups are classified by protocol and severity level (critical, high, medium), each attack object has its own specific severity level: critical, high, medium, low, info. These attack object severity levels map to severity levels for event log entries as follows:

Table 9: Attack Object Severity Levels

Attack Object Severity Level	– Maps to –	Event Log Entry Severity Level
Critical		Critical
High		Error
Medium		Warning
Low		Notification
Info		Information

For example, if the security device detects an attack with the severity level "Medium," the corresponding entry that appears in the event log then has the severity level "Warning."

It is possible to override the default severity level of all attack objects in one or more attack object groups referenced in a policy. You do this at the policy level by entering the context of an existing policy and then assigning a new severity level to all the attack object groups that the policy references.

The following shows how to change the severity level of the attack object groups referenced in a policy through the WebUI and CLI:

WebUI

Policies > Edit (for an existing policy): Do the following, then click **OK**:

> Deep Inspection: Select a severity option in the Severity drop-down list, then click **OK**.

CLI

```
ns-> set policy id number
ns(policy:number)-> set di-severity { info | low | medium | high | critical }
```

To return the severity level for each attack object to its original setting, you again enter the context of a policy and do either of the following:

WebUI

Policies > Edit (for an existing policy): Do the following, then click **OK**:

> Deep Inspection: Select **Default** in the Severity drop-down list, then click **OK**.

CLI

```
ns-> set policy id number
ns(policy:number)-> unset di-severity
```

Example: Deep Inspection for P2P

In this example, you permit any host in the Trust zone to initiate a peer-to-peer (P2P) session with any host in the Untrust zone using HTTP, DNS, and Gnutella services. You then apply Deep Inspection (DI) to the permitted traffic to check for stateful signatures and protocol anomalies as defined in the following attack object groups:

- INFO:DNS:SIGS
- INFO:GNUTELLA:ANOM
- INFO:HTTP:SIGS

NOTE: For security reasons, you do not define a policy permitting any host in the Untrust zone to initiate a P2P session with a host in the Trust zone.

If the security device detects a signature or anomalous behavior, it severs the connection and sends a TCP RST to the client to close the session. You also enable the logging of any discovered attack, which is the default behavior.

NOTE: For information about the various attack actions that the security device can perform, see “Attack Actions” on page 120. For information about logging detected attacks, see “Attack Logging” on page 130.

WebUI

Policies > (From: Trust, To: Untrust) New: Enter the following, then click **OK**:

Source Address:
Address Book Entry: (select), Any
Destination Address:
Address Book Entry: (select), Any
Service: DNS

> Click **Multiple**, select **GNUTELLA** and **HTTP**, then click **OK** to return to the basic policy configuration page.

Action: Permit

> Click **Deep Inspection**, enter the following, click **Add** to enter each attack object group, then click **OK** to return to the basic policy configuration page:

Severity: Default
Group: INFO:DNS:SIGS
Action: Close Client
Log: (select)
Severity: Default
Group: INFO:GNUTELLA:ANOM
Action: Close Client
Log: (select)
Severity: Default
Group: INFO:HTTP:SIGS
Action: Close Client
Log: (select)

CLI

```
set policy id 1 from trust to untrust any any dns permit attack
INFO:DNS:SIGS action close-client
set policy id 1
ns(policy:1)-> set service gnutella
ns(policy:1)-> set service http
ns(policy:1)-> set attack INFO:GNUTELLA:ANOM action close-client
ns(policy:1)-> set attack INFO:HTTP:SIGS action close-client
ns(policy:1)-> exit
save
```

NOTE: Because the logging of detected attacks is enabled by default, you do not have to specify logging through CLI commands.

Disabling Attack Objects

When you reference an attack object group in a policy, the security device checks the traffic to which the policy applies for patterns matching any of the attack objects in that group. At some point, you might not want to use a particular attack object if it repeatedly produces false-positives; that is, if it erroneously interprets legitimate traffic as an attack. If the problem stems from a custom attack object, you can simply remove it from its custom attack object group. However, you cannot remove a predefined attack object from a predefined attack object group. In that case, the best course of action is to disable the object.

Note that a predefined attack object is disabled only within the root system or virtual system (vsys) in which you disable it. For example, disabling a predefined attack object in the root system does not automatically disable it in any virtual systems. Likewise, disabling an attack object in one vsys does not affect that object in any other vsys.

NOTE: Disabling attack objects does not improve throughput performance.

To disable an attack object, do either of the following:

WebUI

Objects > Attacks > Predefined: Clear the checkbox in the **Configure** column for the attack object that you want to disable.

CLI

set attack disable *attack_object_name*

To re-enable a previously disabled attack object, do either of the following:

WebUI

Objects > Attacks > Predefined: Select the checkbox in the **Configure** column for the attack object that you want to enable.

CLI

unset attack disable *attack_object_name*

Attack Actions

When the security device detects an attack, it performs the action that you specify for the attack group containing the object that matches the attack. The seven actions are as follows, from most to least severe:

- **Close** (severs connection and sends RST to client and server)

NOTE: The client is always the initiator of a session; that is, the source address in a policy. The server is always the responder, or the destination address.

Use this option for TCP connections. The security device drops the connection and sends a TCP RST to both the client (source) and server (destination). Because the delivery of RST notifications is unreliable, by sending a RST to both client and server, there is a greater chance that at least one gets the RST and closes the session.

- **Close Server** (severs connection and sends RST to server)

Use this option for inbound TCP connections from an untrusted client to a protected server. If the client tries to launch an attack, the security device drops the connection and sends a TCP RST only to the server for it to clear its resources while the client is left hanging.

- **Close Client** (severs connection and sends RST to client)

  Use this option for outbound TCP connections from a protected client to an untrusted server. If, for example, the server sends a malicious URL string, the security device drops the connection and sends a RST only to the client for it to clear its resources while the server is left hanging.

- **Drop** (severs connection without sending anyone a RST)

  Use this option for UDP or other non-TCP connections, such as DNS. The security device drops all packets in a session, but does not send a TCP RST.

- **Drop Packet** (drops a particular packet, but does not sever connection)

  This option drops the packet in which an attack signature or protocol anomaly occurs but does not terminate the session itself. Use this option to drop malformed packets without disrupting the entire session. For example, if the security device detects an attack signature or protocol anomaly from an AOL proxy, dropping everything would disrupt all AOL service. Instead, dropping just the packet stops the problem packet without stopping the flow of all the other packets.

- **Ignore** (after detecting an attack signature or anomaly, the security device makes a log entry and stops checking—or ignores—the remainder of the connection)

  If the security device detects an attack signature or protocol anomaly, it makes an event log entry but does not sever the session itself. Use this option to tweak false positives during the initial setup phase of your Deep Inspection (DI) implementation. Also, use this option when a service uses a standard port number for nonstandard protocol activities; for example, Yahoo Messenger uses port 25 (SMTP port) for non-SMTP traffic. The security device logs the occurrence once per session (so that it does not fill the log with false positives), but takes no action.

- **None** (no action)

  It is useful when first identifying attack types during the initial setup phase of your DI implementation. When the security device detects an attack signature or protocol anomaly, it makes an entry in the event log but takes no action on the traffic itself. The security device continues to check subsequent traffic in that session and make log entries if it detects other attack signatures and anomalies.

You can create a policy referencing multiple attack object groups, each group having a different action. If the security device simultaneously detects multiple attacks that belong to different attack object groups, it applies the most severe action specified by one of those groups.

Example: Attack Actions—Close Server, Close, Close Client

In this example, there are three zones: Trust, Untrust, and DMZ. You have finished analyzing attacks and have concluded you need the following three policies:

- **Policy ID 1:** Permit HTTP, HTTPS, PING, and FTP-GET traffic from any address in the Untrust zone to the webservers (websrv1 and websrv2) in the DMZ.

  Attack Settings for Policy ID 1:

  - CRITICAL:HTTP:ANOM, CRITICAL:HTTP:SIGS
  - HIGH:HTTP:ANOM, HIGH:HTTP:SIGS
  - MEDIUM:HTTP:ANOM, MEDIUM:HTTP:SIGS
  - CRITICAL:FTP:SIGS
  - Action for all attack object groups: Close Server
  - Logging: Enabled (default setting)

  You choose to drop the connection and send a TCP RST notification only to the protected webservers so they can terminate sessions and clear resources. You anticipate attacks coming from the Untrust zone.

- **Policy ID 2:** Permit HTTP, HTTPS, PING, and FTP traffic from any address in the Trust zone to the webservers (websrv1 and websrv2) in the DMZ

  Attack Settings for Policy ID 2:

  - CRITICAL:HTTP:ANOM, CRITICAL:HTTP:SIGS
  - HIGH:HTTP:ANOM, HIGH:HTTP:SIGS
  - MEDIUM:HTTP:ANOM, MEDIUM:HTTP:SIGS
  - CRITICAL:FTP:SIGS
  - Action for all attack object groups: Close
  - Logging: Enabled (default setting)

  You choose to drop the connection and send a TCP RST notification to both the protected clients and servers so they both can terminate their sessions and clear their resources regardless of the severity level of the attack.

- **Policy ID 3:** Permit FTP-GET, HTTP, HTTPS, PING traffic from any address in the Trust zone to any address in the Untrust zone

  Attack Settings for Policy ID 3:

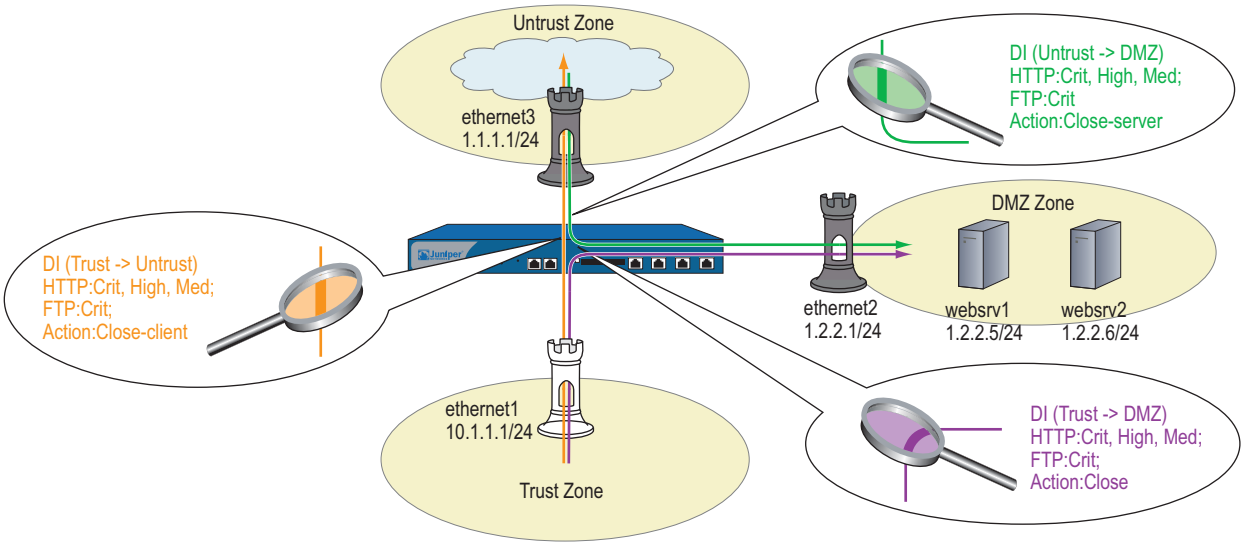
  - CRITICAL:HTTP:ANOM, CRITICAL:HTTP:SIGS
  - HIGH:HTTP:ANOM, HIGH:HTTP:SIGS
  - MEDIUM:HTTP:ANOM, MEDIUM:HTTP:SIGS
  - CRITICAL:FTP:SIGS
  - Action for all attack object groups: Close Client

- Logging: Enabled (default setting)

You choose to drop the connection and send a TCP RST notification to the protected clients so they both can terminate their sessions and clear their resources. In this case, you anticipate an attack coming from an untrusted HTTP or FTP server.

Although the policies permit HTTP, HTTPS, Ping, and FTP-Get or FTP, the security device activates DI only for HTTP and FTP traffic. All zones are in the trust-vr routing domain.

Figure 50: DI Attack Actions



WebUI

1. **Interfaces**

   Network > Interfaces > Edit (for ethernet1): Enter the following, then click **Apply**:

   Zone Name: Trust
   Static IP: (select this option when present)
   IP Address/Netmask: 10.1.1.1/24

   Enter the following, then click **OK**:

   Interface Mode: NAT
   Service Options:
   Management Services: (select all)
   Other services: Ping

   Network > Interfaces > Edit (for ethernet3): Enter the following, then click **OK**:

   Zone Name: Untrust
   Static IP: (select this option when present)
   IP Address/Netmask: 1.1.1.1/24

Network > Interfaces > Edit (for ethernet2): Enter the following, then click **OK**:

Zone Name: DMZ
Static IP: (select this option when present)
IP Address/Netmask: 1.2.2.1/24

2. **Addresses**
   Objects > Addresses > List > New: Enter the following, then click **OK**:

   Address Name: websrv1
   IP Address/Domain Name:
   IP/Netmask: (select), 1.2.2.5/32
   Zone: DMZ

   Objects > Addresses > List > New: Enter the following, then click **OK**:

   Address Name: websrv2
   IP Address/Domain Name:
   IP/Netmask: (select), 1.2.2.6/32
   Zone: DMZ

3. **Route**
   Network > Routing > Routing Entries > trust-vr New: Enter the following, then click **OK**:

   Network Address/Netmask: 0.0.0.0/0
   Gateway: (select)
   Interface: ethernet3
   Gateway IP Address: 1.1.1.250

4. **Policy ID 1**
   Policies > (From: Untrust, To: DMZ) New: Enter the following, then click **OK**:

   Source Address:
   Address Book Entry: (select), Any
   Destination Address:
   Address Book Entry: (select), websrv1

   > Click **Multiple**, select **websrv2**, then click **OK** to return to the basic policy configuration page.

   Service: HTTP

   > Click **Multiple**, select **FTP-GET**, **HTTPS**, **PING**, then click **OK** to return to the basic policy configuration page.

   Action: Permit

   > Click **Deep Inspection**, enter the following, click **Add** to enter each attack object group, then click **OK** to return to the basic policy configuration page:

   Group: CRITICAL:HTTP:ANOM
   Action: Close Server
   Log: (select)
   Group: CRITICAL:HTTP:SIGS
   Action: Close Server

Log: (select)
Group: HIGH:HTTP:ANOM
Action: Close Server
Log: (select)
Group: HIGH:HTTP:SIGS
Action: Close Server
Log: (select)
Group: MEDIUM:HTTP:ANOM
Action: Close Server
Log: (select)
Group: MEDIUM:HTTP:SIGS
Action: Close Server
Log: (select)
Group: CRITICAL:FTP:SIGS
Action: Close Server
Log: (select)

5. Policy ID 2

Policies > (From: Trust, To: DMZ) New: Enter the following, then click **OK**:

Source Address:
Address Book Entry: (select), Any
Destination Address:
Address Book Entry: (select), websrv1

> Click **Multiple**, select **websrv2**, then click **OK** to return to the basic policy configuration page.

Service: HTTP

> Click **Multiple**, select **FTP-GET**, **HTTPS**, **PING**, then click **OK** to return to the basic policy configuration page.

Action: Permit

> Click **Deep Inspection**, enter the following, click **Add** to enter each attack object group, then click **OK** to return to the basic policy configuration page:

Group: CRITICAL:HTTP:ANOM
Action: Close
Log: (select)
Group: CRITICAL:HTTP:SIGS
Action: Close
Log: (select)
Group: HIGH:HTTP:ANOM
Action: Close
Log: (select)
Group: HIGH:HTTP:SIGS
Action: Close
Log: (select)
Group: MEDIUM:HTTP:ANOM
Action: Close
Log: (select)
Group: MEDIUM:HTTP:SIGS
Action: Close
Log: (select)
Group: CRITICAL:FTP:SIGS

Action: Close
Log: (select)

6. **Policy ID 3**
   Policies > (From: Trust, To: Untrust) New: Enter the following, then click **OK**:

   Source Address:
   Address Book Entry: (select), Any
   Destination Address:
   Address Book Entry: (select), Any
   Service: HTTP

   > Click **Multiple**, select **FTP-GET**, **HTTPS**, **PING**, then click **OK** to return to the basic policy configuration page.

   Action: Permit

   > Click **Deep Inspection**, enter the following, click **Add** to enter each attack object group, then click **OK** to return to the basic policy configuration page:

   Group: CRITICAL:HTTP:ANOM
   Action: Close Client
   Log: (select)
   Group: CRITICAL:HTTP:SIGS
   Action: Close Client
   Log: (select)
   Group: HIGH:HTTP:ANOM
   Action: Close Client
   Log: (select)
   Group: HIGH:HTTP:SIGS
   Action: Close Client
   Log: (select)
   Group: MEDIUM:HTTP:ANOM
   Action: Close Client
   Log: (select)
   Group: MEDIUM:HTTP:SIGS
   Action: Close Client
   Log: (select)
   Group: CRITICAL:FTP:SIGS
   Action: Close Client
   Log: (select)

CLI

1. **Interfaces**

```
set interface ethernet1 zone trust
set interface ethernet1 ip 10.1.1.1/24
set interface ethernet1 manage
set interface ethernet3 zone untrust
set interface ethernet3 ip 1.1.1.1/24
set interface ethernet2 zone dmz
set interface ethernet2 ip 2.1.1.1/24
```

2. **Addresses**

```
set address dmz websrv1 1.2.2.5/32
set address dmz websrv2 1.2.2.6/32
```

3. **Route**

   set vrouter trust-vr route 0.0.0.0/0 interface ethernet3 gateway 1.1.1.250

4. **Policy ID 1**

   set policy id 1 from untrust to dmz any websrv1 http permit attack
   CRITICAL:HTTP:ANOM action close-server
   set policy id 1
   ns(policy:1)-> set dst-address websrv2
   ns(policy:1)-> set service ftp-get
   ns(policy:1)-> set service https
   ns(policy:1)-> set service ping
   ns(policy:1)-> set attack CRITICAL:HTTP:SIGS action close-server
   ns(policy:1)-> set attack HIGH:HTTP:ANOM action close-server
   ns(policy:1)-> set attack HIGH:HTTP:SIGS action close-server
   ns(policy:1)-> set attack MEDIUM:HTTP:ANOM action close-server
   ns(policy:1)-> set attack MEDIUM:HTTP:SIGS action close-server
   ns(policy:1)-> set attack CRITICAL:FTP:SIGS action close-server
   ns(policy:1)-> exit

5. **Policy ID 2**

   set policy id 2 from trust to dmz any websrv1 http permit attack
   CRITICAL:HTTP:ANOM action close
   set policy id 2
   ns(policy:2)-> set dst-address websrv2
   ns(policy:2)-> set service ftp
   ns(policy:2)-> set service https
   ns(policy:2)-> set service ping
   ns(policy:2)-> set attack CRITICAL:HTTP:SIGS action close
   ns(policy:2)-> set attack HIGH:HTTP:ANOM action close
   ns(policy:2)-> set attack HIGH:HTTP:SIGS action close
   ns(policy:2)-> set attack MEDIUM:HTTP:ANOM action close
   ns(policy:2)-> set attack MEDIUM:HTTP:SIGS action close
   ns(policy:2)-> set attack CRITICAL:FTP:SIGS action close
   ns(policy:2)-> exit

6. **Policy ID 3**

   set policy id 3 from trust to untrust any any http permit attack
   CRITICAL:HTTP:ANOM action close-client
   set policy id 3
   ns(policy:3)-> set service ftp-get
   ns(policy:3)-> set service https
   ns(policy:3)-> set service ping
   ns(policy:3)-> set attack CRITICAL:HTTP:SIGS action close-client
   ns(policy:3)-> set attack HIGH:HTTP:ANOM action close-client
   ns(policy:3)-> set attack HIGH:HTTP:SIGS action close-client
   ns(policy:3)-> set attack MEDIUM:HTTP:ANOM action close-client
   ns(policy:3)-> set attack MEDIUM:HTTP:SIGS action close-client
   ns(policy:3)-> set attack CRITICAL:FTP:SIGS action close-client
   ns(policy:3)-> exit
   save

Brute Force Attack Actions

A typical brute force attack is accomplished by sending lots of traffic with varying source ports or IP in an attempt to obtain network access. In order to effectively prevent future attempts, ScreenOS allows you to associate an IP action for each attack group in a policy.

Brute force attack is detected based on the threshold values set for the DI supported protocols. For example,

set di service *protocol-name value*

Apart from a DI action, brute force attack actions are configured with the **IP action** command for a configured amount of time for a specified target. If your security device detects a brute force attack, then select one of the following actions to perform:

- Notify: The security device logs the event but does not take any action against further traffic matching the target definition for the period of time specified in the timeout setting.
- Block: The security device logs the event and drops all further traffic matching the target definition for the period of time specified in the timeout setting.
- Close: The security device logs the event and drops all further traffic matching the target definition for the period of time specified in the timeout setting, and sends a Reset (RST) for TCP traffic to the source and destination addresses.

Brute Force Attack Objects

Table 10 lists the brute force attack objects in ScreenOS 5.3 and the threshold parameters that can be used with the IP actions.

Table 10: Brute Force Attack Objects

Brute Force Attack Name	Parameter
HTTP Brute Force Login Attempt	failed_logins
HTTP Brute Search Attempt	brute_search
IMAP Brute Force Login Attempt	failed_logins
LDAP Brute Force Login Attempt	failed_logins
MS-RPC IsSystemActive request flood	Not configurable—32 attempts
MS-SQL Login Brute Force	Not configurable—4 attempts
POP3 Brute Force Login Attempt	failed_login
RADIUS Brute Force Authentication Attempt	failed_auth
SMB Brute Force Directory Create/Delete	Not configurable—200 attempts
SMB Brute Force Login Attempt	failed_login
FTP Brute Force Login Attempt	failed_login
Telnet Brute Force Login Attempt	failed_login
VNC Brute Force Login Attempt	failed_login

Brute Force Attack Target

The target option specifies a set of elements that must match for the security device to consider a packet part of a brute force attack. The specified set of elements in an IP packet arriving during a specified timeout period must match that in the packet that the security device detected as part of a brute force attack for the subsequent packet to be considered part of the same attack. The default target definition is Serv. You can select any of the following target definitions shown in Table 11.

Table 11: Target Options

Target option	Matching elements
Serv	source IP, destination IP, destination port, and protocol
Src-IP	source IP address
Zone-Serv	source security zone, destination IP, destination port number, and protocol
Dst-IP	destination IP address
Zone	source security zone (The security zone to which the ingress interface is bound; that is, the source security zone from which the attacking packets originate)

Brute Force Attack Timeout

Timeout is a period of time following brute force attack detection during which the security device performs an IP action on packets matching specified target parameters. The default timeout is 60 seconds.

Example 1

In this example, you configure an IP action along with the existing DI action for each group in a policy. The following CLI commands block brute force attack object—HTTP Brute Force Login Attempt or HTTP Brute Force Search for 45 seconds. All other attacks in the HIGH:HTTP:ANOM attack group are configured with a DI action of **close**.

CLI

```
ns> get attack group HIGH:HTTP:ANOM
GROUP "HIGH:HTTP:ANOM" is pre-defined. It has the following members
ID     Name
1674 HTTP:INVALID:INVLD-AUTH-CHAR
1675 HTTP:INVALID:INVLD-AUTH-LEN
1711 HTTP:OVERFLOW:HEADER
1713 HTTP:OVERFLOW:INV-CHUNK-LEN
1717 HTTP:OVERFLOW:AUTH-OVFLW
5394 HTTP:EXPLOIT:BRUTE-FORCE
5395 HTTP:EXPLOIT:BRUTE-SEARCH

ns> set policy id 1 from Untrust to DMZ Any Any Any permit attack
MEDIUM:HTTP:ANOM action none
ns> set policy id 1

ns(policy:1)-> set attack HIGH:HTTP:ANOM action close ip-action block target dst-ip
timeout 45
```

If the configured attack group does not have any brute force attack protocol anomalies, IP action is not enforced.

Example 2

In this example, you associate an IP action for each attack group for a configured amount of time from a specified host.

set policy id 1 from trust to untrust any any any permit attack POP3 BRUTE FORCE Login Attempt action close ip-action notify target serv timeout 60

Example 3

In this example, the default threshold value of FTP brute force login attempt is 8 attempts per minute. If a user at IP address 192.168.2.2 is launching a FTP brute force login attempt to FTP server at 10.150.50.5 in order to figure out a user account name and password, the attempt is detected when the attacker makes 8 FTP login attempts within a minute.

If an IP action is configured to "Block" for 120 seconds for target of "serv", any traffic coming from 192.168.2.2 (src IP) to 10.150.50.5 (dst IP) over TCP (protocol) port 21 (dst port) is blocked for 120 seconds.

Note that some IP action targets may affect traffic matching another policy.

Attack Logging

You can enable the logging of detected attacks per attack group per policy. In other words, within the same policy, you can apply multiple attack groups and selectively enable the logging of detected attacks for just some of them.

By default, logging is enabled. You might want to disable logging for attacks that are lower priority for you and about which you do not give much attention. Disabling logging for such attacks helps prevent the event log from becoming cluttered with entries that you do not plan to look at anyway.

Example: Disabling Logging per Attack Group

In this example, you reference the following five attack groups in a policy and enable logging only for the first two:

- HIGH:IMAP:ANOM
- HIGH:IMAP:SIGS
- MEDIUM:IMAP:ANOM
- LOW:IMAP:ANOM
- INFO:IMAP:ANOM

The policy applies to IMAP traffic from all hosts in the Trust zone to a mail server named "mail1" in the DMZ. If any of the predefined IMAP attack objects in the above five groups match an attack, the security device closes the connection. However, it only creates log entries for detected attacks matching attack objects in the first two groups.

WebUI

1. **Address**
   Objects > Addresses > List > New: Enter the following, then click **OK**:

   Address Name: mail1
   IP Address/Domain Name:
   IP/Netmask: (select), 1.2.2.10/32
   Zone: DMZ

2. Policy

Policies > (From: Trust, To: DMZ) New: Enter the following, then click **OK**:

Source Address:
Address Book Entry: (select), Any
Destination Address:
Address Book Entry: (select), mail1
Service: IMAP
Action: Permit

> Click **Deep Inspection**, enter the following, click **Add** to enter each attack object group, then click **OK** to return to the basic policy configuration page:

Group: HIGH:IMAP:ANOM
Action: Close
Log: (select)

Group: HIGH:IMAP:SIGS
Action: Close
Log: (select)

Group: MEDIUM:IMAP:ANOM
Action: Close
Log: (clear)

Group: LOW:IMAP:ANOM
Action: Close
Log: (clear)

Group: INFO:IMAP:ANOM
Action: Close
Log: (clear)

CLI

1. Address

```
set address dmz mail1 1.2.2.10/32
```

2. Policy

```
ns-> set policy id 1 from trust to dmz any mail1 imap permit attack
     HIGH:IMAP:ANOM action close
ns-> set policy id 1
ns(policy:1)-> set attack HIGH:IMAP:SIGS action close
ns(policy:1)-> set attack MEDIUM:IMAP:ANOM action close
ns(policy:1)-> unset attack MEDIUM:IMAP:ANOM logging
ns(policy:1)-> set attack LOW:IMAP:ANOM action close
ns(policy:1)-> unset attack LOW:IMAP:ANOM logging
ns(policy:1)-> set attack INFO:IMAP:ANOM action close
ns(policy:1)-> unset attack INFO:IMAP:ANOM logging
ns(policy:1)-> exit
ns-> save
```

Mapping Custom Services to Applications

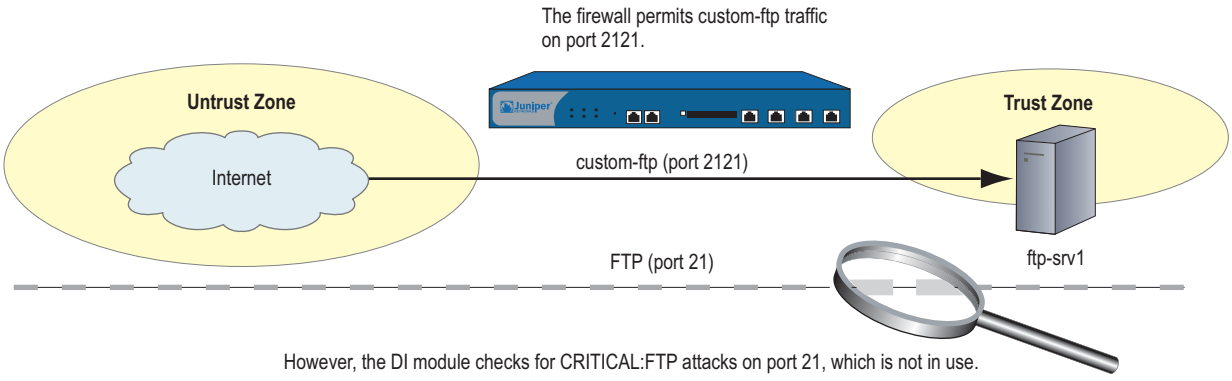
When using a custom service in a policy with a Deep Inspection (DI) component, you must explicitly specify the application that is running on that service so that the DI module can function properly. For example, if you create a custom service for FTP running on a nonstandard port number such as 2121 (see Figure 51), you can reference that custom service in a policy as follows:

```
set service ftp-custom protocol tcp src-port 0-65535 dst-port 2121-2121
set policy id 1 from untrust to trust any ftp-srv1 custom-ftp permit
```

However, if you add a DI component to a policy that references a custom service, the DI module cannot recognize the application because it is using a nonstandard port number.

```
set policy id 1 from untrust to trust any ftp-srv1 custom-ftp permit attack
    CRITICAL:FTP:SIGS action close-server
```

Figure 51: Mapping Custom Service



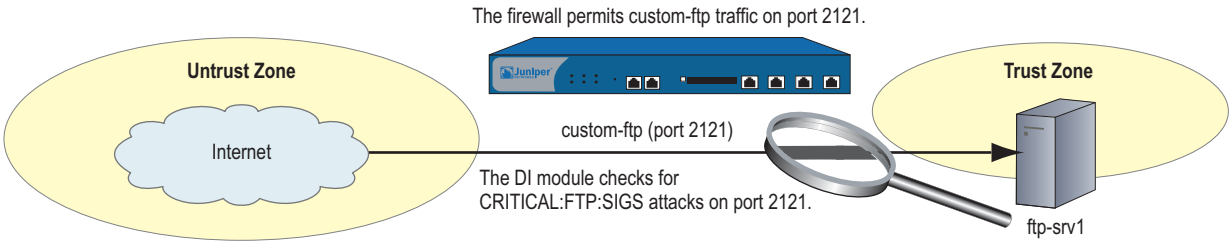
To avoid this problem, you must inform the DI module that the FTP application is running on port 2121 (see Figure 52). Essentially, you must map the protocol in the Application Layer to a specific port number in the Transport Layer. You can do this binding at the policy level:

```
set policy id 1 application ftp
```

When you map the FTP application to the custom service “custom-ftp” and configure DI to examine FTP traffic for the attacks defined in the CRITICAL:FTP:SIGS attack object group in a policy that references custom-ftp, the DI module perform its inspection on port 2121.

```
set policy id 1 from untrust to trust any ftp-srv1 custom-ftp permit attack
    CRITICAL:FTP:SIGS action close-server
set policy id 1 application ftp
```

Figure 52: Mapping Custom Service to Attack Object Group



Example: Mapping an Application to a Custom Service

In this example, you define a custom service named "HTTP1" that uses destination port 8080. You map the HTTP application to the custom service for a policy permitting HTTP1 traffic from any address in the Untrust zone to a webserver named "server1" in the DMZ zone. You then apply Deep Inspection (DI) to the permitted HTTP traffic running on port 8080. The DI settings for this policy are as follows:

- Attack Object Groups:
  - CRITICAL:HTTP:ANOM, CRITICAL:HTTP:SIGS
  - HIGH:HTTP:ANOM, HIGH:HTTP:SIGS
  - MEDIUM:HTTP:ANOM, MEDIUM:HTTP:SIGS
- Action for all attack object groups: Close Server
- Logging: Enabled (default setting)

WebUI

1. **Custom Service**
   Objects > Services > Custom > New: Enter the following, then click **OK**:

   Service Name: HTTP1
   Transport Protocol: TCP (select)
   Source Port Low: 0
   Source Port High: 65535
   Destination Port Low: 8080
   Destination Port High: 8080

2. **Address**
   Objects > Addresses > List > New: Enter the following, then click **OK**:

   Address Name: server1
   IP Address/Domain Name:
   IP/Netmask: 1.2.2.5/32
   Zone: DMZ

3. **Policy**
   Policies > (From: Untrust, To: DMZ) New: Enter the following, then click **OK**:

   Source Address:
   Address Book Entry: (select), Any
   Destination Address:
   Address Book Entry: (select), server1
   Service: HTTP1
   Application: HTTP
   Action: Permit

   > Click **Deep Inspection**, enter the following, click **Add** to enter each attack object group, then click **OK** to return to the basic policy configuration page:

   Group: CRITICAL:HTTP:ANOM
   Action: Close Server
   Log: (select)

   Group: CRITICAL:HTTP:SIGS
   Action: Close Client
   Log: (select)

   Group: HIGH:HTTP:ANOM
   Action: Close Client
   Log: (select)

   Group: HIGH:HTTP:SIGS
   Action: Close Client
   Log: (select)

   Group: MEDIUM:HTTP:ANOM
   Action: Close Client
   Log: (select)

   Group: MEDIUM:HTTP:SIGS
   Action: Close Client
   Log: (select)

CLI

1. **Custom Service**
   set service HTTP1 protocol tcp src-port 0-65535 dst-port 8080-8080

2. **Address**
   set address dmz server1 1.2.2.5/32

3. **Policy**
   ns-> set policy id 1 from untrust to dmz any server1 HTTP1 permit attack CRITICAL:HTTP:ANOM action close-server
   ns-> set policy id 1
   ns(policy:1)-> set attack CRITICAL:HTTP:SIGS action close-server
   ns(policy:1)-> set attack HIGH:HTTP:ANOM action close-server
   ns(policy:1)-> set attack HIGH:HTTP:SIGS action close-server
   ns(policy:1)-> set attack MEDIUM:HTTP:ANOM action close-server
   ns(policy:1)-> set attack MEDIUM:HTTP:SIGS action close-server
   ns(policy:1)-> exit
   ns-> set policy id 1 application http
   save

Example: Application-to-Service Mapping for HTTP Attacks

Some known HTTP attacks use TCP port 8000. At the time of this writing, there are currently two such attacks in the Deep Inspection (DI) attack object database:

- 3656, App: HP Web JetAdmin Framework Infoleak

  DOS:NETDEV:WEBJET-FW-INFOLEAK (in the attack object group MEDIUM:HTTP:SIGS)

- 3638, App: HP Web JetAdmin WriteToFile Vulnerability,

  DOS:NETDEV:WEBJET-WRITETOFILE (in the attack object group CRITICAL:HTTP:SIGS)

However, by default, ScreenOS considers only TCP traffic on port 80 to be HTTP. Therefore, if the security device receives TCP traffic using port 8000, it does not recognize it as HTTP. Consequently the DI engine does not scan such HTTP traffic for these attacks and cannot detect them if they occur—unless you map HTTP as an application to a custom service using port 8000.

In this example, you associate traffic using the nonstandard port of 8000 with HTTP to detect the above attacks.

NOTE: In general, if you are running some services using nonstandard port numbers in your network and you want the DI engine to scan that traffic, you must associate the nonstandard port number to the service.

WebUI

1. **Custom Service**
   Objects > Services > Custom > New: Enter the following, then click **OK**:

   Service Name: HTTP2
   Transport Protocol: TCP (select)
   Source Port Low: 0
   Source Port High: 65535
   Destination Port Low: 8000
   Destination Port High: 8000

2. **Policy**
   Policies > (From: Untrust, To: DMZ) New: Enter the following, then click **OK**:

   Source Address:
   Address Book Entry: (select), Any
   Destination Address:
   Address Book Entry: (select), Any
   Service: HTTP2
   Application: HTTP
   Action: Permit

> Click **Deep Inspection**, enter the following, click **Add** to enter each attack object group, then click **OK** to return to the basic policy configuration page:

Group: CRITICAL:HTTP:SIGS
Action: Close
Log: (select)

Group: MEDIUM:HTTP:SIGS
Action: Close
Log: (select)

CLI

1. **Custom Service**
   set service HTTP2 protocol tcp src-port 0-65535 dst-port 8000-8000
2. **Policy**
   ns-> set policy id 1 from untrust to dmz any any HTTP2 permit attack CRITICAL:HTTP:SIGS action close
   ns-> set policy id 1
   ns(policy:1)-> set attack MEDIUM:HTTP:SIGS action close
   ns(policy:1)-> exit
   ns-> set policy id 1 application http
   save

Customized Attack Objects and Groups

You can define new attack objects and object groups to customize the Deep Inspection (DI) application to best meet your needs. User-defined attack objects can be stateful signatures or—on the NetScreen-5000—TCP stream signatures. You can also adjust various parameters to modify predefined protocol anomaly attack objects.

User-Defined Stateful Signature Attack Objects

You can create a stateful signature attack object for FTP, HTTP, and SMTP. When creating an attack object, you perform the following steps:

- Name the attack object. (All user-defined attack objects must begin with "CS:".)
- Set the context for the DI search. (For a complete list of all the contexts that you can use when creating attack objects, see "Contexts for User-Defined Signatures" on page I)
- Define the signature. ("Regular Expressions" on page 137 examines the regular expressions that you can use when defining signatures.)
- Assign the attack object a severity level. (For information on severity levels, see "Changing Severity Levels" on page 117.)

You must then put a user-defined attack object in a user-defined attack object group for use in policies.

NOTE: A user-defined attack object group can only contain user-defined attack objects. You cannot mix predefined and user-defined attack objects in the same attack object group.

Regular Expressions

When entering the text string for a signature, you can enter an alphanumeric string of ordinary characters to search for an exact character-to-character match, or you can use regular expressions to broaden the search for possible matches to sets of characters. ScreenOS supports the following regular expressions as shown in Table 12.

Table 12: ScreenOS Supported Regular Expressions

Purpose	Meta characters	Example	Meaning
Direct binary match (octal)[1]	\O*octal_number*	\0162 Matches: 162	Exactly match this octal number: 162
Direct binary match (hexadecimal)[2]	\X*hexadecimal_number*\X	\X01 A5 00 00\X Matches: 01 A5 00 00	Exactly match these four hexadecimal numbers: 01 A5 00 00
Case-insensitive matches	\[*characters*\]	\[cat\] Matches: ■ Cat, cAt, caT ■ CAt, CaT, CAT ■ cat, cAt	Match the characters in cat regardless of the case of each character
Match any character	.	c . t Matches: ■ cat, cbt, cct, … czt ■ cAt, cBt, cCt, … cZt ■ c1t, c2t, c3t, … c9t	Match c-any character-t
Match the previous character zero or more times, instead of only once	*	a*b + c Matches: ■ bc ■ bbc ■ abc ■ aaabbbbc	Match zero, one, or multiple occurrences of a, followed by one or more occurrences of b, followed by one occurrence of c
Match the previous character one or more times	+	a + b + c Matches: ■ abc ■ aabc ■ aaabbbbc	Match one or more occurrences of a, followed by one or more occurrences of b, followed by one occurrence of c
Match the previous character zero times or one time	?	drop-?packet Matches: ■ drop-packet ■ droppacket	Match either drop-packet or droppacket
Group expressions	( )		

Purpose	Meta characters	Example	Meaning
Either the previous or the following character—typically used with ( )	|	(drop | packet) Matches: ■ drop ■ packet	Match either drop or packet
Character range	[*start-end*]	[c-f]a(d | t) Matches: ■ cad, cat ■ dad, dat ■ ead, eat ■ fad, fat	Match everything that begins with c, d, e, or f and that has the middle letter a and the last letter d or t
Negation of the following character	[^*character*]	[^0-9A-Z] Matches: a, b, c, d, e, … z	Match lowercase letters

1.Octal is a base-8 number system that uses only the digits 0–7.
2. Hexadecimal is a base-16 number system that uses digits 0–9 as usual, and then the letters A–F representing hexadecimal digits with decimal values of 10–15.

Example: User-Defined Stateful Signature Attack Objects

In this example, you have an FTP server, a webserver, and a mail server in the DMZ zone. You define the following attack objects for the use-defined signature objects as shown in Table 13.

Table 13: User-Defined Stateful Signature Attack Objects

Object Name	Usage
cs:ftp-stor	Block someone from putting files on an FTP server
cs:ftp-user-dm	Deny FTP access to the user with the login name dmartin
cs:url-index	Block HTTP packets with a defined URL in any HTTP request
cs:spammer	Block e-mail from any e-mail address at “spam.com”

You then organize them into a user-defined attack object group named “DMZ DI”, which you reference in a policy permitting traffic from the Untrust zone to the servers in the DMZ zone.

WebUI

1. **Attack Object 1: ftp-stor**
   Objects > Attacks > Custom > New: Enter the following, then click **OK**:

   Attack Name: cs:ftp-stor
   Attack Context: FTP Command
   Attack Severity: Medium
   Attack Pattern: STOR

2. **Attack Object 2: ftp-user-dm**
   Objects > Attacks > Custom > New: Enter the following, then click **OK**:

   Attack Name: cs:ftp-user-dm
   Attack Context: FTP User Name

Attack Severity: Low
Attack Pattern: dmartin

3. **Attack Object 3: url-index**
   Objects > Attacks > Custom > New: Enter the following, then click **OK**:

   Attack Name: cs:url-index
   Attack Context: HTTP URL Parsed
   Attack Severity: High
   Attack Pattern: .*index.html.*

4. **Attack Object 4: spammer**
   Objects > Attacks > Custom > New: Enter the following, then click **OK**:

   Attack Name: cs:spammer
   Attack Context: SMTP From
   Attack Severity: Info
   Attack Pattern: .*@spam.com

5. **Attack Object Group**
   Objects > Attacks > Custom Groups > New: Enter the following group name, move the following custom attack objects, then click **OK**:

   Group Name: CS:DMZ DI

   Select **cs:ftp-stor** and use the < < button to move the address from the Selected Members column to the Selected Members column.

   Select **cs:ftp-user-dm** and use the < < button to move the address from the Available Members column to the Selected Members column.

   Select **cs:url-index** and use the < < button to move the address from the Available Members column to the Selected Members column.

   Select **cs:spammer** and use the < < button to move the address from the Available Members column to the Selected Members column.

6. **Policy**
   Policies > (From: Untrust, To: DMZ) New: Enter the following, then click **OK**:

   Source Address:
   Address Book Entry: (select), Any
   Destination Address:
   Address Book Entry: (select), Any
   Service: HTTP

   > Click **Multiple**, select **FTP**, then click **OK** to return to the basic policy configuration page.

   Action: Permit

> Click **Deep Inspection**, enter the following, click **Add** to enter each attack object group, then click **OK** to return to the basic policy configuration page:

Group: CS:DMZ DI
Action: Close Server
Log: (select)

CLI

1. **Attack Object 1: ftp-stor**
   set attack cs:ftp-stor ftp-command STOR severity medium
2. **Attack Object 2: ftp-user-dm**
   set attack cs:ftp-user-dm ftp-username dmartin severity low
3. **Attack Object 3: url-index**
   set attack cs:url-index http-url-parsed index.html severity high
4. **Attack Object 4: url-index**
   set attack cs:spammer smtp-from .*@spam.com severity info
5. **Attack Object Group**
   set attack group “CS:DMZ DI”
   set attack group “CS:DMZ DI” add cs:ftp-stor
   set attack group “CS:DMZ DI” add cs:ftp-user-dm
   set attack group “CS:DMZ DI” add cs:url-index
   set attack group “CS:DMZ DI” add cs:spammer
6. **Policy**
   set policy id 1 from untrust to dmz any any http permit attack “CS:DMZ DI” action close-server
   set policy id 1
   ns(policy:1)-> set service ftp
   ns(policy:1)-> exit
   save

TCP Stream Signature Attack Objects

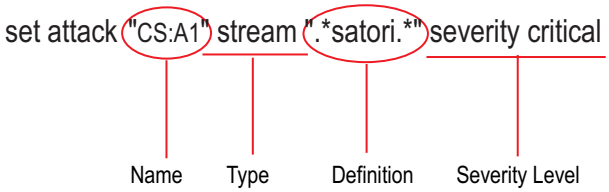
The stateful signatures are context-based within specific applications, such as an FTP username or an SMTP header field. TCP stream signatures look for patterns anywhere in any kind of TCP traffic regardless of the application protocol in use.

NOTE: You can define TCP stream signatures on NetScreen-5000 series systems only.

Because there are no predefined TCP stream signature attack objects, you must define them. When creating a signature attack object, you define the following components:

- Attack object name (All user-defined attack objects must begin with “CS:”.)
- Object type (“stream”)
- Pattern definition
- Severity level

Figure 53: Example of a TCP Stream Signature Attack Object



Example: User-Defined Stream Signature Attack Object

In this example, you define a stream signature object ".*satori.*". You name it "CS:A1" and define its severity level as critical. Because a policy can reference only attack object groups, you create a group named "CS:Gr1", and then add this object to it. Finally, you define a policy that references CS:Gr1 and that instructs the security device to sever the connection and send TCP RST to the client if the pattern appears in any traffic to which the policy applies.

WebUI

1. **Stream Signature Attack Object**
   Objects > Attacks > Custom > New: Enter the following, then click **OK**:

   Attack Name: CS:A1
   Attack Context: Stream
   Attack Severity: Critical
   Attack Pattern: .*satori.*

2. **Stream Signature Attack Object Group**
   Objects > Attacks > Custom Groups > New: Enter the following, then click **OK**:

   Group Name: CS:Gr1

   Select **CS:A1** in the Available Members column and then click < < to move it to the Selected Members column.

3. **Policy**
   Policies > (From: Trust, To: Untrust) New: Enter the following, then click **OK**:

   Source Address:
   Address Book Entry: (select), Any
   Destination Address:
   Address Book Entry: (select), Any
   Service: ANY
   Action: Permit

   > Click **Deep Inspection**, enter the following, click **Add** to enter each attack object group; and then click **OK** to return to the basic policy configuration page:

   Group: CS:Gr1
   Action: Close Client
   Log: (select)

CLI

1. **Stream Signature Attack Object**
   set attack "CS:A1" stream ".*satori.*" severity critical
2. **Stream Signature Attack Group**
   set attack group "CS:Gr1"
   set attack group "CS:Gr1" add "CS:A1"
3. **Policy**
   set policy from trust to untrust any any any permit attack CS:Gr1 action close-client save

Configurable Protocol Anomaly Parameters

You can modify certain parameters of a protocol anomaly attack object. Although Juniper defines protocol anomaly attack objects to find deviations from protocol standards defined in RFCs and common RFC extensions, not all implementations adhere to these standards. If you find that the application of a certain protocol anomaly attack object is producing numerous false positives, you can modify its parameters to better match the accepted use of that protocol in your network.

NOTE: For a complete list of all configurable parameters, see the **di** command in *ScreenOS CLI Reference Guide IPv4 Command Descriptions*.

Example: Modifying Parameters

In this example, you set higher values for the following parameters to reduce the number of false positives that occurred with the default settings:

Protocol Parameter	Default	New
SMB—Maximum number of login failures per minute	4 failures	8 failures
Gnutella—Maximum number of time-to-live (TTL) hops	8 hops	10 hops

For the following parameters, you set lower values to detect anomalous behavior that the security device missed with the default settings:

Protocol Parameter	Default	New
AOL Instant Messenger (AIM)—Maximum OSCAR File Transfer (OFT) file name length. OSCAR = Open System for Communication in Real-time, the protocol that AIM clients use.	10,000 bytes	5,000 bytes
AOL Instant Messenger—Maximum length of a FLAP frame (FLAP header, which is always 6 bytes, plus data). OSCAR makes use of a the FLAP protocol to make connections and open channels between AIM clients.	10,000 bytes	5,000 bytes

WebUI

NOTE: You must use the CLI to modify protocol anomaly parameters.

CLI

```
set di service smb failed_logins 8
set di service gnutella max_ttl_hops 10
set di service aim max_flap_length 5000
set di service aim max_oft_frame 5000
save
```

Negation

Typically, you use attack objects to match patterns that are indicative of malicious or anomalous activity. However, you can also use them to match patterns indicative of benign or legitimate activity. WIth this approach, something is amiss only if a type of traffic does *not* match a particular pattern. To use attack objects in this way, you apply the concept of negation.

A useful application of attack object negation would be to block all login attempts other than those with the correct username and password. It would be difficult to define all invalid usernames and passwords, but quite easy to define the correct ones and then apply negation to reverse what the security device considers an attack; that is, everything except the specified attack object.

Example: Attack Object Negation

In this example (see Figure 54), you define two attack objects: one specifying the correct username required to log in to an FTP server, and another the correct password. You then apply negation to both attack objects, so that the security device blocks any login attempt to that server that uses any other username or password than those defined in the attack objects.

The example uses the following settings:

- The correct username and password are *admin1* and *pass1*.
- The FTP server is at 1.2.2.5 in the DMZ zone. Its address name is *ftp1*.
- You apply DI on FTP traffic to the server from all hosts in the Untrust and Trust zones.
- All security zones are in the trust-vr routing domain.

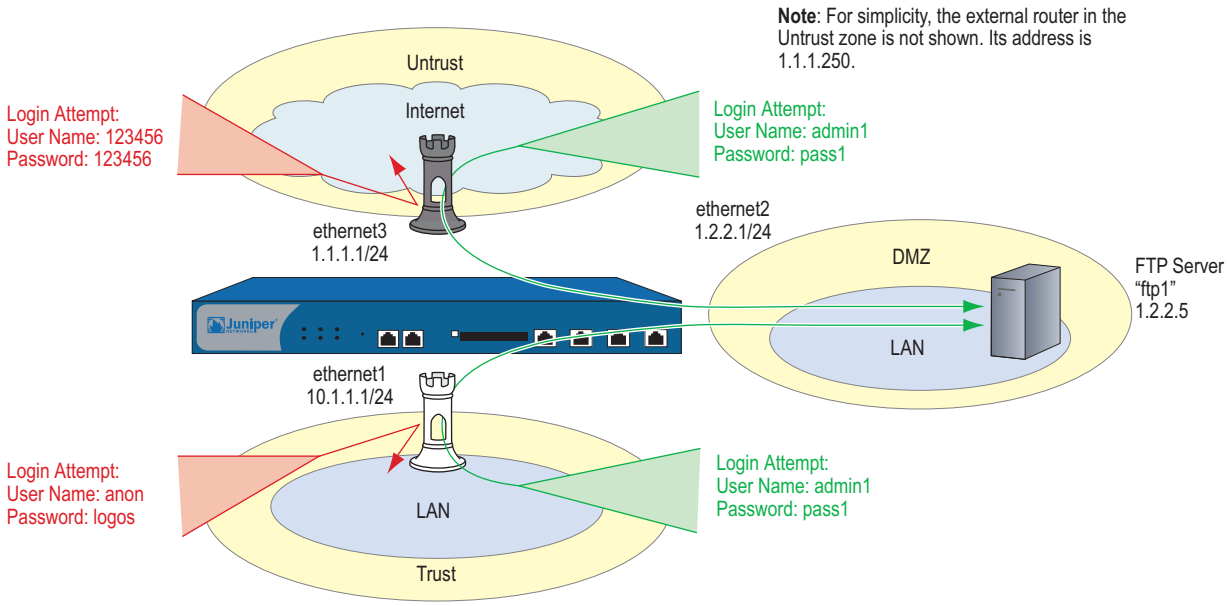
You create the following two attack objects:

- Attack Object #1:
  - Name: CS:FTP1_USR_OK
  - Negation: enabled
  - Context: ftp-username
  - Pattern: admin1
  - Severity: high

- Attack Object #2:
  - Name: CS:FTP1_PASS_OK
  - Negation: enabled
  - Context: ftp-password
  - Pattern: pass1
  - Severity: high

You then put both objects into an attack object group named *CS:FTP1_LOGIN* and reference that attack object group in two policies permitting FTP traffic from the Trust and Untrust zones to ftp1 in the DMZ.

Figure 54: Attack Object Negation



WebUI

1. **Interfaces**

   Network > Interfaces > Edit (for ethernet1): Enter the following, then click **Apply**:

   Zone Name: Trust
   Static IP: (select this option when present)
   IP Address/Netmask: 10.1.1.1/24

   Select the following, then click **OK**:

   Interface Mode: NAT (select)

NOTE: By default, any interface that you bind to the Trust zone is in NAT mode. Consequently, this option is already enabled for interfaces bound to the Trust zone.

Network > Interfaces > Edit (for ethernet2): Enter the following, then click **OK**:

Zone Name: DMZ
Static IP: (select this option when present)
IP Address/Netmask: 1.2.2.1/24

Network > Interfaces > Edit (for ethernet3): Enter the following, then click **OK**:

Zone Name: Untrust
Static IP: (select this option when present)
IP Address/Netmask: 1.1.1.1/24

2. Address

Objects > Addresses > List > New: Enter the following, then click **OK**:

Address Name: ftp1
IP Address/Domain Name:
IP/Netmask: (select), 1.2.2.5/32
Zone: DMZ

3. Attack Object 1: CS:FTP1_USR_OK

Objects > Attacks > Custom > New: Enter the following, then click **OK**:

Attack Name: CS:FTP1_USR_OK
Attack Context: ftp-username
Attack Severity: High
Attack Pattern: admin1
Pattern Negation: (select)

4. Attack Object 2: CS:FTP1_PASS_OK

Objects > Attacks > Custom > New: Enter the following, then click **OK**:

Attack Name: CS:FTP1_PASS_OK
Attack Context: ftp-password
Attack Severity: High
Attack Pattern: pass1
Pattern Negation: (select)

5. Attack Object Group

Objects > Attacks > Custom Groups > New: Enter the following group name, move the following custom attack objects, then click **OK**:

Group Name: CS:FTP1_LOGIN

Select **CS:FTP1_USR_OK** and use the < < button to move the address from the Available Members column to the Selected Members column.

Select **CS:FTP1_PASS_OK** and use the < < button to move the address from the Available Members column to the Selected Members column.

6. **Route**
   Network > Routing > Routing Entries > trust-vr New: Enter the following, then click **OK**:

   Network Address/Netmask: 0.0.0.0/0
   Gateway: (select)
   Interface: ethernet3
   Gateway IP Address: (select) 1.1.1.250

7. **Policies**
   Policies > (From: Untrust, To: DMZ) New: Enter the following, then click **OK**:

   Source Address:
   Address Book Entry: (select), Any
   Destination Address:
   Address Book Entry: (select), ftp1
   Service: FTP
   Action: Permit

   > Click **Deep Inspection**, enter the following, click **Add** to enter each attack object group; and then click **OK** to return to the basic policy configuration page:

   Group: CS:FTP1_LOGIN
   Action: Drop
   Log: (select)

   Policies > (From: Trust, To: DMZ) New: Enter the following, then click **OK**:

   Source Address:
   Address Book Entry: (select), Any
   Destination Address:
   Address Book Entry: (select), ftp1
   Service: FTP
   Action: Permit

   > Click **Deep Inspection**, enter the following, click **Add** to enter each attack object group, then click **OK** to return to the basic policy configuration page:

   Group: CS:FTP1_LOGIN
   Action: Drop
   Log: (select)

CLI

1. **Interfaces**
   set interface ethernet1 zone trust
   set interface ethernet1 ip 10.1.1.1/24
   set interface ethernet1 nat
   set interface ethernet2 zone dmz
   set interface ethernet2 ip 1.2.2.1/24
   set interface ethernet3 zone untrust
   set interface ethernet3 ip 1.1.1.1/24
2. **Address**
   set address dmz ftp1 1.2.2.5/32
3. **Attack Objects**
   set attack CS:FTP1_USR_OK ftp-username not admin1 severity high
   set attack CS:FTP1_PASS_OK ftp-password not pass1 severity high
   set attack group CS:FTP1_LOGIN
   set attack group CS:FTP1_LOGIN add CS:FTP1_USR_OK
   set attack group CS:FTP1_LOGIN add CS:FTP1_PASS_OK
4. **Route**
   set vrouter trust-vr route 0.0.0.0/0 interface ethernet3 gateway 1.1.1.250
5. **Policies**
   set policy from untrust to dmz any ftp1 ftp permit attack CS:FTP1_LOGIN action drop
   set policy from trust to dmz any ftp1 ftp permit attack CS:FTP1_LOGIN action drop
   save

Granular Blocking of HTTP Components

A Juniper Networks security device can selectively block ActiveX controls, Java applets, .zip files, and .exe files sent via HTTP. The danger that these components pose to the security of a network is that they provide a means for an untrusted party to load and then control an application on hosts in a protected network.

When you enable the blocking of one or more of these components in a security zone, the security device examines every HTTP header that arrives at an interface bound to that zone. It checks if the content type listed in the header indicates that any of the targeted components are in the packet payload. If the content type is Java, .exe, or .zip and you have configured the security device to block those HTTP component types, the device blocks the packet. If the content type lists only "octet stream" instead of a specific component type, then the device examines the file type in the payload. If the file type is Java, .exe, or .zip and you have configured the device to block those component types, the device blocks the packet.

When you enable the blocking of ActiveX controls, the device blocks all HTTP packets containing any type of HTTP component in its payload—ActiveX controls, Java applets, .exe files, or .zip files.

NOTE: When ActiveX-blocking is enabled, the security device blocks Java applets, .exe files, and .zip files whether or not they are contained within an ActiveX control.

ActiveX Controls

Microsoft ActiveX technology provides a tool for web designers to create dynamic and interactive web pages. ActiveX controls are components that allow different programs to interact with each other. For example, ActiveX allows your browser to open a spreadsheet or display your personal account from a backend database. ActiveX components might also contain other components such as Java applets, or files such as .exe and .zip files.

When you visit an ActiveX-enabled website, the site prompts you to download ActiveX controls to your computer. Microsoft provides a pop-up message displaying the name of the company or programmer who authenticated the ActiveX code that is offered for download. If you trust the source of the code, you can proceed to download the controls. If you distrust the source, you can refuse them.

If you download an ActiveX control to your computer, it can then do whatever its creator designed it to do. If it is malicious code, it can now reformat your hard drive, delete all your files, send all your personal e-mail to your boss, and so on.

Java Applets

Serving a similar purpose as ActiveX, Java applets also increase the functionality of web pages by allowing them to interact with other programs. You download Java applets to a Java Virtual Machine (VM) on your computer. In the initial version of Java, the VM did not allow the applets to interact with other resources on your computer. Starting with Java 1.1, some of these restrictions were relaxed to provide greater functionality. As a result, Java applets can now access local resources outside the VM. Because an attacker can program Java applets to operate outside the VM, they pose the same security threat as ActiveX controls do.

EXE Files

If you download and run an executable file (that is, a file with a .exe extension) obtained off the Web, you cannot guarantee that the file is uncontaminated. Even if you trust the site from which you downloaded it, it is possible that somebody sniffing download requests from that site has intercepted your request and responded with a doctored .exe file that contains malicious code.

ZIP Files

A zip file (that is, a file with a .zip extension) is a type of file containing one or more compressed files. The danger of downloading a .exe file presented in the previous section about .exe files applies to .zip files, because a .zip file can contain one or more .exe files.

Example: Blocking Java Applets and .exe Files

In this example, you block any HTTP traffic containing Java applets and .exe files in packets arriving at an Untrust zone interface.

WebUI

Screening > Screen (Zone: Untrust): Select **Block Java Component** and **Block EXE Component**, then click **Apply**.

CLI

```
set zone untrust screen component-block jar
set zone untrust screen component-block exe
save
```

Chapter 6

Suspicious Packet Attributes

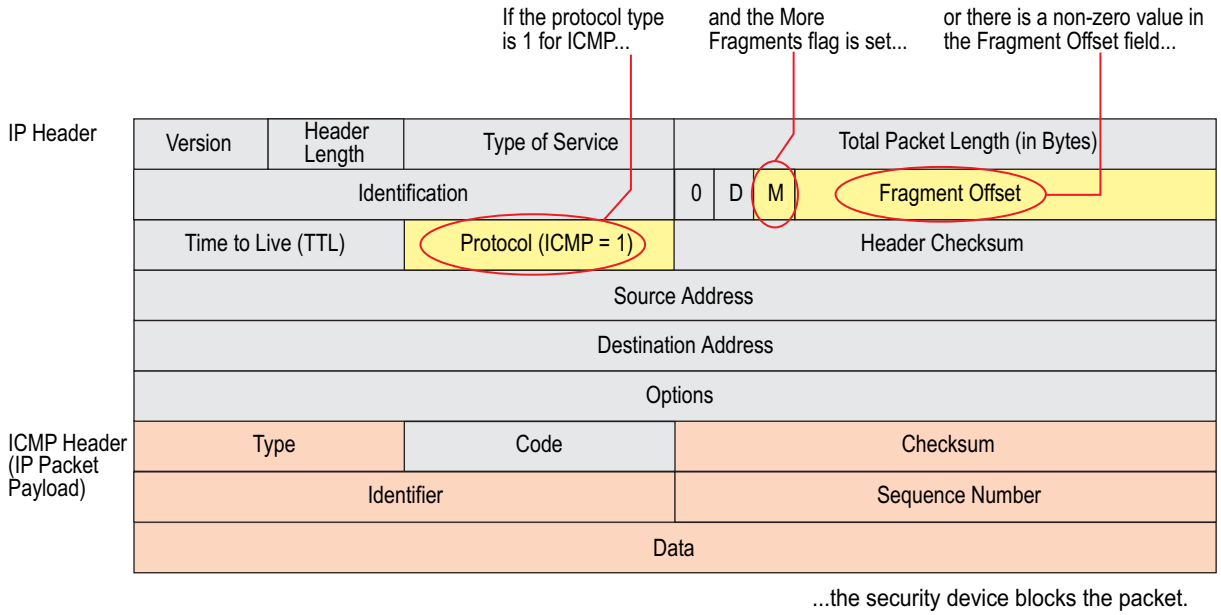
As shown in the other chapters in this volume, attackers can craft packets to perform reconnaissance or launch denial-of-service (DoS) attacks. Sometimes it is unclear what the intent of a crafted packet is, but the very fact that it is crafted suggests that its being put to some kind of insidious use. All of the SCREEN options presented in this chapter block suspicious packets that might contain hidden threats:

- "ICMP Fragments" on page 152
- "Large ICMP Packets" on page 153
- "Bad IP Options" on page 154
- "Unknown Protocols" on page 155
- "IP Packet Fragments" on page 156
- "SYN Fragments" on page 157

ICMP Fragments

Internet Control Message Protocol (ICMP) provides error reporting and network probe capabilities. Because ICMP packets contain very short messages, there is no legitimate reason for ICMP packets to be fragmented. If an ICMP packet is so large that it must be fragmented, something is amiss. When you enable the ICMP Fragment Protection SCREEN option, the security device blocks any ICMP packet that has the More Fragments flag set or that has an offset value indicated in the offset field.

Figure 55: Blocking ICMP Fragments



To block fragmented ICMP packets, do either of the following, where the specified security zone is the one from which the fragments originate:

WebUI

Screening > Screen (Zone: select a zone name): Select **ICMP Fragment Protection**, then click **Apply**.

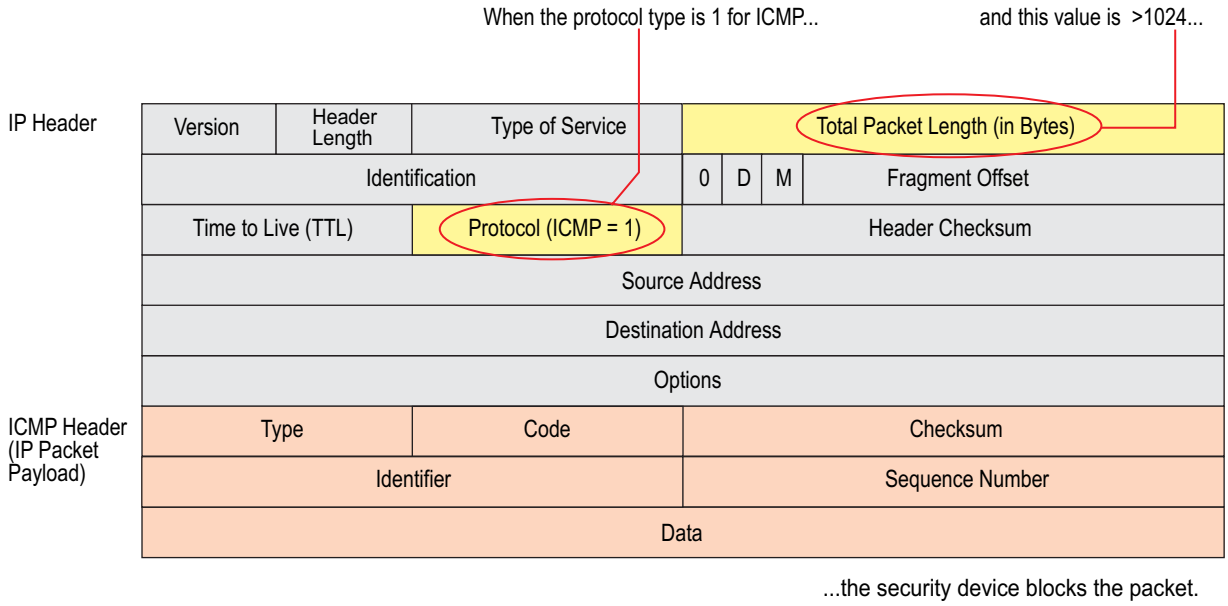
CLI

set zone *zone* screen icmp-fragment

Large ICMP Packets

As stated in "ICMP Fragments" on page 152, Internet Control Message Protocol (ICMP) provides error reporting and network probe capabilities. Because ICMP packets contain very short messages, there is no legitimate reason for large ICMP packets. If an ICMP packet is unusually large, something is wrong. For example, the Loki program uses ICMP as a channel for transmitting covert messages. The presence of large ICMP packets might expose a compromised machine acting as a Loki agent. It also might indicate some other kind of questionable activity.

Figure 56: Blocking Large ICMP Packets



When you enable the Large Size ICMP Packet Protection SCREEN option, the security device checks drops ICMP packets with a length greater than 1024 bytes.

To block large ICMP packets, do either of the following, where the specified security zone is the one from which the ICMP packets originate:

WebUI

Screening > Screen (Zone: select a zone name): Select **Large Size ICMP Packet (Size > 1024) Protection**, then click **Apply**.

CLI

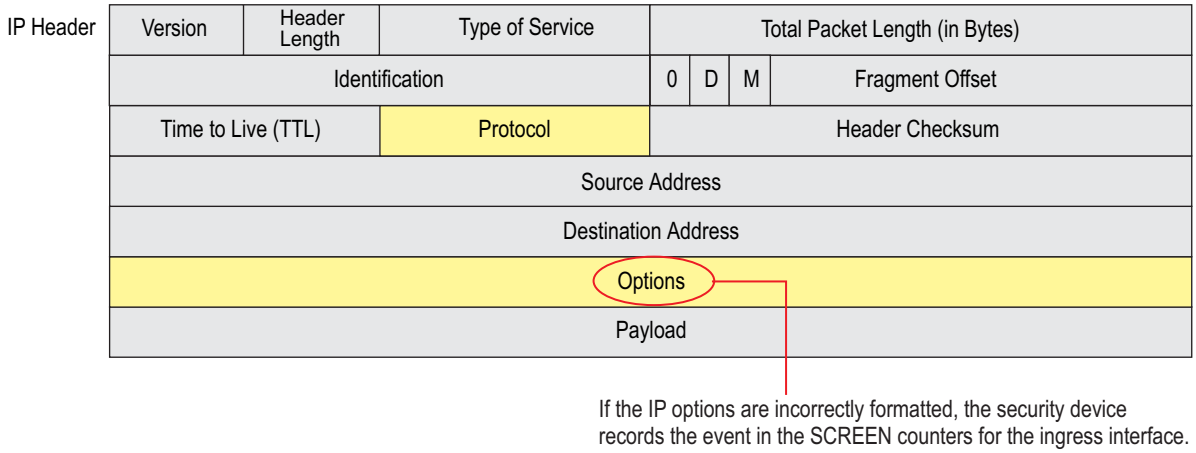
set zone *zone* screen icmp-large

Bad IP Options

The Internet Protocol standard RFC 791, *Internet Protocol*, specifies a set of eight options that provide special routing controls, diagnostic tools, and security. Although the original, intended uses for these options served worthy ends, people have figured out ways to twist these options to accomplish less commendable objectives. (For a summary of the exploits that attackers can initiate from IP options, see "Network Reconnaissance Using IP Options" on page 10.)

Either intentionally or accidentally, attackers sometimes configure IP options incorrectly, producing either incomplete or malformed fields. Regardless of the intentions of the person who crafted the packet, the incorrect formatting is anomalous and potentially harmful to the intended recipient.

Figure 57: Incorrectly Formatted IP Options



When you enable the Bad IP Option Protection SCREEN option, the security device blocks packets when any IP option in the IP packet header is incorrectly formatted. The security device records the event in the event log.

To detect and block IP packets with incorrectly formatted IP options, do either of the following, where the specified security zone is the one from which the packets originate:

WebUI

Screening > Screen (Zone: select a zone name): Select **Bad IP Option Protection**, then click **Apply**.

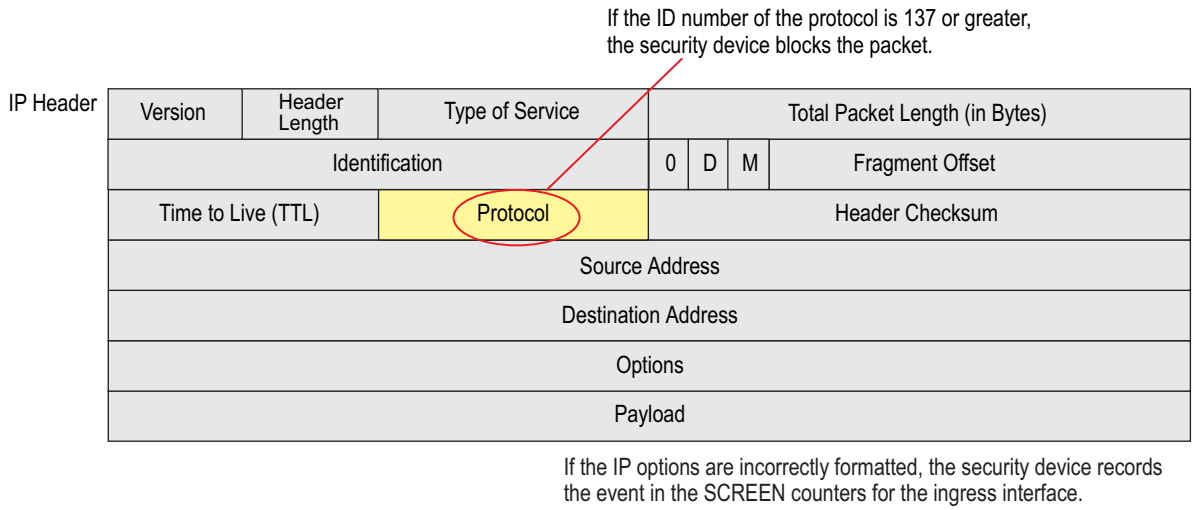
CLI

set zone *zone* screen ip-bad-option

Unknown Protocols

These protocol types with ID numbers of 137 or greater are reserved and undefined at this time. Precisely because these protocols are undefined, there is no way to know in advance if a particular unknown protocol is benign or malicious. Unless your network makes use of a nonstandard protocol with an ID number of 137 or greater, a cautious stance is to block such unknown elements from entering your protected network.

Figure 58: Unknown Protocols



When you enable the Unknown Protocol Protection SCREEN option, the security device drops packets when the protocol field is contains a protocol ID number of 137 or greater.

To drop packets using an unknown protocol, do either of the following, where the specified security zone is the one from which the packets originate:

WebUI

Screening > Screen (Zone: select a zone name): Select **Unknown Protocol Protection**, then click **Apply**.

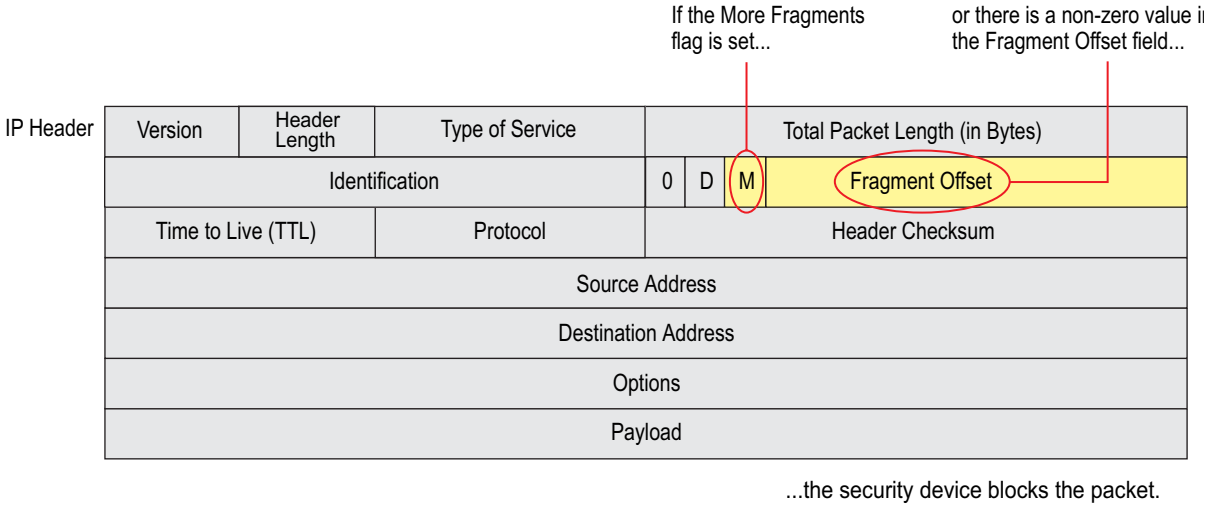
CLI

set zone *zone* screen unknown-protocol

IP Packet Fragments

As packets traverse different networks, it is sometimes necessary to break a packet into smaller pieces (fragments) based upon the maximum transmission unit (MTU) of each network. IP fragments might contain an attacker's attempt to exploit the vulnerabilities in the packet reassembly code of specific IP stack implementations. When the victim receives these packets, the results can range from processing the packets incorrectly to crashing the entire system.

Figure 59: IP Packet Fragments



When you enable the security device to deny IP fragments on a security zone, the device blocks all IP packet fragments that it receives at interfaces bound to that zone.

To drop fragmented IP packets, do either of the following, where the specified security zone is the one from which the fragments originate:

WebUI

Screening > Screen (Zone: select a zone name): Select **Block Fragment Traffic**, then click **Apply**.

CLI

set zone *zone* screen block-frag

SYN Fragments

The Internet Protocol (IP) encapsulates a Transmission Control Protocol (TCP) SYN segment in the IP packet that initiates a TCP connection. Because the purpose of this packet is to initiate a connection and invoke a SYN/ACK segment in response, the SYN segment typically does not contain any data. Because the IP packet is small, there is no legitimate reason for it to be fragmented. A fragmented SYN packet is anomalous, and as such suspect. To be cautious, block such unknown elements from entering your protected network.

When you enable the SYN Fragment Detection SCREEN option, the security device detects packets when the IP header indicates that the packet has been fragmented and the SYN flag is set in the TCP header. The security device records the event in the SCREEN counters list for the ingress interface.

To drop IP packets containing SYN fragments, do either of the following, where the specified security zone is the one from which the packets originate:

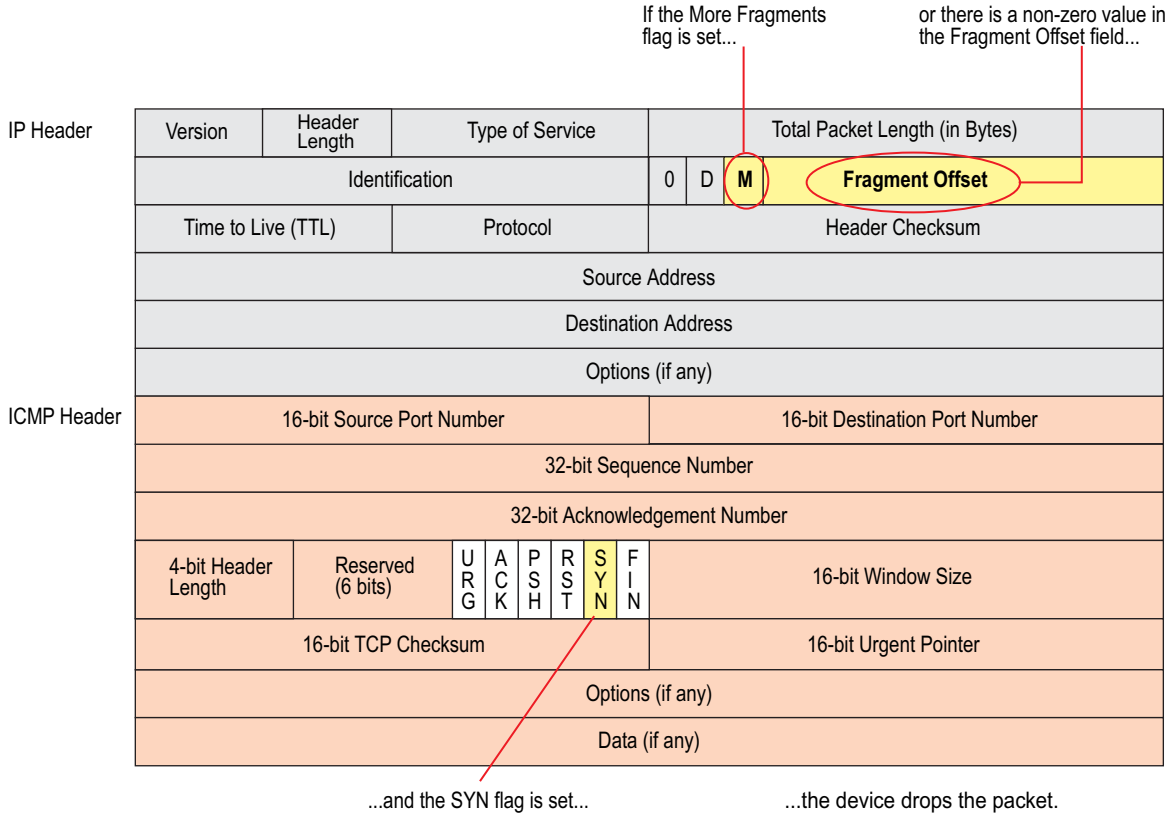
WebUI

Screening > Screen (Zone: select a zone name): Select **SYN Fragment Protection**, then click **Apply**.

CLI

set zone *zone* screen syn-frag

Figure 60: SYN Fragments



Appendix A

Contexts for User-Defined Signatures

The context defines the location in the packet where the Deep Inspection (DI) module searches for a signature matching the attack object pattern. When defining a stateful signature attack object, you can specify any of the contexts in the following lists. After you define an attack object, you must then put it in a user-defined attack object group for use in policies.

NOTE: A user-defined attack object group can only contain user-defined attack objects. You cannot mix predefined and user-defined attack objects in the same attack object group.

When the DI module examines traffic for TCP stream signatures, it does so without regard for contexts. TCP stream signatures look for patterns anywhere in any kind of TCP traffic regardless of the application protocol in use. Stream signatures are defined on NetScreen-5000 and 2000 series systems only. Stream256, however looks for patterns in the first 256 bytes of data.

Table 1: Contexts for User-Defined Signatures

Protocol	Context	Description (Sets the Context As...)
AIM	aim-chat-room-desc	The description of a chat room in an America Online Instant Messenger (AIM) or ICQ (I Seek You) session.
	aim-chat-room-name	The name of a chat room in an AIM or ICQ session.
	aim-get-file	The name of a file that a user is transferring from a peer.
	aim-nick-name	The nickname of an AIM or ICQ user.
	aim-put-file	The name of a file that a user is transferring to a peer.
	aim-screen-name	The screen name of an AIM or ICQ user.
DNS	dns-cname	The CNAME (canonical name) in a Domain Name System (DNS) request or response, as defined in RFC 1035, *Domain Names—Implementation and Specification*).
FTP	ftp-command	One of the FTP commands specified in RFC 959, *File Transfer Protocol (FTP)*.
	ftp-password	An FTP login password.
	ftp-pathname	A directory or file name in any FTP command.
	ftp-username	The name that a user enters when logging in to an FTP server.
Gnutella	gnutella-http-get-filename	The name of a file that a Gnutella client intends to retrieve.

Protocol	Context	Description (Sets the Context As...)
HTTP	http-authorization	The username and password decoded from an Authorization: Basic header in an HyperText Transfer Protocol (HTTP) request, as specified in RFC 1945, *HyperText Transfer Protocol—HTTP/1.0*.
	http-header-user-agent	The user-agent field in the header of an HTTP request. (When users visit a website, they provide information about their browsers in this field.)
	http-request	An HTTP request line.
	http-status	The status line in an HTTP reply. (The status line is a three-digit code that a webserver sends a client to communicate the state of a connection. For example, 401 means "Unauthorized" and 404 means "Not found".)
	http-text-html	The text, or HyperText Markup Language (HTML) data, in an HTTP transaction.
	http-url	The uniform resource locator (URL) in an HTTP request as it appears in the data stream.
	http-url-parsed	A "normalized" text string decoded from a unicode string that comprises a URL used in HTTP.
	http-url-variable-parsed	A decoded common gateway interface (CGI) variable in the URL of an HTTP-GET request.
IMAP	imap-authenticate	an argument in an Internet Mail Access Protocol (IMAP) AUTHENTICATE command. The argument indicates the type of authentication mechanism that the IMAP client proposes to the server. Examples are KERBEROS_V4, GSSAPI (see RFC 1508, *Generic Security Service Application Program Interface*), and SKEY. For information about IMAP, see RFC 1730, *Internet Message Access Protocol - Version 4*, and RFC 1731, *IMAP4 Authentication Mechanisms*.
	imap-login	Either the username or plaintext password in an IMAP LOGIN command.
	imap-mailbox	The mailbox text string in an IMAP SELECT command.
	imap-user	The username in an IMAP LOGIN command.
MSN Messenger	msn-display-name	The display name of a user in a Microsoft Network (MSN) Instant Messaging session.
	msn-get-file	The name of a file that a client is downloading from a peer.
	msn-put-file	The name of a file that a client is sending to a peer.
	msn-sign-in-name	The screen name (login name) of an MSN Instant Messaging user.
POP3	pop3-auth	The AUTH command in a Post Office Protocol, version 3 (POP3) session. For information about POP3, see RFC 1939, *Post Office Protocol—Version 3*.
	pop3-header-from	The text string in the "From:" header of an email in a POP3 transaction.
	pop3-header-line	The text string in any header line of an email in a POP3 transaction.
	pop3-header-subject	The text string in the "Subject:" header of an email in a POP3 transaction.
	pop3-header-to	The text string in the "To:" header of an email in a POP3 transaction.
	pop3-mime-content-filename	The content file name of a Multipurpose Internet Mail Extensions (MIME) attachment in a POP3 session.
	pop3-user	The username in a POP3 session.
SMB	smb-account-name	The name of a Server Message Blocks (SMB) account in a SESSION_SETUP_ANDX request in an SMB session.
	smb-connect-path	The connect path in the TREE_CONNECT_ANDX request in an SMB session.
	smb-connect-service	The name of the connect service in the TREE_CONNECT_ANDX request in an SMB session.
	smb-copy-filename	The name of a file in a COPY request in an SMB session.

Protocol	Context	Description (Sets the Context As...)
	smb-delete-filename	The name of a file in a DELETE request in an SMB session.
	smb-open-filename	The name of a file in the NT_CREATE_ANDX and OPEN_ANDX requests in an SMB session.
SMTP	smtp-from	The text string in a "MAIL FROM" command line in a Simple Mail Transfer Protocol (SMTP) session, as described in RFC 2821, *Simple Mail Transfer Protocol*.
	smtp-header-from	The text string in the "From:" header in an SMTP session.
	smtp-header-line	The text string in any header line in an SMTP session.
	smtp-header-subject	The text string in the "Subject:" header in an SMTP session.
	smtp-header-to	The text string in the "To:" header in an SMTP session.
	smtp-mime-content-filename	The content file name of a Multipurpose Internet Mail Extensions (MIME) attachment in an SMTP session.
	smtp-rcpt	The text string in a "RCPT TO" command line in an SMTP session.
–	stream256	The first 256 bytes of a reassembled, normalized TCP data stream.
Yahoo! Messenger	ymsg-alias	The alternate identifying name associated with the main username of a Yahoo! Instant Messaging user.
	ymsg-chatroom-message	The text in messages exchanged in a Yahoo! Instant Messaging chatroom.
	ymsg-chatroom-name	The name of a Yahoo! Instant Messaging chatroom.
	ymsg-nickname	The nickname of a Yahoo! Instant Messaging user.
	ymsg-p2p-get-filename-url	The location of a file on a Yahoo! Instant Messaging peer's machine from which it can be downloaded.
	ymsg-p2p-put-filename-url	The location of a file on a Yahoo! Instant Messaging peer's machine to which it can be downloaded.

Index

A

ActiveX controls, blocking 148
address sweep 8
agents, zombie 27, 29
aggressive aging ..... 30 to 32
AIM 113
ALG 55
America Online Instant Messaging
    See AIM
Application Layer Gateway
    See ALG
attack actions ..... 120 to 127
    close 120
    close client 121
    close server 120
    drop 121
    drop packet 121
    ignore 121
    none 121
attack object database ..... 102 to 110
    auto notification and manual update 104, 107
    automatic update 104, 106
    changing the default URL 109
    immediate update 103, 105
    manual update 104, 108
attack object groups 117
    applied in policies 111
    changing severity 117
    Help URLs 114
    logging 130
    severity levels 117
attack objects ..... 99, 110 to 116
    brute force 127, 128
    disabling 119
    negation 143
    protocol anomalies 116, 142
    re-enabling 120
    stateful signatures 115
    stream signatures 116
    TCP stream signatures 140
attack protection
    policy level 4
    security zone level 4
attacks
    common objectives 1
    detection and defense options ..... 2 to 4
    DOS ..... 27 to 51
    ICMP
        floods 46
        fragments 152
    IP packet fragments 156
    Land 48
    large ICMP packets 153
    Ping of Death 49
    session table floods 17, 28
    stages of 2
    SYN floods ..... 34 to 39
    SYN fragments 157
    Teardrop 50
    UDP floodsUDP floods 47
    unknown MAC addresses 39
    unknown protocols 155
    WinNuke 51
AV objects
    timeout 73
AV scanning ..... 57 to 72
    AV resources per client 69
    decompression 74
    fail-mode 69
    file extensions 75
    FTP 58
    HTTP 59
    HTTP keep-alive 70
    HTTP trickling 71
    HTTP webmail 61
    IMAP 61
    MIME 60
    POP3 61
    SMTP 63
    subscription 64

B

brute force
    attack actions 127
brute force attack objects 128

C

Chargen 112
content filtering ..... 53 to 96
cookies, SYN 44

D

DDoS 27
decompression, AV scanning 74
Deep Inspection (DI) ..... 117 to 140
  attack actions ..... 120 to 127
  attack object database ..... 102 to 110
  attack object groups 117
  attack object negation 143
  attack objects 99
  changing severity 117
  context I
  custom attack objects 136
  custom services ..... 132 to 136
  custom signatures ..... 137 to 140
  disabling attack objects 119
  license keys 100
  logging attack object groups 130
  overview 98
  protocol anomalies 116
  re-enabling attack objects 120
  regular expressions ..... 137 to 138
  signature packs 102
  stateful signatures 115
  stream signatures 116
Denial-of-Service
  See DoS
DHCP 112
Discard 112
DNS 112
DoS
  firewall ..... 28 to 33
  network ..... 34 to 48
  OS-specific ..... 49 to 51
  session table floods 17, 28
DoS attacks ..... 27 to 51
drop-no-rpf-route 19
dynamic packet filtering 3

E

Echo 112
evasion ..... 15 to 25
exe files, blocking 148
exploits
  See attacks

F

fail-mode 69
file extensions, AV scanning 75
FIN scans 15
FIN without ACK flag 13
Finger 112
floods
  ICMP 46
  session table 28
  SYN ..... 34 to 39, 44
  UDP 47
fragment reassembly ..... 54 to 57

G

Gopher 112

H

high-watermark threshold 30
HTTP
  blocking components ..... 147 to 149
  keep-alive 70
  session timeout 31
  trickling 71

I

ICMP 112
  fragments 152
  large packets 153
ICMP floods 46
IDENT 112
inspections 3
Instant Messaging 113
  AIM 113
  IRC 113
  MSN Messenger 113
  Yahoo! Messenger 113
IP
  packet fragments 156
IP options ..... 10 to 11
  attributes ..... 10 to 11
  incorrectly formatted 154
  loose source route ..... 10, 23 to 25
  record route 10, 11
  security 10, 11
  source route 23
  stream ID 10, 11
  strict source route ..... 11, 23 to 25
  timestamp 11
IP spoofing ..... 18 to 23
  drop-no-rpf-route 19
  Layer 2 ..... 19, 22
  Layer 3 ..... 18, 20
IRC 113

J

Java applets, blocking....148

L

Land attacks....48
LDAP....112
license keys
  advanced mode....100
  attack pattern update....100
logging
  attack object groups....130
loose source route IP option....10, 23 to 25
low-watermark threshold....31
LPR spooler....112

M

malicious URL protection....54 to 57
Microsoft Network Instant Messenger
  See MSN Instant Messenger
Microsoft-Remote Procedure Call
  See MS-RPC
MIME, AV scanning....60
MSN Messenger....113
MS-RPC....114

N

negation, Deep Inspection (DI)....143
NetBIOS....114
NFS....112
NNTP....113
NTP....113

O

operating systems, probing hosts for....12 to 14

P

P2P....114
  BitTorrent....114
  DC....114
  eDonkey....114
  FastTrack....114
  Gnutella....114
  KaZaa....114
  MLdonkey....114
  Skype....114
  SMB....114
  WinMX....114
Peer-to-Peer
  See P2P
Ping of Death....49
policies
  context....102
  core section....17, 100
  web filtering....94
port scan....9
Portmapper....113
probes
  network....8
  open ports....9
  operating systems....12, 14
protocol anomalies....116
  ALGs....114
  basic network protocols....112
  configuring parameters....142
  Instant Messaging applications....113
  P2P applications....114
  supported protocols....112 to 115

R

RADIUS....113
reconnaissance....7 to 25
  address sweep....8
  FIN scans....15
  IP options....10
  port scan....9
  SYN and FIN flags set....12
  TCP packet without flags....14
record route IP option....10, 11
regular expressions....137 to 138
rexec....113
RFCs
  1038, *Revised IP Security Option*....10
  791, *Internet Protocol*....10
  793, *Transmission Control Protocol*....13
rlogin....113
rsh....113
RTSP....113

S

SCREEN
  address sweep....8
  bad IP options, drop....154
  drop unknown MAC addresses....39
  FIN with no ACK....15
  FIN without ACK flag, drop....13
  ICMP
    fragments, block....152
  ICMP floods....46
  IP options....10
  IP packet fragments, block....156
  IP spoofing....18 to 23
  Land attacks....48
  large ICMP packets, block....153
  loose source route IP option, detect....25
  Ping of Death....49
  port scan....9
  source route IP option, deny....25
  strict source route IP option, detect....25

SCREEN (continued)
SYN and FIN flags set 12
SYN floods ..... 34 to 39
SYN fragments, detect 157
SYN-ACK-ACK proxy floods 32
TCP packet without flags, detect 14
Teardrop 50
UDP floods 47
unknown protocols, drop 155
VLAN and MGT zones 2
WinNuke attacks 51
security IP option 10, 11
Server Message Block
See SMB
services
custom 132
session limits ..... 28 to 30
destination-based 29, 30
source-based 28, 29
session table floods 17, 28
session timeout
HTTP 31
session timeouts
TCP 31
UDP 31
signature packs, DI 102
signatures
stateful 115
SMB
NetBIOS 114
SNMPTRAP 113
SSH 113
SSL 113
stateful 3
inspection 3
signatures 115
stream ID IP option 10, 11
stream signatures 116
strict source route IP option ..... 11, 23 to 25
SurfControl 81, 89
SYN and FIN flags set 12
SYN checking ..... 15, 15 to 18
asymmetric routing 16
reconnaissance hole 17
session interruption 17
session table floods 17
SYN cookies 44
SYN floods ..... 34 to 39
alarm threshold 38
attack threshold 37
attacks 34
destination threshold 38
drop unknown MAC addresses 39
queue size 39
source threshold 38
SYN cookies 44
threshold 35
timeout 39
SYN fragments 157
SYN-ACK-ACK proxy floods 32
syslog 113

T

TCP
packet without flags 14
session timeouts 31
stream signatures 140
Teardrop attacks 50
Telnet 113
TFTP 113
three-way handshakes 34
threshold
low-watermark 31
thresholds
high-watermark 30
timestamp IP option 11
Transparent mode
drop unknown MAC addresses 39

U

UDP
session timeouts 31
unknown protocols 155

V

VNC 113

W

web filtering ..... 89 to 96
applying profiles to policies 86
blocked URL message 93
blocked URL message type 92
cache 88
communication timeout 92
entering a context 82
integrated 81
policy-level application 94
profiles 84
redirect 89
routing 94
server status 94
servers per vsys 90
SurfControl CPA servers 81
SurfControl SCFP 91
SurfControl server name 92
SurfControl server port 92
SurfControl servers 88
URL categories 82
Websense server name 92
Websense server port 92
Whois 113
WinNuke attacks 51

Y

Yahoo! Messenger 113

Z

zip files, blocking 148
zombie agents 27, 29