

ASA/PIX 7.2: Block Certain Websites (URLs) Using Regular Expressions with MPF Configuration Examples

Document ID: 100513

Contents

Introduction

Prerequisites

- Requirements
- Components Used
- Related Products
- Conventions

Background Information

- Modular Policy Framework Overview
- Regular Expression

Configure

- Network Diagram
- Configurations
- ASA CLI Configuration
- ASA Configuration 7.2(x) with ASDM 5.2

Verify

Troubleshoot

Related Information

Introduction

This document describes how to configure the Cisco Security Appliances ASA/PIX 7.2 with Regular Expressions with Modular Policy Framework (MPF) in order to block certain websites (URLs).

Note: This configuration does not block all application downloads. For reliable file blocks, a dedicated appliance, such as Websense, etc., or module, such as the CSC module for the ASA, must be used.

HTTPS filtering is not supported on ASA. ASA cannot do deep packet inspection or inspection based on regular expression for HTTPS traffic because, in HTTPS, the content of packet is encrypted (ssl).

Prerequisites

Requirements

This document assumes that Cisco Security Appliance is configured and works properly.

Components Used

- Cisco 5500 Series Adaptive Security Appliance (ASA) that runs Software Version 7.2(2)
- Cisco Adaptive Security Device Manager (ASDM) Version 5.2(2) for ASA 7.2(2)

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, make sure

that you understand the potential impact of any command.

Related Products

This configuration can also be used with the Cisco 500 Series PIX that runs Software Version 7.2(2).

Conventions

Refer to the Cisco Technical Tips Conventions for more information on document conventions.

Background Information

Modular Policy Framework Overview

MPF provides a consistent and flexible way to configure security appliance features. For example, you can use MPF to create a timeout configuration that is specific to a particular TCP application, as opposed to one that applies to all TCP applications.

MPF supports these features:

- TCP normalization, TCP and UDP connection limits and timeouts, and TCP sequence number randomization
- CSC
- Application inspection
- IPS
- QoS input policing
- QoS output policing
- QoS priority queue

The configuration of the MPF consists of four tasks:

1. Identify the Layer 3 and 4 traffic to which you want to apply actions. Refer to [Identifying Traffic Using a Layer 3/4 Class Map](#) for more information.
2. (Application inspection only) Define special actions for application inspection traffic. Refer to [Configuring Special Actions for Application Inspections](#) for more information.
3. Apply actions to the Layer 3 and 4 traffic. Refer to [Defining Actions Using a Layer 3/4 Policy Map](#) for more information.
4. Activate the actions on an interface. Refer to [Applying a Layer 3/4 Policy to an Interface Using a Service Policy](#) for more information.

Regular Expression

A regular expression matches text strings either literally as an exact string, or with metacharacters, so you can match multiple variants of a text string. You can use a regular expression to match the content of certain application traffic; for example, you can match a URL string inside an HTTP packet.

Note: Use **Ctrl+V** to escape all the special characters in the CLI, such as a question mark (?) or tab. For example, type **d[Ctrl+V]g** to enter **d?g** in the configuration.

In order to create a regular expression, use the **regex** command, which can be used for various features that require text matching. For example, you can configure special actions for application inspection with Modular Policy Framework with an inspection policy map (see the policy map type inspect command). In the

inspection policy map, you can identify the traffic you want to act upon if you create an inspection class map that contains one or more **match** commands, or you can use **match** commands directly in the inspection policy map. Some **match** commands let you identify text in a packet with a regular expression; for example, you can match URL strings inside HTTP packets. You can group regular expressions in a regular expression class map (see the class-map type regex command).

Table 1 lists the metacharacters that have special meanings.

Character	Description	Notes
.	Dot	Matches any single character. For example, d.g matches dog, dag, dtg, and any word that contains those characters, such as doggonnit.
(exp)	Subexpression	A subexpression segregates characters from surrounding characters, so that you can use other metacharacters on the subexpression. For example, d(o a)g matches dog and dag, but do ag matches do and ag. A subexpression can also be used with repeat quantifiers to differentiate the characters meant for repetition. For example, ab(xy){3}z matches abxyxyxyz.
	Alternation	Matches either expression it separates. For example, dog cat matches dog or cat.
?	Question mark	A quantifier that indicates that there are 0 or 1 of the previous expression. For example, lo?se matches lse or lose. Note: You must enter Ctrl+V and then the question mark or else the help function is invoked.
*	Asterisk	A quantifier that indicates that there are 0, 1 or any number of the previous expression. For example, lo*se matches lse, lose, loose, and so on.
{x}	Repeat quantifier	Repeat exactly x times. For example, ab(xy){3}z matches abxyxyxyz.
{x,}	Minimum repeat quantifier	Repeat at least x times. For example, ab(xy){2,}z matches abxyxyz, abxyxyxyz, and so on.
[abc]	Character class	Matches any character in the brackets. For example, [abc]

		matches a, b, or c.
[^abc]	Negated character class	Matches a single character that is not contained within the brackets. For example, [^abc] matches any character other than a, b, or c. [^A-Z] matches any single character that is not an uppercase letter.
[a-c]	Character range class	Matches any character in the range. [a-z] matches any lowercase letter. You can mix characters and ranges: [abcq-z] matches a, b, c, q, r, s, t, u, v, w, x, y, z, and so does [a-cq-z]. The dash (-) character is literal only if it is the last or first character within the brackets: [abc-] or [-abc].
""	Quotation marks	Preserves trailing or leading spaces in the string. For example, " test" preserves the leading space when it looks for a match.
^	Caret	Specifies the beginning of a line.
\	Escape character	When used with a metacharacter, matches a literal character. For example, \[matches the left square bracket.
char	Character	When a character is not a metacharacter, matches the literal character.
\r	Carriage return	Matches a carriage return 0x0d.
\n	Newline	Matches a new line 0x0a.
\t	Tab	Matches a tab 0x09.
\f	Formfeed	Matches a form feed 0x0c.
\xNN	Escaped hexadecimal number	Matches an ASCII character with hexadecimal (exactly two digits).
\NNN	Escaped octal number	Matches an ASCII character as octal (exactly three digits). For example, the character 040 represents a space.

Configure

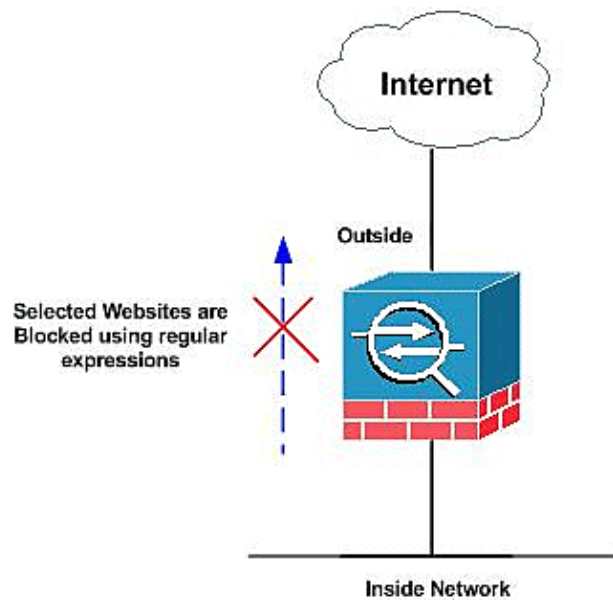
In this section, you are presented with the information to configure the features described in this document.

Note: Use the Command Lookup Tool (registered customers only) to obtain more information on the

commands used in this section.

Network Diagram

This document uses this network setup:



Configurations

This document uses these configurations:

- ASA CLI Configuration
- ASA Configuration 7.2(x) with ASDM 5.2

ASA CLI Configuration

ASA CLI Configuration

```
ciscoasa#show running-config
: Saved
:
ASA Version 7.2(2)
!
hostname ciscoasa
domain-name default.domain.invalid
enable password 8Ry2YjIyt7RRXU24 encrypted
names
!
interface Ethernet0/0
 nameif inside
 security-level 100
 ip address 10.1.1.1 255.255.255.0
!
interface Ethernet0/1
 nameif outside
 security-level 0
 ip address 192.168.1.5 255.255.255.0
!
interface Ethernet0/2
 nameif DMZ
 security-level 90
```

```
ip address 10.77.241.142 255.255.255.192
!
interface Ethernet0/3
shutdown
no nameif
no security-level
no ip address
!
interface Management0/0
shutdown
no nameif
no security-level
no ip address
!
passwd 2KFQnbNIdI.2KYOU encrypted
regex urllist1 ".*\.([Ee][Xx][Ee]|[Cc][Oo][Mm]|[Bb][Aa][Tt]) HTTP/1.[01]"

!--- Extensions such as .exe, .com, .bat to be captured and
!--- provided the http version being used by web browser must be either 1.0 or 1.1

regex urllist2 ".*\.([Pp][Ii][Ff]|[Vv][Bb][Ss]|[Ww][Ss][Hh]) HTTP/1.[01]"

!--- Extensions such as .pif, .vbs, .wsh to be captured
!--- and provided the http version being used by web browser must be either
!--- 1.0 or 1.1

regex urllist3 ".*\.([Dd][Oo][Cc]|[Xx][Ll][Ss]|[Pp][Pp][Tt]) HTTP/1.[01]"

!--- Extensions such as .doc(word), .xls(ms-excel), .ppt to be captured and provided
!--- the http version being used by web browser must be either 1.0 or 1.1

regex urllist4 ".*\.([Zz][Ii][Pp]|[Tt][Aa][Rr]|[Tt][Gg][Zz]) HTTP/1.[01]"

!--- Extensions such as .zip, .tar, .tgz to be captured and provided
!--- the http version being used by web browser must be either 1.0 or 1.1

regex domainlist1 "\.yahoo\.com"
regex domainlist2 "\.myspace\.com"
regex domainlist3 "\.youtube\.com"

!--- Captures the URLs with domain name like yahoo.com,
!--- youtube.com and myspace.com

regex contenttype "Content-Type"
regex applicationheader "application/*"

!--- Captures the application header and type of
!--- content in order for analysis

boot system disk0:/asa802-k8.bin
ftp mode passive
dns server-group DefaultDNS
domain-name default.domain.invalid

access-list inside_mpc extended permit tcp any any eq www
```

```
access-list inside_mpc extended permit tcp any any eq 8080
```

```
!--- Filters the http and port 8080  
!--- traffic in order to block the specific traffic with regular  
!--- expressions
```

```
pager lines 24  
mtu inside 1500  
mtu outside 1500  
mtu DMZ 1500  
no failover  
icmp unreachable rate-limit 1 burst-size 1  
asdm image disk0:/asdm-602.bin  
no asdm history enable  
arp timeout 14400  
route DMZ 0.0.0.0 0.0.0.0 10.77.241.129 1  
timeout xlate 3:00:00  
timeout conn 1:00:00 half-closed 0:10:00 udp 0:02:00 icmp 0:00:02  
timeout sunrpc 0:10:00 h323 0:05:00 h225 1:00:00 mgcp 0:05:00 mgcp-pat 0:05:00  
timeout sip 0:30:00 sip_media 0:02:00 sip-invite 0:03:00 sip-disconnect 0:02:00  
timeout uauth 0:05:00 absolute  
dynamic-access-policy-record DfltAccessPolicy  
http server enable  
http 0.0.0.0 0.0.0.0 DMZ  
no snmp-server location  
no snmp-server contact  
snmp-server enable traps snmp authentication linkup linkdown coldstart  
no crypto isakmp nat-traversal  
telnet timeout 5  
ssh timeout 5  
console timeout 0  
threat-detection basic-threat  
threat-detection statistics access-list  
!  
class-map type regex match-any DomainBlockList  
  match regex domainlist1  
  match regex domainlist2  
  match regex domainlist3
```

```
!--- Class map created in order to match the domain names  
!--- to be blocked
```

```
class-map type inspect http match-all BlockDomainsClass  
  match request header host regex class DomainBlockList
```

```
!--- Inspect the identified traffic by class  
!--- "DomainBlockList"
```

```
class-map type regex match-any URLBlockList  
  match regex urllist1  
  match regex urllist2  
  match regex urllist3  
  match regex urllist4
```

```
!--- Class map created in order to match the URLs  
!--- to be blocked
```

```

class-map inspection_default
  match default-inspection-traffic

class-map type inspect http match-all AppHeaderClass
  match response header regex contenttype regex applicationheader

!--- Inspect the captured traffic by regular
!--- expressions "content-type" and "applicationheader"

class-map httptraffic
  match access-list inside_mpc

!--- Class map created in order to match the
!--- filtered traffic by ACL

class-map type inspect http match-all BlockURLsClass
  match request uri regex class URLBlockList
!

!--- Inspect the identified traffic by class
!--- "URLBlockList"

!
policy-map type inspect dns preset_dns_map
  parameters
    message-length maximum 512
policy-map type inspect http http_inspection_policy
  parameters
    protocol-violation action drop-connection
  class AppHeaderClass
    drop-connection log
  match request method connect
    drop-connection log
  class BlockDomainsClass
    reset log
  class BlockURLsClass
    reset log

!--- Define the actions such as drop, reset or log
!--- in the inspection policy map

policy-map global_policy
  class inspection_default
    inspect dns preset_dns_map
    inspect ftp
    inspect h323 h225
    inspect h323 ras
    inspect netbios
    inspect rsh
    inspect rtsp
    inspect skinny
    inspect esmtp
    inspect sqlnet
    inspect sunrpc
    inspect tftp
    inspect sip
    inspect xdmcp

policy-map inside-policy

```

```

class httptraffic
  inspect http http_inspection_policy

!--- Map the inspection policy map to the class
!--- "httptraffic" under the policy map created for the
!--- inside network traffic

!
service-policy global_policy global
service-policy inside-policy interface inside

!--- Apply the policy to the interface inside where the
websites will be blocked

prompt hostname context
Cryptochecksum:e629251a7c37af205c289cf78629fc11
: end
ciscoasa#

```

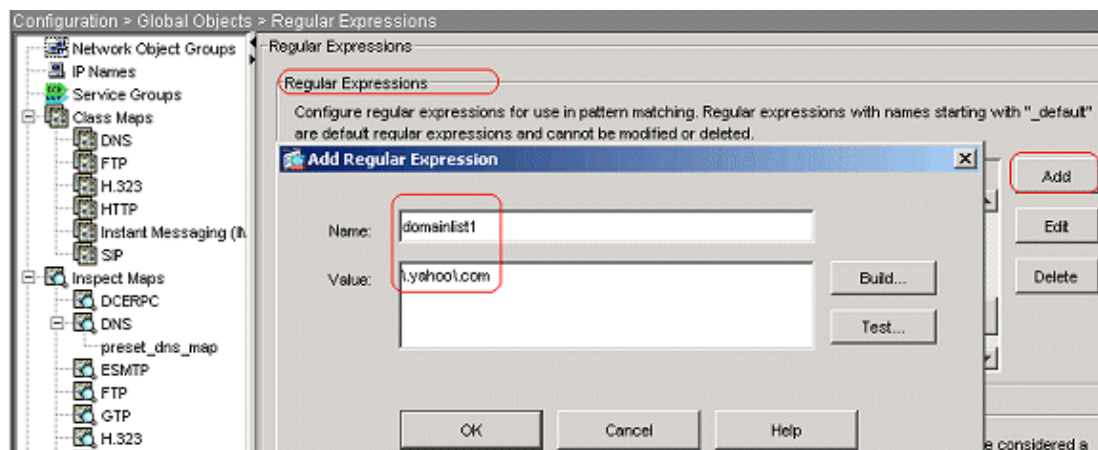
ASA Configuration 7.2(x) with ASDM 5.2

Complete these steps in order to configure the regular expressions and apply them to MPF to block the specific websites:

1. Create Regular Expressions

Choose **Configuration > Global Objects > Regular Expressions** and click **Add** under the Regular Expression tab in order to create regular expressions.

- a. Create a regular expression **domainlist1** in order to capture the domain name **yahoo.com**. Click **OK**.



- b. Create a regular expression **domainlist2** in order to capture the domain name **myspace.com**. Click **OK**.

Add Regular Expression

Name: domainlist2

Value: \.myspace\.com

Build... Test... OK Cancel Help

- c. Create a regular expression **domainlist3** in order to capture the domain name **youtube.com**. Click **OK**.

Add Regular Expression

Name: domainlist3

Value: \.youtube\.com

Build... Test... OK Cancel Help

- d. Create a regular expression **urllist1** in order to capture the file extensions such as **exe**, **com**, and **bat** provided that the http version used by the web browser must be either 1.0 or 1.1. Click **OK**.

Add Regular Expression

Name: urllist1

Value: ^\.([Ee][Xx][Ee][Cc][Oo][Mm][Bb][Aa][Tt]) HTTP/1.[01]

Build... Test... OK Cancel Help

- e. Create a regular expression **urllist2** in order to capture the file extensions, such as **pif**, **vbs**, and **wsh** provided that the HTTP version that is used by the web browser is either 1.0 or 1.1. Click **OK**.

Add Regular Expression

Name:

Value:

Build Test...

OK Cancel Help

- f. Create a regular expression **urllist3** in order to capture the file extensions, such as **doc**, **xls**, and **ppt** provided that the HTTP version that is used by the web browser is either 1.0 or 1.1. Click **OK**.

Add Regular Expression

Name:

Value:

Build... Test...

OK Cancel Help

- g. Create a regular expression **urllist4** in order to capture the file extensions, such as **zip**, **tar**, and **tgz** provided that the HTTP version that is used by the web browser is either 1.0 or 1.1. Click **OK**.

Add Regular Expression

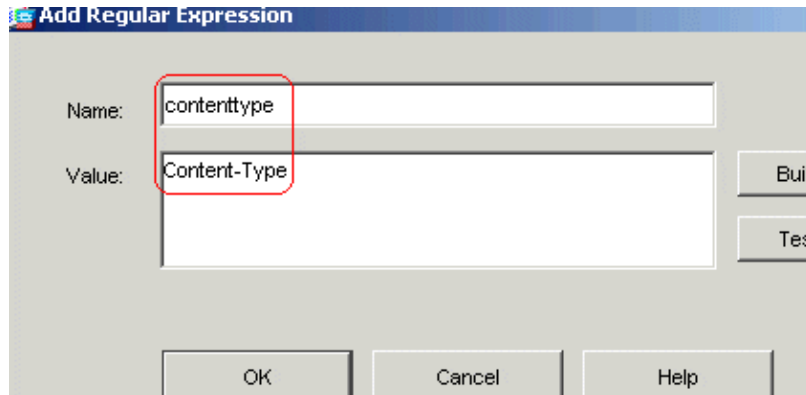
Name:

Value:

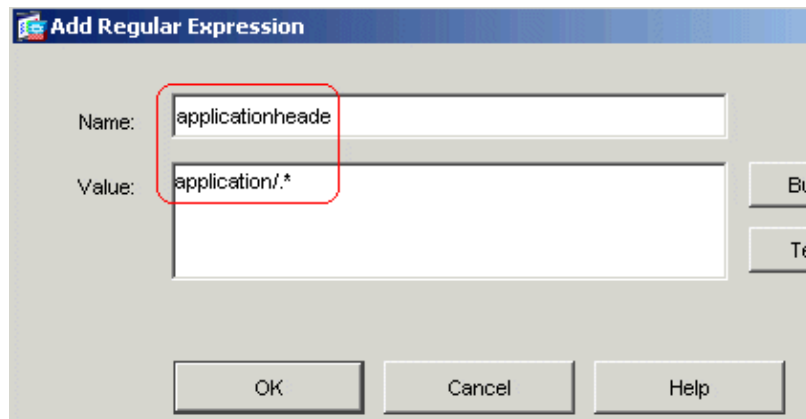
Build... Test...

OK Cancel Help

- h. Create a regular expression **contenttype** in order to capture the content type. Click **OK**.



- i. Create a regular expression **applicationheader** in order to capture the various application header. Click **OK**.



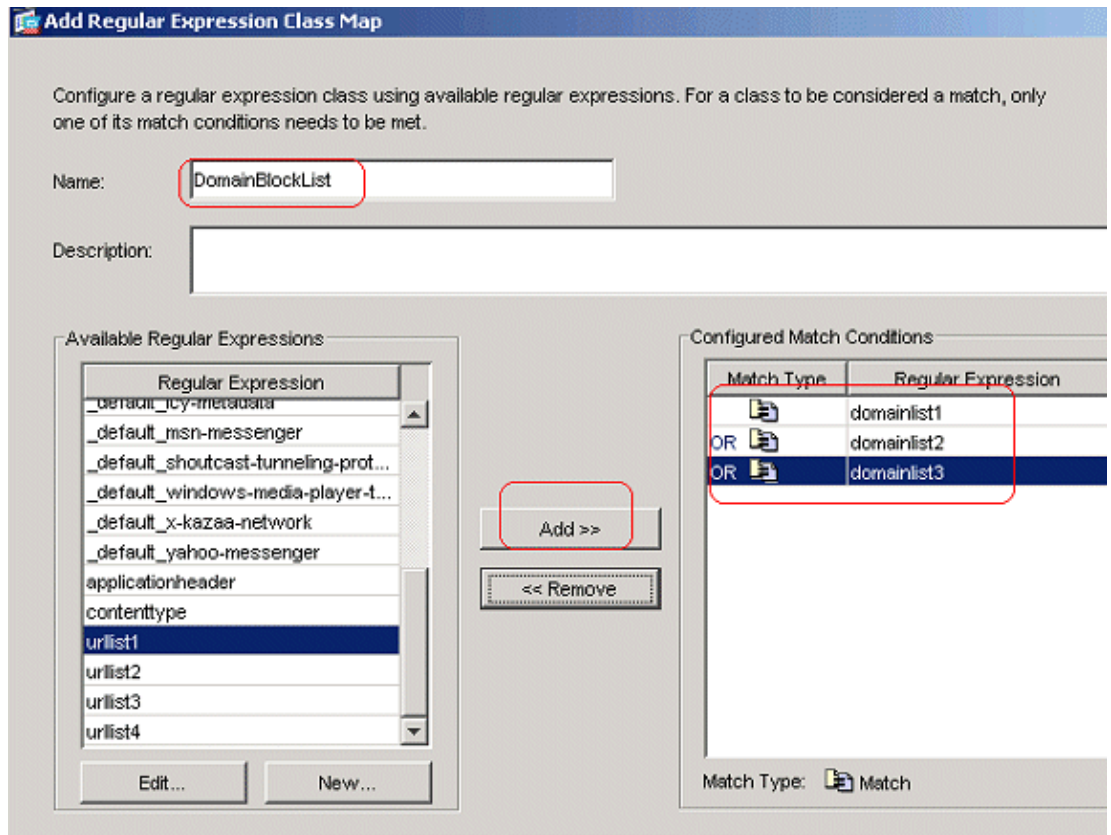
Equivalent CLI Configuration

ASA CLI Configuration
<pre> ciscoasa#configure terminal ciscoasa(config)#regex urllist1 ".*\.[Ee][Xx][Ee] [Cc][Oo][Mm] [Bb][Aa][Tt])\$ ciscoasa(config)#regex urllist2 ".*\.[Pp][Ii][Ff] [Vv][Bb][Ss][Ww][Ss][Hh])\$ ciscoasa(config)#regex urllist3 ".*\.[Dd][Oo][Cc][Xx][Ll][Ss][Pp][Pp][Tt])\$ ciscoasa(config)#regex urllist4 ".*\.[Zz][Ii][Pp][Tt][Aa][Rr][Tt][Gg][Zz])\$ ciscoasa(config)#regex domainlist1 "\.yahoo\.com" ciscoasa(config)#regex domainlist2 "\.myspace\.com" ciscoasa(config)#regex domainlist3 "\.youtube\.com" ciscoasa(config)#regex contenttype "Content-Type" ciscoasa(config)#regex applicationheader "application/*" </pre>

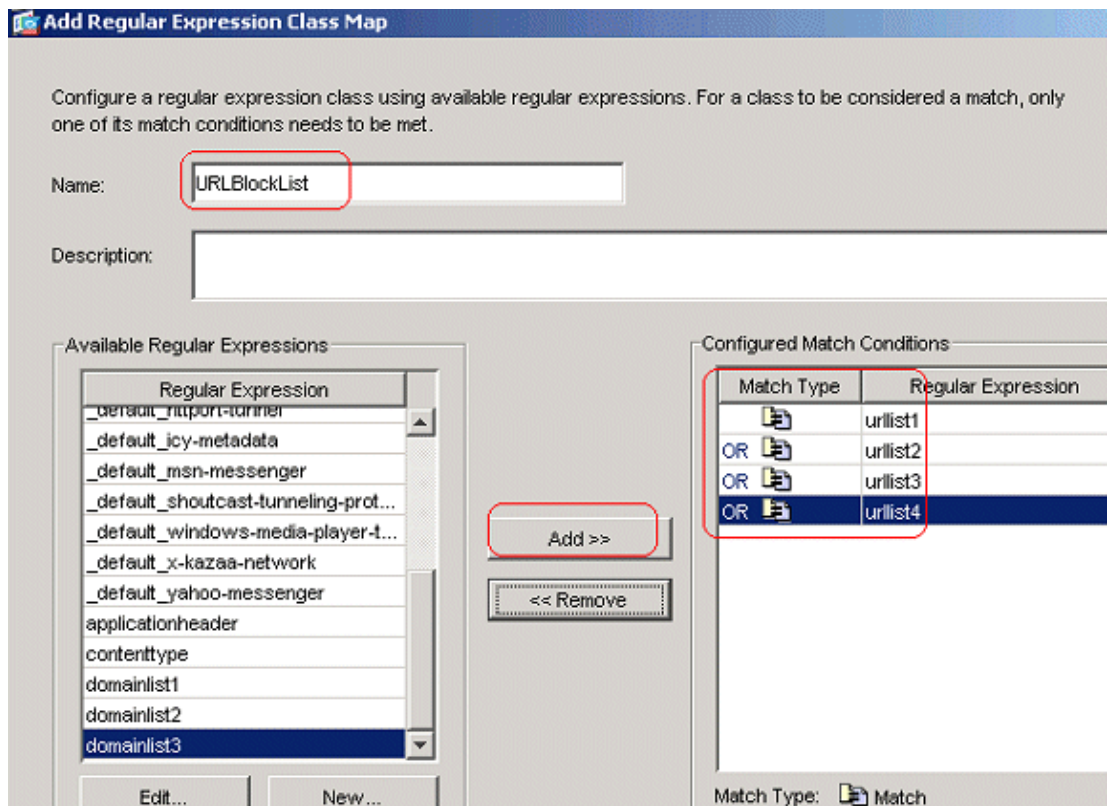
2. Create Regular Expression Classes

Choose **Configuration > Global Objects > Regular Expressions**, and click **Add** under the **Regular Expression Classes** tab in order to create the various classes.

- a. Create a regular expression class **DomainBlockList** in order to match any of the regular expressions: domainlist1, domainlist2, and domainlist3. Click **OK**.



- b. Create a regular expression class **URLBlockList** in order to match any of the regular expressions: urllist1, urllist2, urllist3, and urllist4. Click **OK**.



Equivalent CLI Configuration

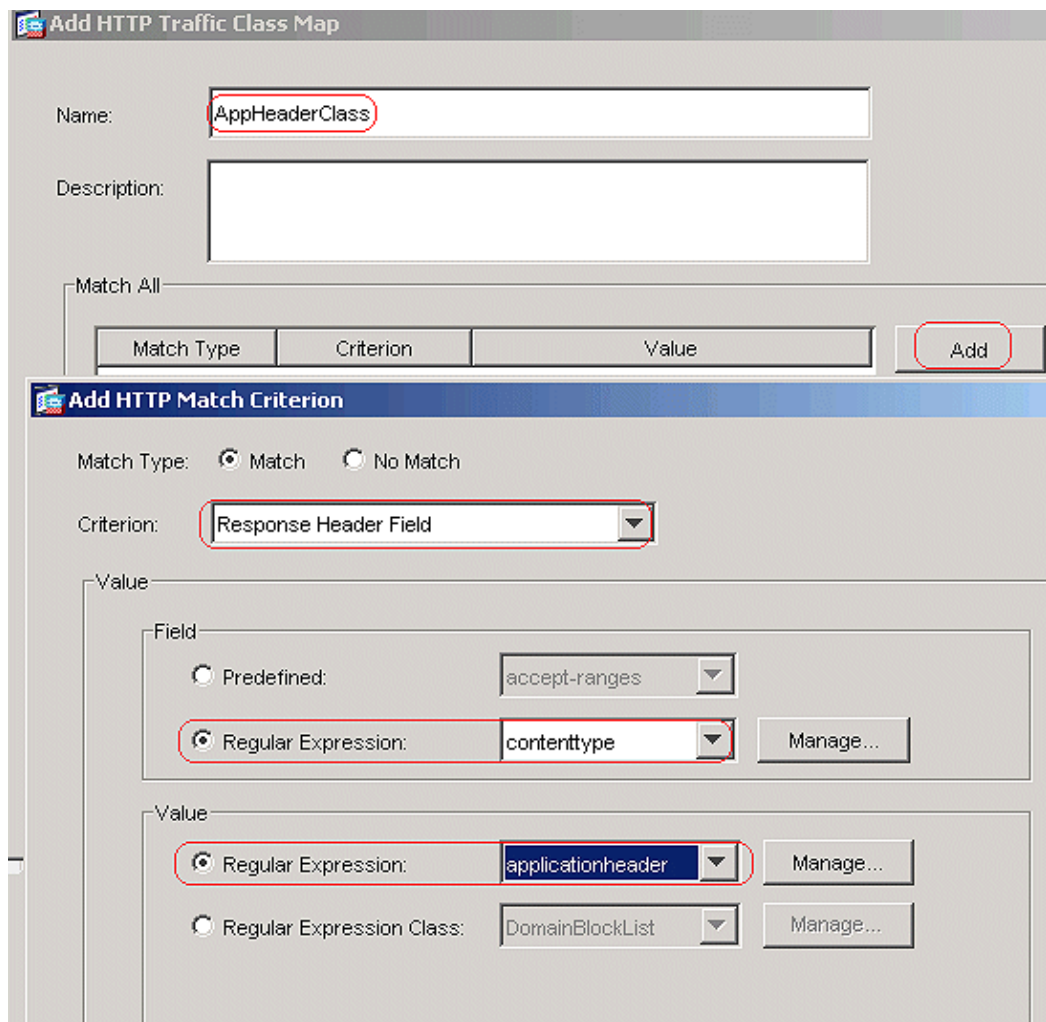
ASA CLI Configuration

```
ciscoasa#configure terminal
ciscoasa(config)#class-map type inspect http match-all BlockDomainsClass
ciscoasa(config-cmap)#match request header host regex class DomainBlockList
ciscoasa(config-cmap)#exit
ciscoasa(config)#class-map type regex match-any URLBlockList
ciscoasa(config-cmap)#match regex urllist1
ciscoasa(config-cmap)#match regex urllist2
ciscoasa(config-cmap)#match regex urllist3
ciscoasa(config-cmap)#match regex urllist4
ciscoasa(config-cmap)#exit
```

3. Inspect the identified traffic with Class maps

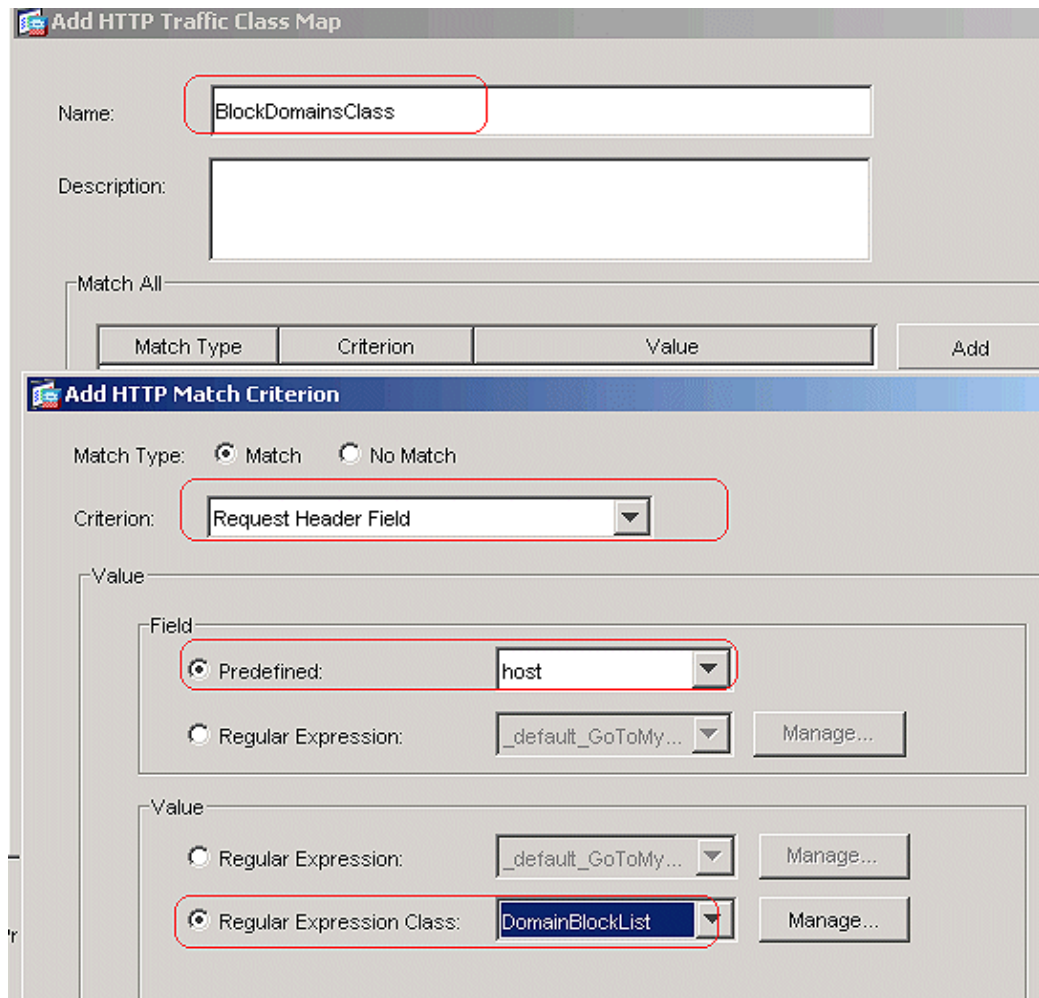
Choose **Configuration > Global Objects > Class Maps > HTTP > Add** in order to create a class map to inspect the HTTP traffic identified by various regular expressions.

- a. Create a class map **AppHeaderClass** in order to match the response header with regular expression captures.



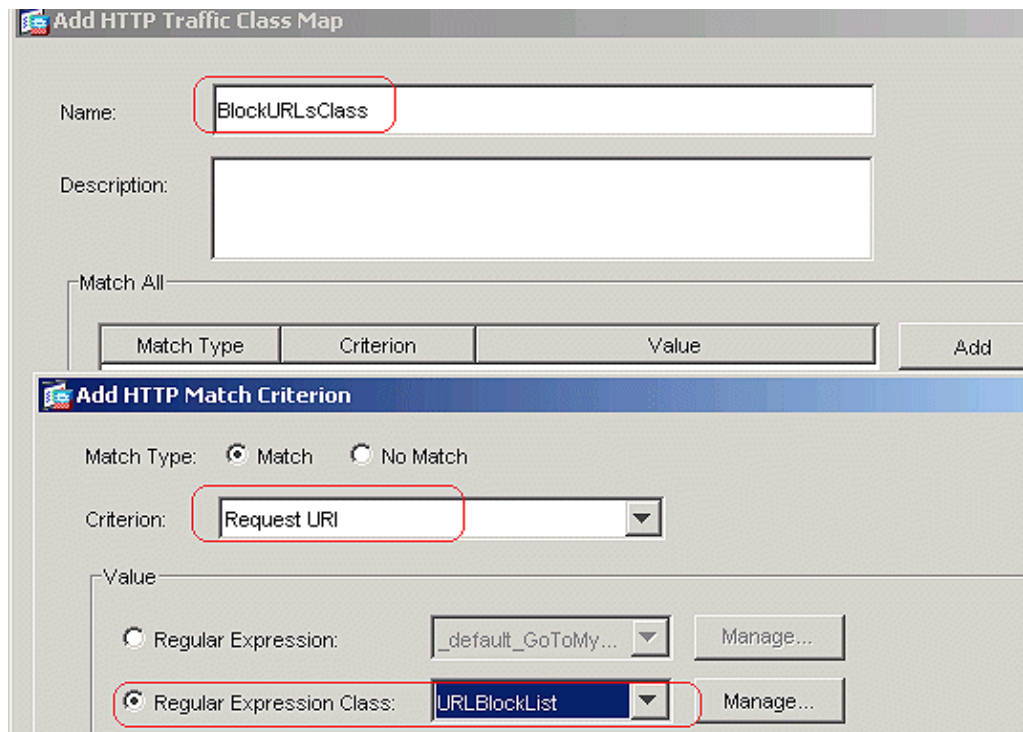
Click **OK**.

- b. Create a class map **BlockDomainsClass** in order to match the request header with regular expression captures.



Click **OK**.

- c. Create a class map **BlockURLsClass** in order to match the request URI with regular expression captures.



Click **OK**.

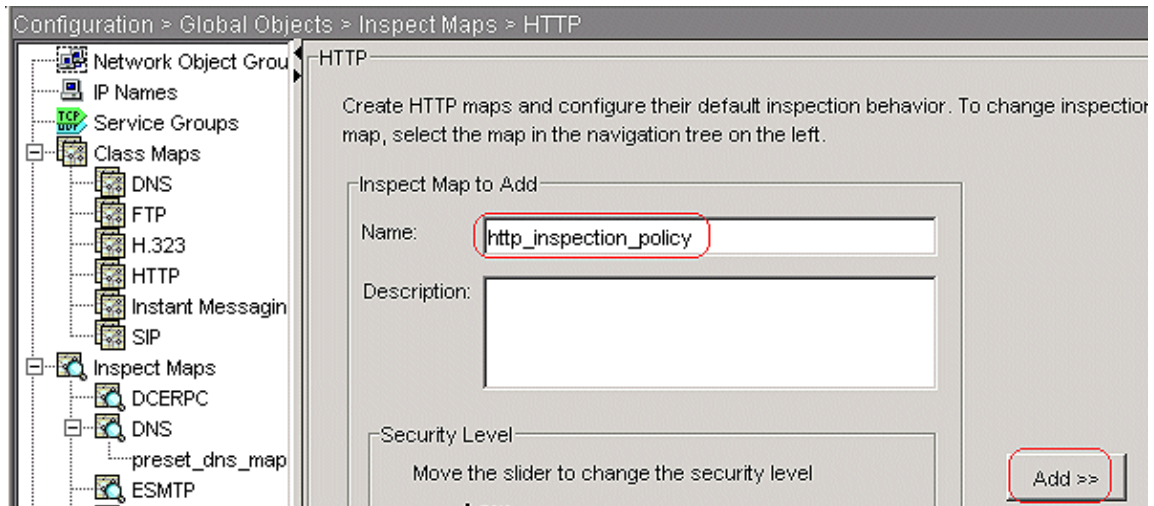
Equivalent CLI Configuration

```
ASA CLI Configuration

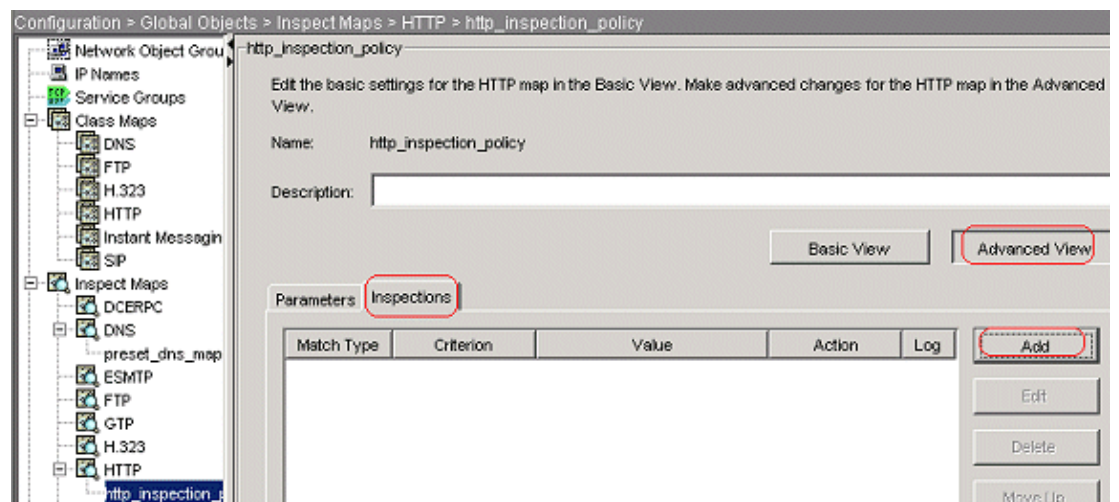
ciscoasa#configure terminal
ciscoasa(config)#class-map type inspect http match-all AppHeaderClass
ciscoasa(config-cmap)#match response header regex contenttype regex
applicationheader
ciscoasa(config-cmap)#exit
ciscoasa(config)#class-map type inspect http match-all BlockDomainsClass
ciscoasa(config-cmap)#match request header host regex class DomainBlockList
ciscoasa(config-cmap)#exit
ciscoasa(config)#class-map type inspect http match-all BlockURLsClass
ciscoasa(config-cmap)#match request uri regex class URLBlockList
ciscoasa(config-cmap)#exit
```

4. Set the actions for the matched traffic in the inspection policy

Choose **Configuration > Global Objects > Inspect Maps > HTTP** in order to create a **http_inspection_policy** to set the action for the matched traffic. Click **Add** and **Apply**.



- a. Choose **Configuration > Global Objects > Inspect Maps > HTTP > http_inspection_policy** and click **Advanced View > Inspections > Add** in order to set the actions for the various Classes created so far.



Click **OK**.

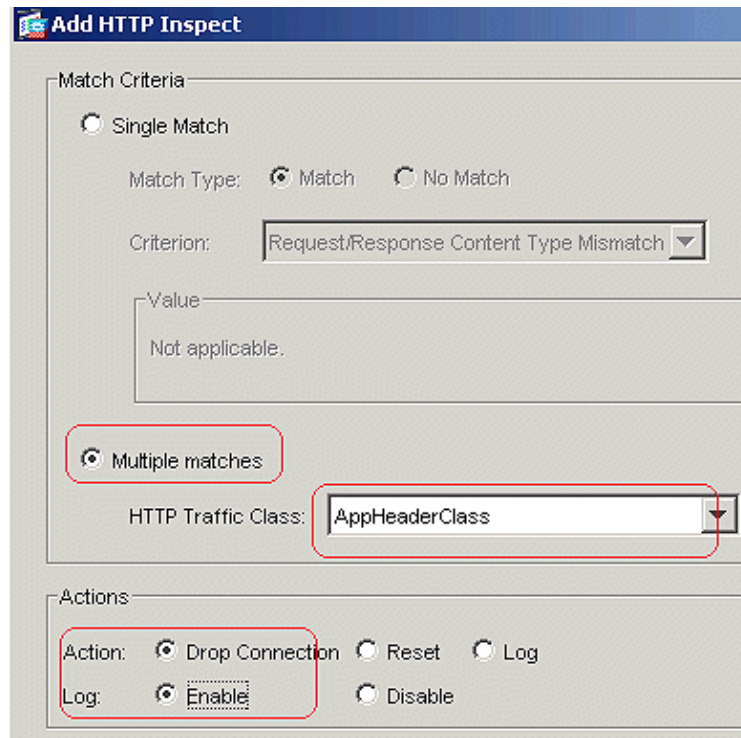
- b. Set the action as **Drop Connection**; **Enable** the logging for the Criterion as Request Method and Value as **connect**.

The screenshot shows the 'Add HTTP Inspect' dialog box with the following settings:

- Match Criteria:**
 - Single Match
 - Match Type: Match No Match
 - Criterion: Request Method
- Value:**
 - Method: connect
 - Regular Expression
 - Regular Expression: _default_GoTo
 - Regular Expression Class: DomainBlockLi
- Multiple matches:** Multiple matches
- HTTP Traffic Class:** AppHeaderClass
- Actions:**
 - Action: Drop Connection Reset Log
 - Log: Enable Disable

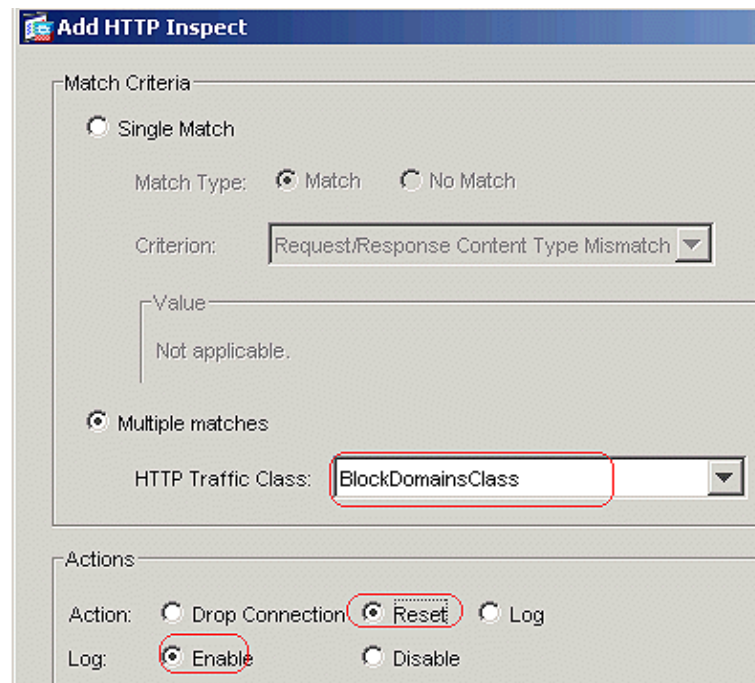
Click **OK**.

- c. Set the action as **Drop Connection**, and **Enable** the logging for the class **AppHeaderClass**.



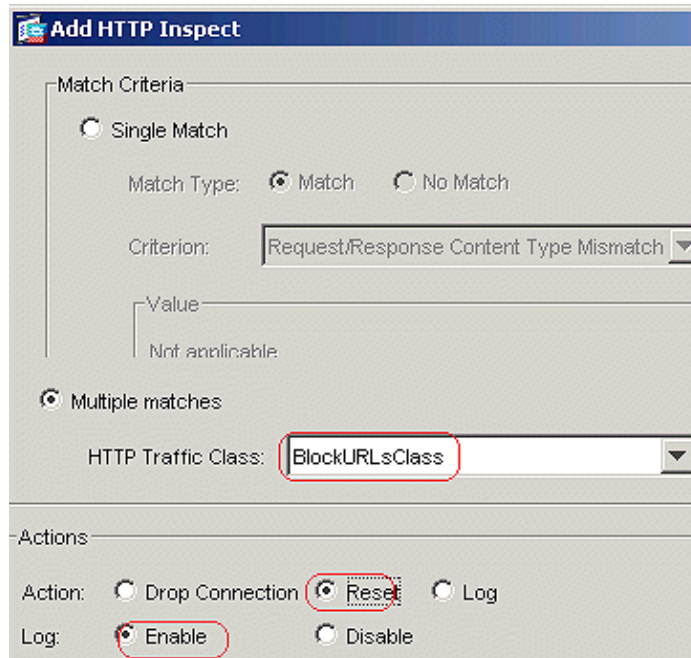
Click **OK**.

- d. Set the action as **Reset**, and **Enable** the logging for the class **BlockDomainsClass**.



Click **OK**.

- e. Set the action as **Reset**, and **Enable** the logging for the class **BlockURLsClass**.



Click **OK**.

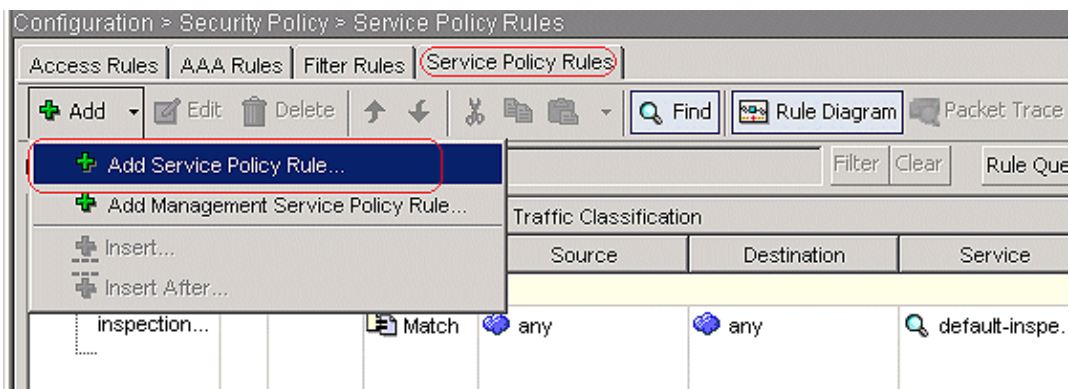
Click **Apply**.

Equivalent CLI Configuration

ASA CLI Configuration
<pre> ciscoasa#configure terminal ciscoasa(config)#policy-map type inspect http http_inspection_policy ciscoasa(config-pmap)#parameters ciscoasa(config-pmap-p)#match request method connect ciscoasa(config-pmap-c)#drop-connection log ciscoasa(config-pmap-c)#class AppHeaderClass ciscoasa(config-pmap-c)#drop-connection log ciscoasa(config-pmap-c)#class BlockDomainsClass ciscoasa(config-pmap-c)#reset log ciscoasa(config-pmap-c)#class BlockURLsClass ciscoasa(config-pmap-c)#reset log ciscoasa(config-pmap-c)#exit ciscoasa(config-pmap)#exit </pre>

5. Apply the inspection http policy to the interface

Choose **Configuration > Security Policy > Service Policy Rules > Add > Add Service Policy Rule** under the Service Policy Rules tab.



HTTP Traffic

- a. Choose the **Interface** radio button with the **inside** interface from the drop-down menu and the Policy Name as **inside-policy**. Click **Next**.

Add Service Policy Rule Wizard - Service Policy

Adding a new service policy rule requires three steps:

- Step 1: Configure a service policy.
- Step 2: Configure the traffic classification criteria for the service policy rule.
- Step 3: Configure actions on the traffic classified by the service policy rule.

Create a service policy and apply to:

Only one service policy can be configured per interface or at global level. If a service policy already exists, then you can add a new rule into the existing service policy. Otherwise, you can create a new service policy.

Interface: inside - (create new service policy)

Policy Name:

Description:

Global - applies to all interfaces

Policy Name:

Description:

- b. Create a class map **httptraffic**, and check the **Source and Destination IP Address (uses ACL)**. Click **Next**.

Add Service Policy Rule Wizard - Traffic Classification Criteria

Create a new traffic class:

Description (optional):

Traffic match criteria

- Default Inspection Traffic
- Source and Destination IP Address (uses ACL)
- Tunnel Group
- TCP or UDP Destination Port
- RTP Range
- IP DiffServ CodePoints (DSCP)
- IP Precedence
- Any traffic

If traffic does not match a existing traffic class, then it will match the class-default traffic class. Class-default can be used in catch all situation.

Use class-default as the traffic class.

< Back **Next >** Cancel

c. Choose the Source and Destination as **any** with the TCP port as **HTTP**. Click **Next**.

Add Service Policy Rule Wizard - Traffic Match - Source and Destination Address

Action:

Source: Type:

Destination: Type:

Protocol and Service

Protocol:

Source Port: Service:

Destination Port: Service:

Options

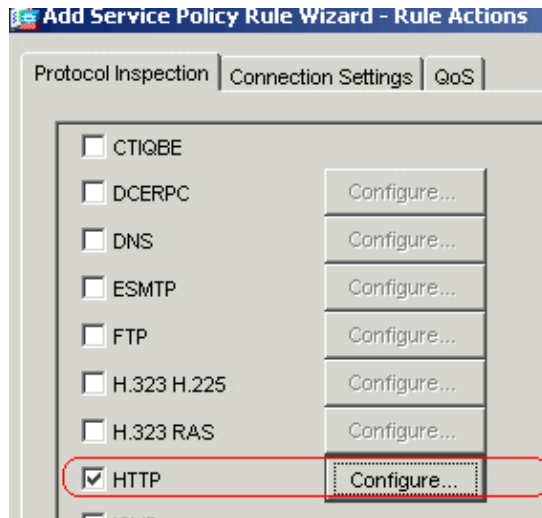
Time Range:

Description:

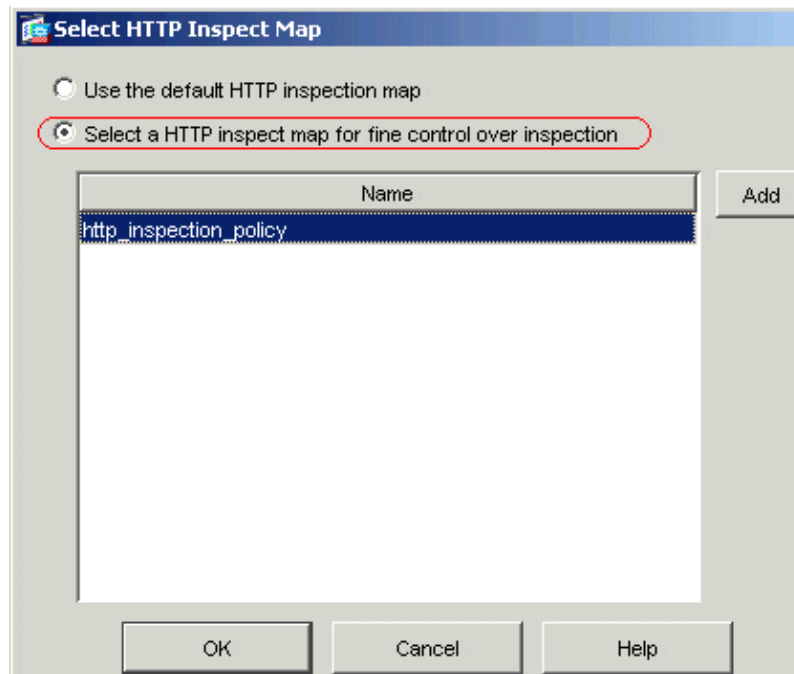
< Back **Next >** Cancel

d. Check the **HTTP** radio button, and click **Configure**.

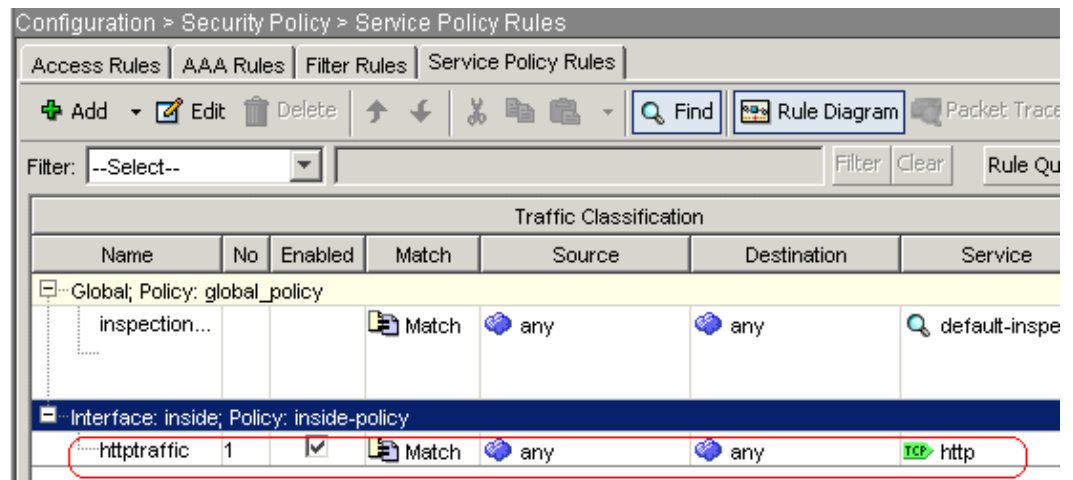
a.



e. Check the radio button **Select a HTTP inspect map for the control over inspection**. Click **OK**.

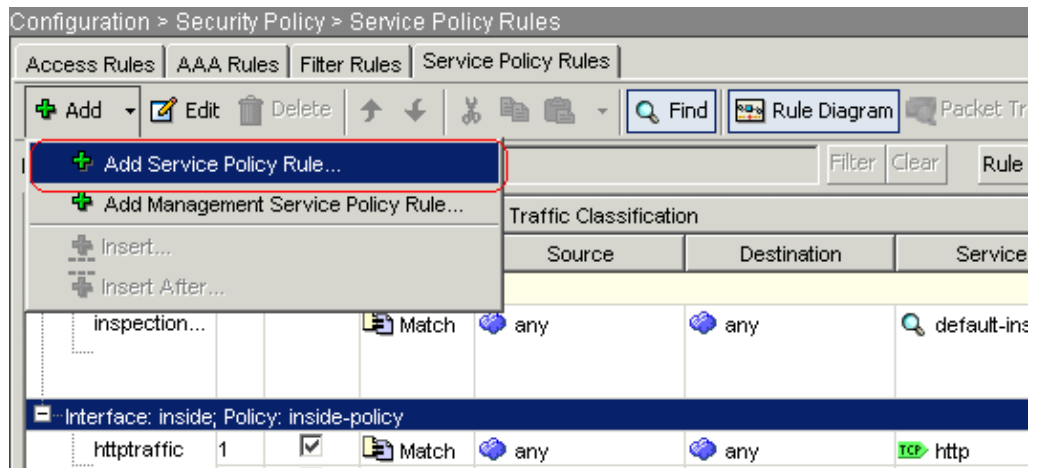


f. Click **Finish**.

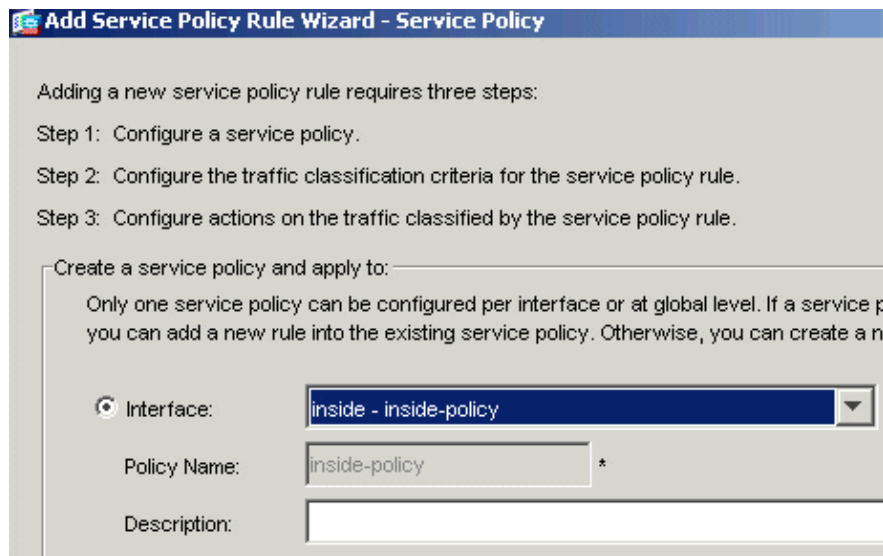


b. **Port 8080 Traffic**

a. Again, click **Add > Add Service Policy Rule**.



b. Click **Next**.



c. Choose the **Add rule to existing traffic class** radio button, and choose **httptraffic** from the drop-down menu. Click **Next**.

Add Service Policy Rule Wizard - Traffic Classification Criteria

Create a new traffic class:

Description (optional):

Traffic match criteria

- Default Inspection Traffic
- Source and Destination IP Address (uses ACL)
- Tunnel Group
- TCP or UDP Destination Port
- RTP Range
- IP DiffServ CodePoints (DSCP)
- IP Precedence
- Any traffic

Rule can be added to existing class map if that class map uses access control list (ACL) as traffic match criteria. Following class maps use ACL as traffic match criteria

Add rule to existing traffic class:

If traffic does not match a existing traffic class, then it will match the class-default traffic class. Class-default can be used in catch all situation.

Use class-default as the traffic class.

d. Choose the Source and Destination as **any** with the TCP port as **8080**. Click **Next**.

Add Service Policy Rule Wizard - Traffic Match - Source and Destination Address

Action:

Source

Type:

Destination

Type:

Protocol and Service

Protocol:

Source Port

Service: =

Group:

Destination Port

Service: =

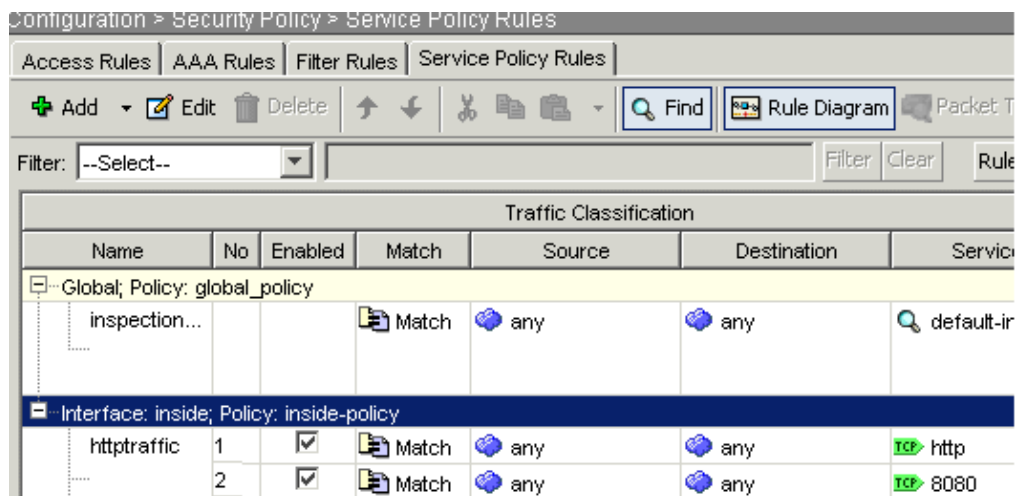
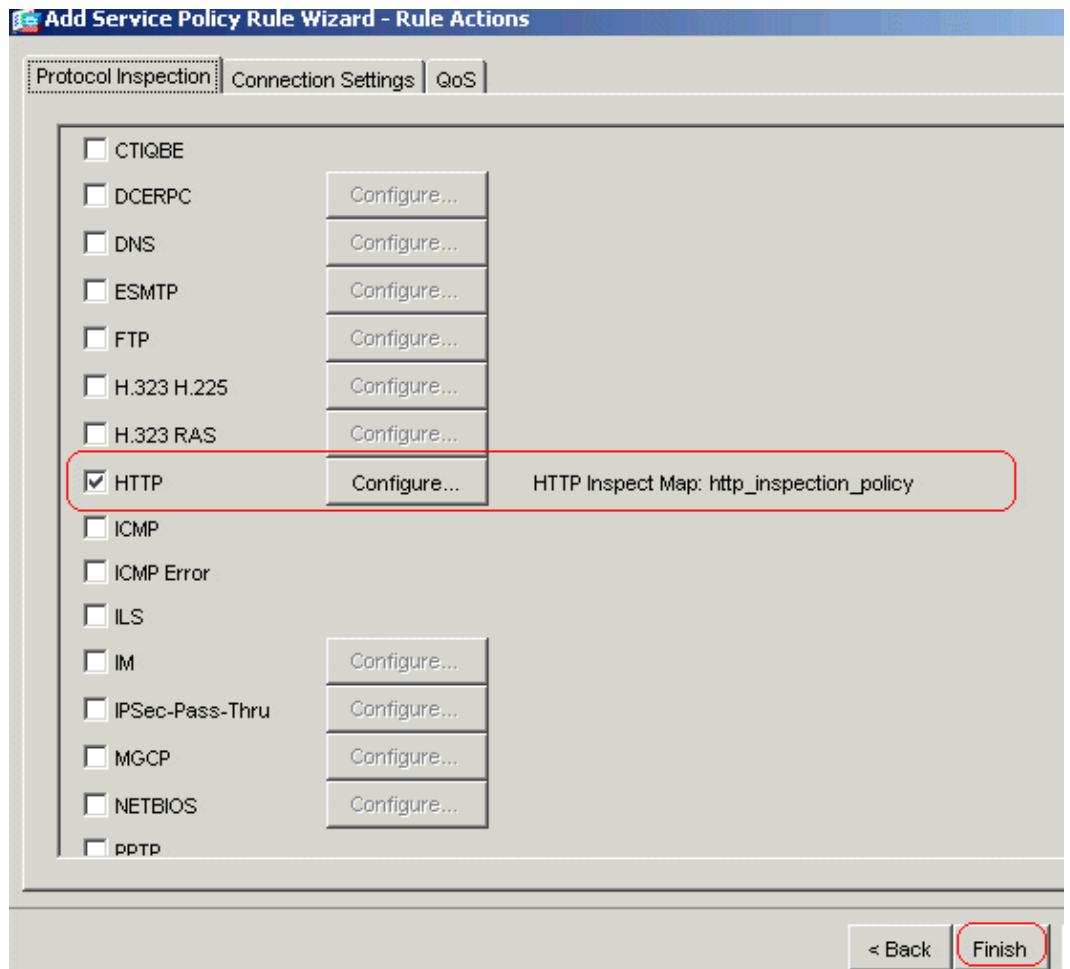
Group:

Options

Time Range:

Description:

e. Click **Finish**.



Click **Apply**.

Equivalent CLI Configuration

ASA CLI Configuration
<pre> ciscoasa#configure terminal ciscoasa(config)#access-list inside_mpc extended permit tcp any any eq www ciscoasa(config)#access-list inside_mpc extended permit tcp any any eq 8080 ciscoasa(config)#class-map httptraffic </pre>

```
ciscoasa(config-cmap)#match access-list inside_mpc
ciscoasa(config-cmap)#exit
ciscoasa(config)#policy-map inside-policy
ciscoasa(config-pmap)#class httptraffic
ciscoasa(config-pmap-c)#inspect http http_inspection_policy
ciscoasa(config-pmap-c)#exit
ciscoasa(config-pmap)#exit
ciscoasa(config)#service-policy inside-policy interface inside
```

Verify

Use this section to confirm that your configuration works properly.

The Output Interpreter Tool (registered customers only) (OIT) supports certain **show** commands. Use the OIT to view an analysis of **show** command output.

- **show running-config regex** Shows the regular expressions that have been configured

```
ciscoasa#show running-config regex
regex urllist1 ".*\.([Ee][Xx][Ee]|[Cc][Oo][Mm]|[Bb][Aa][Tt]) HTTP/1.[01]"
regex urllist2 ".*\.([Pp][Ii][Ff]|[Vv][Bb][Ss]|[Ww][Ss][Hh]) HTTP/1.[01]"
regex urllist3 ".*\.([Dd][Oo][Cc]|[Xx][Ll][Ss]|[Pp][Pp][Tt]) HTTP/1.[01]"
regex urllist4 ".*\.([Zz][Ii][Pp]|[Tt][Aa][Rr]|[Tt][Gg][Zz]) HTTP/1.[01]"
regex domainlist1 "\.yahoo\.com"
regex domainlist2 "\.myspace\.com"
regex domainlist3 "\.youtube\.com"
regex contenttype "Content-Type"
regex applicationheader "application/*"
ciscoasa#
```

- **show running-config class-map** Shows the class maps that have been configured

```
ciscoasa#show running-config class-map
!
class-map type regex match-any DomainBlockList
  match regex domainlist1
  match regex domainlist2
  match regex domainlist3
class-map type inspect http match-all BlockDomainsClass
  match request header host regex class DomainBlockList
class-map type regex match-any URLBlockList
  match regex urllist1
  match regex urllist2
  match regex urllist3
  match regex urllist4
class-map inspection_default
  match default-inspection-traffic
class-map type inspect http match-all AppHeaderClass
  match response header regex contenttype regex applicationheader
class-map httptraffic
  match access-list inside_mpc
class-map type inspect http match-all BlockURLsClass
  match request uri regex class URLBlockList
!
ciscoasa#
```

- **show running-config policy-map type inspect http** Shows the policy maps that inspects the http traffic that have been configured

```
ciscoasa#show running-config policy-map type inspect http
!
policy-map type inspect http http_inspection_policy
  parameters
    protocol-violation action drop-connection
```

```

class AppHeaderClass
  drop-connection log
match request method connect
  drop-connection log
class BlockDomainsClass
  reset log
class BlockURLsClass
  reset log
!
ciscoasa#

```

- **show running-config policy-map** Displays all the policy-map configurations as well as the default policy-map configuration

```

ciscoasa#show running-config policy-map
!
policy-map type inspect dns preset_dns_map
  parameters
    message-length maximum 512
policy-map type inspect http http_inspection_policy
  parameters
    protocol-violation action drop-connection
  class AppHeaderClass
    drop-connection log
  match request method connect
    drop-connection log
  class BlockDomainsClass
    reset log
  class BlockURLsClass
    reset log
policy-map global_policy
  class inspection_default
    inspect dns preset_dns_map
    inspect ftp
    inspect h323 h225
    inspect h323 ras
    inspect netbios
    inspect rsh
    inspect rtsp
    inspect skinny
    inspect esmtp
    inspect sqlnet
    inspect sunrpc
    inspect tftp
    inspect sip
    inspect xdmcp
policy-map inside-policy
  class httptraffic
    inspect http http_inspection_policy
!
ciscoasa#

```

- **show running-config service-policy** Displays all currently running service policy configurations

```

ciscoasa#show running-config service-policy
service-policy global_policy global
service-policy inside-policy interface inside

```

- **show running-config access-list** Displays the access-list configuration that runs on the security appliance

```

ciscoasa#show running-config access-list
access-list inside_mpc extended permit tcp any any eq www

access-list inside_mpc extended permit tcp any any eq 8080
ciscoasa#

```

Troubleshoot

This section provides information you can use to troubleshoot your configuration.

Note: Refer to Important Information on Debug Commands before you use **debug** commands.

- **debug http** Shows the debug messages for HTTP traffic.

Related Information

- [Cisco Adaptive Security Appliance Support Page](#)
 - [Cisco Adaptive Security Device Manager \(ASDM\) Support Page](#)
 - [Cisco 500 Series PIX Support Page](#)
 - [Technical Support & Documentation – Cisco Systems](#)
-

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

© 2010 – 2011 Cisco Systems, Inc. All rights reserved. [Terms & Conditions](#) | [Privacy Statement](#) | [Cookie Policy](#) | [Trademarks of Cisco Systems, Inc.](#)

Updated: Sep 30, 2008

Document ID: 100513
