



TrafficShield[®] Security Policy User Manual

version 3.2

Service and Support Information

Product Version

This manual applies to product version 3.2 of the TrafficShield® Application Firewall.

Legal Notices

Copyright

Copyright 2002 - 2005, F5 Networks, Inc. All rights reserved.

F5 Networks, Inc. (F5) believes the information it furnishes to be accurate and reliable. However, F5 assumes no responsibility for the use of this information, nor any infringement of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent, copyright, or other intellectual property right of F5 except as specifically described by applicable Control user licenses. F5 reserves the right to change specifications at any time without notice.

Trademarks

F5, F5 Networks, the F5 logo, BIG-IP, 3-DNS, iControl, GLOBAL-SITE, SEE-IT, EDGE-FX, FireGuard, Internet Control Architecture, IP Application Switch, iRules, OneConnect, Packet Velocity, SYN Check, Control Your World, ZoneRunner, uRoam, FirePass, and TrafficShield are registered trademarks or trademarks of F5 Networks, Inc. in the U.S. and certain other countries. All other trademarks mentioned in this document are the property of their respective owners. F5 Networks' trademarks may not be used in connection with any product or service except as permitted in writing by F5.

Export Regulation Notice

This product may include cryptographic software. Under the Export Administration Act, the United States government may consider it a criminal offense to export this product from the United States.

Export Warning

This is a Class A product. In a domestic environment this product may cause radio interference in which case the user may be required to take adequate measures.

FCC Compliance

This equipment generates, uses, and may emit radio frequency energy. The equipment has been type tested and found to comply with the limits for a Class A digital device pursuant to Part 15 of FCC rules, which are designed to provide reasonable protection against such radio frequency interference.

Operation of this equipment in a residential area may cause interference, in which case the user at his own expense will be required to take whatever measures may be required to correct the interference.

Any modifications to this device, unless expressly approved by the manufacturer, can void the user's authority to operate this equipment under part 15 of the FCC rules.

Canadian Regulatory Compliance

This class A digital apparatus complies with Canadian I CES-003.

Standards Compliance

The product conforms to ANSI/UL 60950-1-2002 1st edition and Certified to CAN/CSA C22.2 No. 60950-1-3 first edition.



Table of Contents

I		
Introduction		
	Product overview	1-1
	Document objectives	1-1
	How this manual is organized	1-1
	Audience and assumed knowledge	1-2
	Conventions	1-2
	Related documentation	1-2
2		
The Security Policy		
	Concept	2-1
	How the policy works	2-1
	The security policy components	2-3
	Object types	2-3
	Web objects	2-3
	Application flows	2-3
	Flow parameters	2-4
	What happens to illegal requests	2-5
	Defaults tab	2-6
	Negative Regular Expressions Policy Defaults	2-6
	Creating a Pool of Regular Expressions	2-6
	Assigning Expressions	2-8
3		
TrafficShield Workflow		
	Guidelines to workflow	3-1
	Preliminary stage	3-2
	Stage 1 - Defining the web application	3-2
	Stage 2 - Creating a policy	3-2
	Stage 3 - Testing and fine-tuning the policy	3-3
	Stage 4 - Putting the policy into effect: blocking	3-3
4		
Policy Management Configuration		
	Scope	4-1
	Add a new policy	4-2
	Policy properties	4-4
	Editing the current policy's properties	4-4
	Policy-specific negative regular expressions	4-9
	Setting the active policy of a web application	4-10
	Switching between policies	4-10
	Automatic update of policy versions	4-11
	Blocking Policy table	4-12
	RFC violations	4-13
	Access Violations	4-13
	Length violations	4-14
	Input violations	4-15
	Cookie Violations	4-16
	Negative security violations	4-16
	Other policy activities	4-19
	Edit a policy	4-19
	Remove a policy	4-20

Copy a policy	4-22
Roll back to a previous policy version	4-23
Flow properties	4-24
Policy component editing	4-25
Adding Object types	4-25
Allowed objects RegExp - Object list relaxation	4-28
Web objects	4-30
Object properties	4-31
Flows to object	4-33
Adding a Web object	4-35
Removing a Web object	4-35
Displaying web application flow model	4-35
Application flow	4-36
Defining the Flow parameters	4-40
Defining negative regular expressions	4-46
Character sets	4-47
Policy audit tools	4-50

5

Crawler

Crawler overview	5-1
Populating the policy using the Crawler	5-1
Configuring and launching the Crawler	5-2
Starting the Crawler using the Wizard	5-2
Crawler scheduling - Step 1 in Crawler Wizard	5-2
Start points - Step 2 in Crawler Wizard	5-3
Form Filler - Step 3 in Crawler Wizard	5-4
Page not found criteria - Step 4 in Crawler Wizard	5-6
Logout pages - Step 5 in Crawler Wizard	5-7
Properties - Step 6 in Crawler Wizard	5-7
HTTP authentication - Step 7 in Crawler Wizard	5-10
File type associations - Step 8 in Crawler Wizard	5-11
Crawler configuration settings - Step 9 in Crawler Wizard	5-12
Data collection with policy browser	5-13
Running the Crawler Manually	5-13
Crawler Learning tool	5-15

6

Learning - Testing & Fine Tuning the Policy

Overview	6-1
Learning tool	6-1
Learning duration	6-2
Auto Accept build tool	6-2
Accessing the Learning data	6-5
Access violations	6-6
Illegal object type	6-7
Non-existent object	6-10
Illegal flow to object	6-12
Illegal entry point	6-15
Illegal method	6-17
Length violations	6-18
Object type lengths errors	6-18
Header length errors	6-21

Input violations	6-22
Illegal query-string or POST-data	6-23
Illegal parameter	6-23
Illegal static parameter value	6-24
Illegal empty parameter value	6-25
Illegal parameter value length	6-26
Illegal parameter numeric value	6-28
Illegal parameter data type	6-29
Illegal meta character in parameter value	6-30
Malicious parameter value	6-32
Negative security violations	6-35
Illegal meta character in header	6-35
Illegal meta character in object	6-37
Illegal meta character in parameter name	6-37
Illegal meta character in parameter value	6-38
Illegal pattern in object	6-39
Illegal pattern in response	6-40
Illegal pattern in header	6-40
Illegal pattern in user input	6-41
Cookie violations	6-43
Modified domain cookies	6-43
Objects that modified domain cookies	6-44
Forensics	6-45
Illegal requests	6-45
Ignored requests	6-48
Ignored items	6-49

Glossary

Table of Contents



I

Introduction

- Product overview
- Document objectives
- How this manual is organized
- Audience and assumed knowledge
- Conventions
- Related documentation

Product overview

The F5[®] Networks TrafficShield[®] Application Firewall is targeted at protecting mission-critical Web infrastructure against application layer attacks, and to monitor the protected web applications. These services complement the limited protection provided by firewalls, load balancers, and other types of data and service protection devices. The TrafficShield security application analyzes traffic at network and application levels to handle a variety of threats, such as:

- Manipulation of cookies or hidden fields.
- Insertions of SQL commands or HTTP structures into user input fields in order to expose confidential information or to deface content.
- Malicious exploitations of the application memory buffer to stop services, to get shell access and to propagate worms.
- Unauthorized changes to server content via HTTP Delete and Put commands.
- Attempts aimed at causing the Web application to be unavailable or to respond slowly to legitimate users.
- Forceful browsing.

Document objectives

This manual explains how to set up a TrafficShield security policy and how to apply it to a Web application. The manual presents TrafficShield security application's security concepts, and shows how the concepts are implemented in the security policy context.

How this manual is organized

This manual consists of the following chapters:

Chapter 1 - Introduction: This chapter provides an overview of the F5[®] Networks TrafficShield[®] Application Firewall product, describes the manual chapter organization and provides information about the color conventions used in the TrafficShield security application, and about related documentation.

Chapter 2 - The Security Policy: This chapter explains how a TrafficShield security policy works, describes its components, and presents the Policy Browser, Crawler, and Learning tools that will help you to automatically collect the components.

Chapter 3 - TrafficShield Workflow: This chapter is your guide to the TrafficShield security policy workflow; it describes the steps to follow in order to create, adjust, and maintain a security policy. Subsequent chapters explain each step in detail.

Chapter 4 - Policy Management Configuration: This chapter explains how to create and maintain policies, and describes the different components of the policies.

Chapter 5 - Crawler: This chapter guides you step-by-step through the procedure required to create an initial policy using the Crawler tool. This chapter also provides instructions on how to use the more advanced Crawler parameters.

Chapter 6 - Learning: Testing and Fine-Tuning the Policy: This chapter explains how to use the Learning tool to adapt the policy to real-life traffic requirements. It also covers the Policy editing feature, which allows you to view and manually adjust the entire security policy.

Glossary- The Glossary lists and defines relevant terms.

Audience and assumed knowledge

This manual is intended for the Web application security administrator or application owner. It assumes acquaintance with the nature of Web application attacks and a working knowledge of the Internet and of HTTP requests.

Conventions

Gold-colored lettered URLs point to referrer objects (see *Referrer* in the *Glossary* for definition). Green URLs belong to non-referrer objects.

Frame Target: 1	
GET(0)	» [HTTP] /bidhistory.php
GET(2)	» [HTTP] /browse.php
POST(7)	» [HTTP] /buy2.php
GET(0)	» [HTTP] /email_request.php
GET(0)	» [HTTP] /help.php
GET(0)	» [HTTP] /images/linea.gif
GET(0)	» [HTTP] /images/logo.gif
GET(0)	» [HTTP] /index.php
GET(2)	» [HTTP] /item.php
GET(2)	» [HTTP] /search.php
GET(0)	» [HTTP] /sell.php
GET(0)	» [HTTP] /user_login.php

Related documentation

The *TrafficShield® Installation and Configuration Manual version 3.2* explains how to configure the deployed TrafficShield unit and its backup.



2

The Security Policy

- Concept
- How the policy works
- The security policy components
- Defaults tab

Concept

The F5 TrafficShield Application Firewall uses positive security logic, as well as negative security logic according to the user's selection.

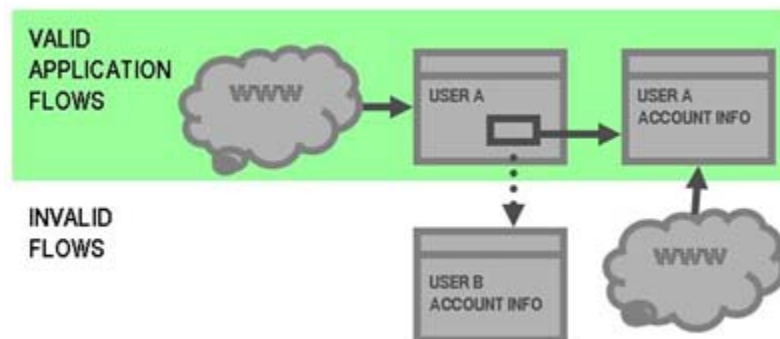
This means that all traffic is considered illegal unless it is specifically known to be legal.

The security policy is therefore a map of the application itself, containing all the application objects, flows, parameter values, and attributes.

The core of TrafficShield system's security functionality is the security policy. This policy determines which requests are valid and therefore can deny any request which does not match the policy's definitions. Depending on the work mode established, an invalid request can be blocked and reported, or only reported. When the TrafficShield system policy is initialized for the first time, all the objects, object types and web server application flows are not yet recognized and accordingly, TrafficShield security application will analyze all incoming requests.

How the policy works

We call this map the "Application Flow Model." Think of it as a model of the entire application: every object, every parameter, and every value range for each parameter is part of the flow. By checking incoming traffic against the Application Flow Model, TrafficShield security application can screen out requests that do not follow the user behavior the application expects.



From every object in an application, a user may request access to a limited number of destinations. For example, when users log in to an online banking application, they are provided with several links to their respective accounts: savings, checking, and so on. They can click on each link to be directed to their personal account information and view it securely. This is the legitimate flow of the application, and this is the series of requests which are captured in the Application Flow Model.

Requests that are out of sequence or whose parameter values have been altered can be blocked once this security policy is in place. For instance, a user requesting an account information page, without first passing through

login sequence, can be rejected, as this is not the correct order of the flow. Likewise, a user who logs in and then tampers with the account links provided on a page (attempting to access other people's accounts) would be rejected since the parameter values have changed.

◆ **Note**

In each of these cases, the format and structure of the request are valid, according to the HTTP protocol. It is only within the specific context of the application that these requests can be considered malicious.

The security policy components

The main components of the TrafficShield positive security policy are described in this section.

Object types

The Object Types section lists the existing file types in the protected Web site. For example, a list of valid object types for a specific policy could be: **GIF JPG** and **HTML** only. If your policy contains the above list, then any request for a PDF file would be considered illegal.

◆ **Note**

The object types are case sensitive. As a result, JPEG and jpeg are considered as two object types.

Web objects

The Objects (files) section lists the existing objects in the protected Web site. For example, a list of valid objects could be: myPict.gif, myPict.jpg and myFile.html only. If your policy contains this list, then any request for yourFile.html would be considered illegal.

◆ **Note**

Object name files that are not included in the list are also case sensitive.

Application flows

The Application Flow (path) is the defined access path leading from one object to another object. For example, a list of valid flows would be:

- from abc.html to abc.gif, OK
- from abc.html to def.html, OK

If your policy contains this list, then any request that tries to access abc.gif from def.html would be considered illegal.

◆ **Tip**

Back flows are created automatically.

Flow parameters

The parameters used by the request. For example:

A list of valid parameters can be:

https://192.168.51.51:1043/dms/policy/pl_flows.php?m_id=0_4&uid=123

In this example we have a single parameter: `m_id`.

If your policy contains the above list, then any request that tries to read a variable with a different name from `m_id` would be considered illegal.

Please refer to the next section for more details.

Parameter value properties

The TrafficShield security application provides an option whereby you can define the allowed value format for each parameter of the request. For example, a list of valid parameters can be:

<https://192.168.51.51:1043/dms/policy/login?username=john&password=secret>

If your policy contains the above list, then calling this request with a value other than **john** would be considered illegal.

Character sets

For each policy, the user can define the allowed character sets for the following request parts: Object, Parameter Name, Parameter Value and HTTP header.

If your policy contains a specific allowed character set that excludes the letter "Z" in the HTTP header part, then any request containing the letter "Z" in its header will be considered illegal.

Negative Regular Expressions

Negative regular expressions describe possible attacks.

For example: a regular expression that defines a cross-site scripting attack:

```
(?si)%3cscript\b
```

TrafficShield security application enables the user to define a list of regular expressions that will be used to check each of the parts in the request: The requested object, the request parameters, name and value pair, and the request HTTP header.

In addition, the user can define a list of regular expressions that will be used to check the response.

If your policy contains the above negative regular expression, then any request for a URL matching this list of directories will be considered illegal.

The policy build tools

The policy is an intelligent map of your Web application. It contains not only a list of the files included in the Web application but also other data such as the types of the files, the length of some crucial strings, allowed value ranges for parameters, and the relationships (links) between the files and the parameters passed from one file to another in a specific link.

You do not build this complex map yourself, which would be a tedious undertaking, especially if the Web application is updated frequently. TrafficShield security application provides the following tools for building this map:

- The Policy Browser collects important information about the site that the Crawler later uses while scanning the application. The user simply browses the application with it. The browser saves to a file the browsing information it encounters.
- The Crawler scans your application and builds a list of existing object types, objects, flows, parameters, and parameter value attributes, including objects generated by client-side Java Script code. It can also use as input the file created by the Policy Browser.
- The Learning mechanism can analyze traffic from sources such as real live traffic, and the Crawler.
- The Policy Management graphical user interface allows you to see the entire policy built for the Web application. It is a visual representation of the application flow model, which can be easily edited. Although a policy could, in theory, be built using just the Crawler and Learning, viewing and editing the policy is an effective way to ensure its accuracy.

The Crawler, Learning mechanism, and Policy Editing capabilities complement each other. The Crawler is generally used to generate the initial policy. Subsequently, the Learning tool shows you whether the Crawler's decisions are consistent with the requirements of real-life traffic, and allows you to further tune your policy until it is ready. For more details, please refer to 5, *Crawler* in this document.

What happens to illegal requests

When the TrafficShield security application diagnoses a request as illegal, it processes it according to the policy settings: it either warns you and lets the request through, or warns you and blocks the request.

Defaults tab

The following sections describe the Defaults tab options under Administration.

Negative Regular Expressions Policy Defaults

TrafficShield security application policies use expressions to check the existence of certain text strings in incoming requests as a way of identifying attacks. For example, you can use a set of regular expressions to verify that the request URI does include malicious content.

◆ **Note**

Negative regular expressions means that requests whose content matches one of the regular expressions are considered malicious and therefore will generate a security alert and then be blocked by the TrafficShield security application according to that policy.

The use of negative regular expressions involves the following tasks:

- Create a pool of regular expressions.
- Apply the regular expression to the request component it is designed to check (e.g., URI, header).
- Use the regular expression in the policy.

The regular expressions become active only after you assign them to policies. The sections that follow explain how to build the pool of expressions and how to associate them with request elements they are designed to check. For details on how to actually assign the regular expressions, see *Assigning Expressions*, on page 2-8.

Creating a Pool of Regular Expressions

When you create an expression, it goes to a pool of expressions. Subsequently, you can select expressions from the pool and assign them to various application elements.

To create a regular expression

1. Click the **Administration** button.
2. On the navigation panel, under **Configuration**, select **Defaults**. The Regular Expressions page opens, listing any expressions you may have defined previously.

Configuration » Defaults Current User: root Version: 3.2.0

RegExp Pool Add Edit Remove

<input type="checkbox"/>	Used	RegExp Name	RegExp	Description
<input type="checkbox"/>	✓	Directory Traversal 1	(?:\.[?:(?:25)?(?:?:%3)?2](?:e % [46]5))(?:\.[?:(?:25)? /](?:%:(?:25)?(?:?:%3)?2)(?:f % [46]6)) (?:\.[?:(?:25)? (?:25)?(?:?:%3)?5)(?:c % [46]3)))	Directory Traversal 1
<input type="checkbox"/>	✓	Directory Traversal 2	(?:\.[?:(?:25)?(?:?:%3)?2](?:e % [46]5))(?:\.[?:(?:25)? ?:%:(?:25)?(?:?:%3)?2)(?:e % [46]5))	Directory Traversal 2
<input type="checkbox"/>	✓	Directory Traversal 3	(?:\.[?:(?:25)?(?:?:%3)?2](?:f % [46]6))(?:\.[?:(?:25)? ?:%:(?:25)?(?:?:%3)?2)(?:e % [46]5))	Directory Traversal 3
<input type="checkbox"/>	✓	Directory Traversal 4	(?:\.[?:(?:25)?(?:?:%3)?5)(?:c % [46]3))(?:\.[?:(?:25)? ?:%:(?:25)?(?:?:%3)?2)(?:e % [46]5))	Directory Traversal 4
<input type="checkbox"/>		Directory Traversal -- Double Uplink 1	(?:\.[?:(?:25)?(?:?:%3)?2](?:e % [46]5))(?:\.[?:(?:25)? ?:%:(?:25)?(?:?:%3)?2)(?:e % [46]5)) (?:\.[?:(?:25)? (?:25)?(?:?:%3)?2)(?:f % [46]6))+ (?:\.[?:(?:25)? ?:%:(?:25)?(?:?:%3)?5)(?:c % [46]3))+)(?:\.[?:(?:25)? (?:25)?(?:?:%3)?2)(?:e % [46]5)) (?:\.[?:(?:25)? (?:25)?(?:?:%3)?2)(?:e % [46]5))	Directory Traversal -- Double Uplink 1
			(?:\.[?:(?:25)?(?:?:%3)?2](?:e % [46]5))	

- In **RegExp Pool** page, click **Add**.
The Add RegExp page opens.

Configuration » Defaults Current User: root Version: 3.2.0

Add RegExp Save Cancel

RegExp Name: *

RegExp: *

Description:

- In **RegExp Name**, enter a name that will help you identify the regular expressions when creating policies.
- In **RegExp**, type the expression by following the standard Regular Expression syntax.
- In **Description**, optionally type a few words that describe the expression.
- Click **Save**.
The regular expression definition appears on the main page.
- Repeat the above procedure for all the expressions you intend to use.

Applying changes in RegExp Pool requires updating each policy containing modified regular expressions. This can be done by clicking the **Update TrafficShield** button in Policy Management.

Assigning Expressions

Regular expressions residing in the pool can be used to check various components such as object path, headers, parameter key and value pairs, as well as the web server response. The next task is to determine what each of the expressions included in the pool is for.

To assign an expression to an application element

1. Click the **Administration** button.
2. On the navigation panel, under **Configuration**, select **Defaults**.
3. Scroll down to **Negative RegExp Policy Defaults**, and then click **Add**.

The Add Negative RegExp page opens.

4. In **RegExp Name**, select the name of the regular expression you want to assign to an application element.
The drop-down list displays the regular expressions currently included in the pool.
5. In **Apply To**, select where to apply the expression.

The options are as follows:

Option	Applies the regular expression to
Object	The Object path of the request.
Response	The response returned from the Web server.
Header value	The request's HTTP header.
User input	The parameter key and value pairs included in the request. Both in the query string and in the Post data.

6. In **Except RegExp**, you can enter another regular expression that defines an exception to the rule set by the selected expression.
7. Click the **Save** button.
The regular expression definition appears on the main page.
8. Repeat the above procedure for all the expressions you intend to use.



3

TrafficShield Workflow

- Guidelines to workflow
- Preliminary stage
- Stage 1 - Defining the web application
- Stage 2 - Creating a policy
- Stage 3 - Testing and fine-tuning the policy
- Stage 4 - Putting the policy into effect: blocking

Guidelines to workflow

This chapter is your guide to the TrafficShield security application workflow: it describes the steps to follow for creating, adjusting, and maintaining a security policy.

The following table provides a summary of the steps to follow, and the resources needed to implement them.

Stage	Resource Required	Time Required
Preliminary Stage: Installing and Configuring the TrafficShield unit	Administrator	1-2 hours depending on the network configuration
Stage 1: Defining the Web application	Administrator or Web application manager	0.5-1 hour for small to medium Web applications and 3-4 hours for bigger and more complex Web applications.
Stage 2: Creating and modifying the initial policy.	Policy Builder: A person who has knowledge of the Web application.	2 hours to set up. Crawler may take several minutes to several hours to run the automatic process. (Allow 1 hour for all static pages, and several minutes for each dynamic script.)
Stage 3: Testing and fine-tuning the policy	Policy builder	1 hour a day for 1-2 weeks
Stage 4: Putting the policy into effect: Blocking	Policy builder	1-2 hours

Preliminary stage

At this stage, the TrafficShield security application has been installed and configured according to the following order:

- TrafficShield unit was installed and configured.
- TrafficShield security application license was activated.
- Web application was configured.

Stage 1 - Defining the web application

This stage includes creating the Web application and defining the TrafficShield hardware units included in the Web application.

This stage is described in the *Defining a new Web Application*, on page 4-1 in the *TrafficShield® Installation and Configuration Manual version 3.2*. The remaining stages are described in this manual.

After you configure a web application, the TrafficShield system automatically creates a default policy. This default policy serves as the basis on which further settings can be applied to build up the final policy that will secure the web application.

Stage 2 - Creating a policy

After defining the Web application, it is necessary to populate a policy with the specific web application policy components.

This stage includes:

- Defining a new policy
- Running the Crawler

The Crawler automatically creates a preliminary security policy for the application. Typically, the Crawler maps most of the objects, flows, and parameter value ranges in a Web application, including those generated dynamically using Java Script and other client-side scripting means. This initial policy is never fully accurate, however.

For instance, while the Crawler can determine parameter values for static parameters such as drop-down lists, it cannot always provide reasonable value ranges for user-input parameters. You can enter these finishing touches to the policy using the automated Learning mechanism and the Policy Management Configuration tools (stage 3).

Stage 3 - Testing and fine-tuning the policy

After creating the initial policy using the Crawler, you can expose the application to user traffic in a non-blocking, "what if" mode. This can be safe traffic, that is, traffic generated by users who are not potential attackers. This safe traffic is typically a small group of QA persons or the employees of your company. If the application is already active (i.e., a legacy application), you can apply the same procedure (again, in a non-blocking mode) and adjust the policy in order to maximize security and minimize the chance of false positives.

During the testing stage, TrafficShield security application captures the "illegal" requests and displays the appropriate information, such as URI lengths that exceed your expectations or attempts to access non-existing objects. Although you know what the values should be, and you may have entered them during your review, the real-life traffic may return unforeseen but legal user behavior and may lead you to further fine-tune the reviewed policy. This might involve adding missing objects to the policy, and adding parameters as well as parameter values. Through the real-life traffic, TrafficShield security application learns the real nature of legitimate requests and allows you to adapt the policy accordingly.

As real-life traffic is propagated through TrafficShield security application in none-blocking mode, the administrator can verify that:

- No false positive alarms have been posted.
- TrafficShield security application warns you in case real attacks are detected.

Stage 4 - Putting the policy into effect: blocking

You know that your policy is ready when all the alerts generated in the Learning tables represent invalid requests, such as one-off requests for invalid information or automated scripting attacks. The absence of false warnings ("false positives", that is, warnings on requests that are actually legal) means that your policy contains all the necessary objects and flows, and that all of the parameters are set to values that are characteristic of non-harmful, real-life traffic.

The next step is to activate TrafficShield security application's Blocking Mode. This can be done gradually, as the policy is more mature and tested. Through a set of simple checkboxes, you tell TrafficShield security application what to block. For example, by activating the "Illegal Object Type" blocking, TrafficShield security application will consider illegal any request referring to a file whose type is not included in the policy.

All the warnings that the Learning tool might return after you activate all of the desired blockings should be considered as potentially harmful behavior warnings. For more information about warnings that are generated after a first revision of the policy was performed, please refer to Chapter 4, *Policy Management Configuration*.



4

Policy Management Configuration

- Scope
- Add a new policy
- Policy properties
- Setting the active policy of a web application
- Blocking Policy table
- Other policy activities
- Flow properties
- Policy component editing
- Policy audit tools

Scope

This chapter explains the procedure for creating a new policy.

Defining a web application automatically creates a default policy in the TrafficShield Management Station (TSMS). The default policy deals only with negative security violations. If you wish to edit the policy and enhance it to deal with positive security violations also, you can either manually modify it, or run the Crawler to update the policy.

◆ Tip

*After modifying a Policy, always click the **Update TrafficShield** button to ensure that the policy is re-activated with the modifications.*

◆ Note

*A policy can be created only if at least one Web Application entry was created. For more details on how to define Web applications in the TrafficShield security application, please refer to Chapter 4, **Web Applications**, in the **TrafficShield® Installation and Configuration Manual version 3.2**.*

Add a new policy

1. Navigate to the **Policy Management** tab > **Policies List** tab.
A list of existing policies appears displaying additional information such as security level and blocking mode status. If you ran the TrafficShield configuration wizard, the first time you access this page you will see the policy you defined or selected via the wizard.



2. Click the **Add** button.
The Add New Policy page opens.

3. Enter the information described below and click **Save** to save your information. This will automatically open the Policy Properties tab.

Policy Name

Enter a name for this policy. You can use any name.

Web Application

Specify the address (www...) of the Web application to which this policy will be applied.

You can define different policies for the same Web application but only one policy can be active for a certain Web application at any given time.

Policy Description

Optionally, enter a few words that describe this policy.

Security Level

There are three security levels, **Standard**, **High** and **Custom**. Each level defines a different set of violation-driven actions (i.e., for which violation TrafficShield security application is blocking the request). The default security level is **Standard**.

***Tip:** You must save the policy before you can view the Custom security level and edit the violation driven actions. For more information, please refer to **Blocking Policy table**, on page 4-12.*

Disable Blocking

See *Blocking Policy table*, on page 4-12.

Max HTTP Header Length

The maximum allowed length for a header in a request processed by this policy. The value can be defined manually by the user or by the Learning process.

By choosing the **Any** button, any HTTP header length will be allowed.

Max Cookie Header Length

The maximum length a cookie processed by this policy is allowed.

By choosing the **Any** button, any Cookie header length will be allowed.

Flow Mode

Two flow modes are available: **Simple** and **Advanced**. The **Simple** flow mode is the default mode.


- By selecting the **Simple** button in the Flow mode area, the user is instructing the TrafficShield system Crawler and Learning processes to create a simplified policy, where all objects are defined as entry points.
- By selecting the **Advanced** button in the Flow Mode area, the user instructs TrafficShield system to automatically create the policy with a full flow mode.



***Tip:** Always maintain the same Flow Mode option that was used to initially create a specific policy. We do not recommended that you switch back and forth between **Simple** and **Advanced** flow modes.*

Policy properties

After creating a new policy for the web application, the policy is saved and its properties appear in the Policy Properties tab, which you can access by clicking the **Policy Properties** button on the left navigation panel.

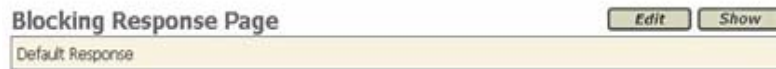
Editing the current policy's properties

The Active policy is the policy that currently protects the web application, and is always flagged with an icon .

After applying a modification to a policy, click the **Save** button. The change is saved in the system but not yet saved to the policy. At this stage, an icon  appears at the top of your screen, notifying you about the modification. Clicking the **Update TrafficShield** button saves the modification into the policy and removes the icon .

Blocking Response Page

TrafficShield security application has a default response page that it returns to the user in case the user request, or the response returned by the web server, is blocked by TrafficShield security application. In this case, the user can replace the default response page with a customized response page that he can define in the Blocking Response page section.



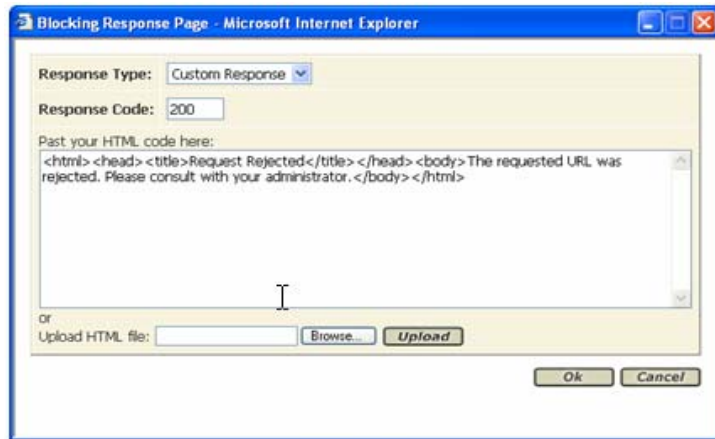
The response is an HTML page that you can build inside the TrafficShield security application, or load into the TrafficShield security application after creating the page elsewhere.

Click the **Show** button to display the current blocking response page in a popup window.

To edit a response page

1. In the **Blocking Response Page** section, click the **Edit** button. The Blocking Response Page opens.

2. Upon Completion, click **OK** to save your changes.



Response Type

This field defines the type of response page that will be displayed to the user. If you select the default response, you can see its HTML code but you cannot change it. The possible values that can be selected here are:

- Default Response - This is the default web page in the TrafficShield security application.
- Redirect URL - This means that instead of a web page, the TrafficShield security application returns to the user an HTTP redirect URL.
- Custom Response - This means that the user has defined that this is the page the TrafficShield security application will returned to the user.

Response Code

The HTTP response code returned to the user. We recommend that you not change the HTTP response code.

Paste your HTML code here

You can either paste or type the page's HTML code into the **Paste your HTML code here** field. Or upload a file in the next field.

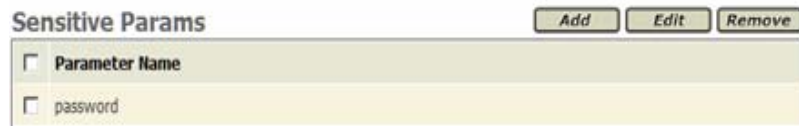
Upload HTML file

Use the browser button to select the HTML file that will serve as the response page, and click the Upload button to load the file as the response page.

Sensitive parameters

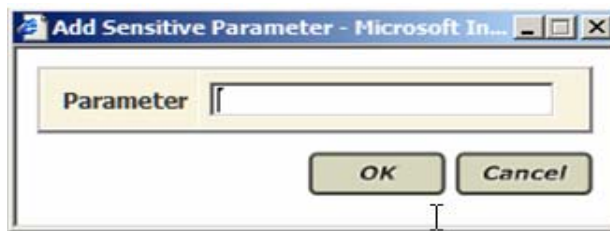
Incoming requests validated by TrafficShield security application are stored in plain text format. Some requests may include user input, such as a password or a credit card number, that you may not want to store once the

request has been processed (a string of asterisks will be stored instead of the actual value). You can avoid storing this sensitive data by entering the names of the input fields in the Sensitive Parameters sections.



To specify a sensitive parameter

1. Click **Add**.
The Add Sensitive Parameter box opens.



2. In Parameter, enter the name of a sensitive field.
Enter the name of the input field exactly as defined in the request.
For example:
http://siterequest.com/bank.php?account=12345
If you defined the field account to be a sensitive parameter, it will be displayed in the following manner:
`/bank.php?account=XXXXXX`
3. Click **OK**.

◆ Tip

*Upon installation, a sensitive parameter called **password** is created by default.*

Allowed modified cookies

You can set the policy to ignore certain cookies included in the request even if they do not meet the expected criteria. This is done in the Allowed Modified Cookies section by simply listing their names.

To define an allowed cookie

1. Click the **Add** button.
The Add Allowed Cookie box opens.



2. In **Cookie Name**, enter the name of an allowed cookie.
Enter the name of a cookie exactly as it is expected to appear in the request.
3. Click **OK**.

Allowed methods

TrafficShield security application accepts certain methods upon installation. The default methods are listed in this section when you first access it. See example below.

TrafficShield security application considers as invalid all requests that use HTTP methods other than those listed in the Allowed Methods section.

You can set other HTTP methods valid by adding them to the list.

Allowed Methods		
<input type="checkbox"/> Method Name	Act As Method	Check Trusted IP's for extended methods
<input type="checkbox"/> GET	GET	NO
<input type="checkbox"/> HEAD	GET	NO
<input type="checkbox"/> POST	POST	NO

To allow an additional method

1. Click **Add**.
2. The Add Allowed Method window opens.
3. Enter the new method's information and click OK to save and return to the Policy properties window.
-Or-
To exit the window without saving the information, click **Cancel**.

Method Name

Select the name of an allowed method.

Act as Method

Select the mode of operation allowed for the additional method.

Check trusted IPs for extended methods

Select the **Check trusted IPs for extended methods** checkbox to allow this additional method only if it appears in requests sent by one of the trusted IPs.

Clearing this checkbox will make the method valid in all incoming requests. For details about trusted IP addresses, see Chapter 4, *Web Applications*, in the *TrafficShield® Installation and Configuration Manual Version 3.2*.

Navigation Parameters

In some Web applications, pages are generated based on parameters that appear in the request.

If you want TrafficShield security application to differentiate between pages that are generated by requests with the same object name but with different parameters, and to build the appropriate flows, you need to specify the exact names of the parameters that triggered the creation of these pages in the web application. The parameter names are specified in the Navigation Parameters section.

◆ Note

*The two examples below demonstrate how the user can define a specific object path plus parameter, or, if the policy contains a common parameter used by more than one object path, how the user will need to define **Any** as the Navigation path, and the parameter name, as displayed below.*

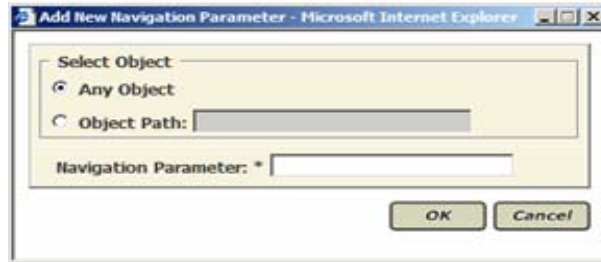
Navigation Parameters		Add	Edit	Remove
<input type="checkbox"/> Object Path	Parameter			
<input type="checkbox"/> /cgi-bin/neomail-prefs.pl	action			

Navigation Parameters		Add	Edit	Remove
<input type="checkbox"/> Object Path	Parameter			
<input type="checkbox"/> Any	action			

To specify a navigation parameter passed to the web server for dynamic page building

1. Click the **Add** button.
The Add New Navigation Parameter window opens.

2. Enter the new navigation parameter's information and click OK to save.



3. In **Select Object**, select one of the following:

Any Object

If the Web application consists of just one physical page (the index page), select Any Object.

Object Path

If the Web application contains physical pages and dynamic page building starts from one of them, select Object Path and enter the URL of that object.

4. In **Navigation Parameter**, enter the name of the parameter passed to the Web server for page building purposes.

Policy-specific negative regular expressions

When you create a new policy, the policy automatically inherits all of the negative regular expressions defined in the Administration tool, and these expressions are listed in this tab.

Existing policies do not inherit expressions that have been created after them. You can add policy-specific negative regular expressions by choosing the tab under **Configuration > Negative RegExp** and add them just like adding default regular expressions.

For more details, see *Assigning Expressions*, on page 2-8.

◆ **Tip**

Violations created due to negative regular expressions are related to illegal pattern violations.

Setting the active policy of a web application

At any given time, TrafficShield security application enforces only one of the available security policies. The security policy according to which the Web application is currently protected is called the active security policy.

You need to set the active security policy in the following cases:

- Before opening the Web application to user traffic, for testing or for regular business.
- Every time that you enter a change in the policy. If you do not re-activate the policy, the latest changes are not reflected to the Web application. A policy that has not been activated after it has been modified is marked with the **M** icon.
- Whenever you switch from one policy to another (see section below).

To activate a policy

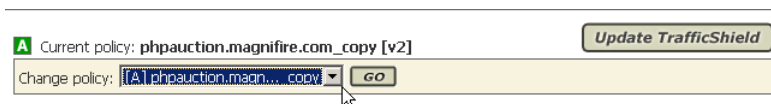
1. Access the Management screen.
2. Click the **Update TrafficShield** button in the upper right corner of the screen.
The currently edited policy is activated.

Switching between policies

In the Policy Management module, you can edit any existing policy, and switch between policies while working in the following pages:

- Policies List
- Policy Properties
- Object Types
- Web Objects
- Application Flow
- Negative RegExp
- Character Sets
- Policy Audit Tools.

In these pages, the following section appears at the top of the page, below the **Update TrafficShield** button.



To switch between policies

From the **Change Policy** drop-down list of existing policies, select the one you want. You can now view its settings, edit it or set it as the active policy.

Automatic update of policy versions

Policy versions are automatically updated every time that you click the **Update Policy** button.

For instance, before modifying the following sample policy, the policy was tagged [V3].

A Current policy: `phpauction.magnifire.com_copy [v3]` **Update TrafficShield**

Change policy: `[A] phpauction.magn..._copy` **GO**

Object Types **Add** **Save** **Remove**

<input type="checkbox"/>	Type	Check Objects	Check Flows	Is Referrer	Length		Query String Length	POST Data Length	Check Response
					Object	Request			
<input type="checkbox"/>	gif	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	28	575	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Suppose you changed the object length from 28 to 29, and then click **Save** and **Update TrafficShield**. The change will be saved into the policy, which will be tagged now as the active policy, with a new policy version [V4], as you see in the following example.

A Current policy: `phpauction.magnifire.com_copy [v4]` **Update TrafficShield**

Change policy: `[A] phpauction.magn..._copy` **GO**

Object Types **Add** **Save** **Remove**

<input type="checkbox"/>	Type	Check Objects	Check Flows	Is Referrer	Length		Query String Length	POST Data Length	Check Response
					Object	Request			
<input type="checkbox"/>	gif	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	29	575	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Blocking Policy table

To navigate to this table, in the **Policy Properties** section, choose **Policy management > Policy Properties > Security Level > Edit**.

In order to customize the security level, the user may edit the current default security levels. When the security level is saved, the modified security level is now labeled as “Custom”.

Blocking Policy: Standard Level				Make Active
<input checked="" type="checkbox"/> Disable Blocking				
RFC Violations				
Violation	Severity	Alarm	Block	
Illegal HTTP format	Warning	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Non-RFC request	Error	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Not RFC compliant cookie	Info	<input checked="" type="checkbox"/>	<input type="checkbox"/>	

In this page, you can define for each violation a related action that will be enforced by TrafficShield security application when such violation occurs.

When the **Disable Blocking** option is selected (Transparent Mode):

In case of violation, the end user will be able to access the requested page, and TrafficShield system will log the violation event. In such a case, the Blocking Mode of the policy will be set to **Transparent**.

When the **Disable Blocking** option is not selected (Block Mode), then if the Block flag for a violation is on, and this violation is detected by TrafficShield security application, the user receives the TrafficShield Blocking Response page followed by Support ID information. In such a case, the Blocking Mode of the policy will be set to **Blocked** with an additional Blocking Hand flag 🖐️.

The violations are categorized separately.

- RFC violations
- Access violations
- Length violations
- Input violations
- Cookie violations
- Negative security violations

The following sections describe various violations belonging to the above categories.

RFC violations

Filter	Description
Violation: Illegal HTTP format	Request line is illegal in the following cases: - Method, resource or HTTP version is missing. - HTTP version is not HTTP/1.0 or HTTP/1.1. - Host header is missing the method in the request. See <i>Allowed methods</i> , on page 4-7.
Non RFC request	Binary Data in the user input contradicts user input type or method.
Not RFC compliant cookie	Cookie format does not follow RFC.

Access Violations

Filter	Description
Illegal access to method by not allowed IP	Request was received from a Client IP that is not allowed to use the method in the request. See <i>Allowed methods</i> , on page 4-7.
Illegal domain (Web Application)	Host header value doesn't match any of the Web application FQDNs or Aliases defined in the TSMS.
Illegal entry point	The requested resource is not an acceptable entry page to the Web Application.
Illegal flow to object	The transition from the previous resource to the requested one is illegal.
Illegal method	The method is not defined in the policy properties as an allowed method.
Illegal object type	Requested resource type (extension) is not defined in the policy.
Non existent object	Requested object is not listed in the policy. To better understand, please refer to <i>Non-existent object</i> , on page 6-10.

Length violations

Filter	Description
Cookie length error	Cookie header value length exceeds the threshold set in the policy.
Header length error	Header name + value length exceeds the HTTP Header Length set in the Policy Properties.
Object length error	Resource name length exceeds the policy limit.
POST-data length error	Request method is POST and the user input data length exceeds the policy limit.
Query-string length error	The request Query string length exceeds the policy limit.
Request length error	Request length exceeds the maximum request length defined in the policy.

Input violations

Filter	Description
Failed to convert character	Some characters in the object or user input cannot be mapped into the Latin-1 characters table.
Forbidden Null in request	Forbidden null byte in request.
Illegal dynamic parameter value	Parameter value doesn't match the dynamically generated pool of legal values.
Illegal empty parameter value	Empty is not allowed for the specific parameter value.
Illegal meta character in parameter value	The parameter value contains a character that is set to "N" (false) in the Administration > Character Sets > User Input: language
Illegal number of mandatory parameters	The number of mandatory parameters in the flow is different from the number of mandatory parameters defined in the policy.
Illegal parameter	Parameter is not defined in the flow.
Illegal parameter data type	Parameter value differs from the type assigned to the parameter in the policy.
Illegal parameter numeric value	Numeric (decimal or integer) parameter value exceeds the value range set for it in the policy.
Illegal parameter value length	Parameter value length exceeds the length limitation set for it in the policy.
Illegal Query-String or POST-Data	Request contains user input not expected to be found in the flow.
Illegal static parameter value	Parameter value doesn't match any of the values in the Static pool of values for a given parameter.
Malicious parameter value	Parameter value matches one of the regular expressions describing common web attacks, i.e., XSS, SQL injection.
Null in multi-part parameter value	NULL character found in the parameter non-binary type in multi-parted POST-data.
Parameter value doesn't comply with regular expression	The Parameter value doesn't evaluate to the positive regular expression which defines the valid values for this parameter.

Cookie Violations

Violation	Description
Expired timestamp	TrafficShield cookie was returned after the TTL expired.
Modified Domain cookie(s)	The modified domain cookies. TrafficShield system has detected that the web application domain cookies have been modified by the client.
Modified TS cookie	The TrafficShield state cookie has been tampered with. TrafficShield security application has detected that the web application domain cookie has been modified by the client.
Wrong message key	Suspected TrafficShield cookie hijacking.

Negative security violations

Filter	Description
Illegal HTTP status in response	Server responded with HTTP status of type 4XX or 5XX. Statuses 400, 401, 404, 407, 503 are not included in this rule. These settings are configurable.
Illegal meta character in header	The HTTP header value contains a character that is set to N (false) in the Administration > CharSets > HTTP Headers field.
Illegal meta character in object	The Object part of the URI contains a character that is set to N (false) in the Administration > Character Sets > Object Path field.
Illegal meta character in parameter name	The parameter name contains a character that is set to N (false) in the Administration > Character Sets > Param Name.
Illegal pattern in header	One of the HTTP header values evaluates to at least one negative regular expression applied to the Header value . See <i>Negative Regular Expressions Policy Defaults</i> , on page 2-6.
Illegal pattern in object	Evaluates to a negative regular expression applied to the Object part of the URI.
Illegal pattern in response	Data in the server response matches negative regular expression applied to Response . Violation triggering is done by setting the Check Response flag of a specific object type to true.
Illegal pattern in user input	Evaluates to a negative regular expression applied to the Key-value pairs . Test is done on user input for both POST and GET methods.

During the Learning stage, the alarms should diminish. At this point you can be confident that all missing objects have been added, and other attributes are attuned to real-life traffic requirements. The blocking mode should be activated only after monitoring traffic without any Learning alarms for several days.

The trigger for activating the Blocking mode is any point in time that the user can reasonably assume that the policy is accurate: meaning, all resources are present and all attribute values meet the requirements of legitimate real-life traffic and, therefore, any further alarm should be considered as suspicious.

After activating the blocking mechanism, illegal requests may continue to appear in the Learning pages: you can still accept their suggestions if they are justified, or you can alternatively clear them out.

Blocking by categories

Blocking is implemented by instructing the TrafficShield security application on which violation to block the request.

For example, by setting the Block flag on **Illegal file types**, you instruct the TrafficShield security application to block a request if it tries to access an object of a type not included in the policy.

You do not have to activate all of the available blockings.

To set blocking categories

1. From **Policy Management > Policies List**, select the relevant policy.
2. Click the **Policy Properties** tab on the left side menu or the **Edit** button above the policy list to open the Policy Properties window. The properties displayed belong to the currently chosen policy.
3. In **Security Level**, select one of the standard levels, or select **Custom**, if this security level already exists.

The **Standard** level provides minimal blocking and the **High Security** level provides comprehensive blocking. The **Alarm/Block** set of flags of both levels may be edited and saved as a **Custom** security level.

The rest of this procedure relates to the **Custom** option. If you want to disable blocking temporarily, check the **Disable Blocking** checkbox in the **Policy Properties** tab; clearing the box reactivates the selected blockings.

4. Go over each blocking category and define what the TrafficShield security application should do when an illegal request matches the category's definitions. The options are:

Alarm

Check the **Alarm** checkbox to instruct the TrafficShield security application to only post an alarm to the Security Events log and the Learning pages without blocking the Web application user.

Block

This option acts like **Alarm**, but the request that triggered the violation is blocked.

You can check both boxes. Some **Block** boxes are checked and grayed, meaning that requests that commit that specific violation are always blocked.

5. Click **Make Active**, and then the **Update TrafficShield** button.

Using Learning in Blocking Mode

After you enable the blocking mechanism, the Learning system continues to analyze traffic. The requests that end up in the Learning tabs are those that contradict the policy. You can still accept some or all of them if they warrant policy changes, or clear them if they do not.

Other policy activities

There are several other activities that you may want to use with your policies. You have the option to:

- Edit a policy
- Remove a policy

Edit a policy

There are two ways to choose the existing policy you would like to edit.

To choose a policy via the Policies List

1. In the Policy Management > Policies List tab, select the relevant policy to edit by checking the radio button at the left of the policy name.

Policy	Web Application	Security Level	Last Set Active
<input checked="" type="radio"/> Test	phpauction.magnifire.com	Standard	last set by root at 2004-11-26 11:47:13
<input type="radio"/> phpauction.magnifire.com_default	phpauction.magnifire.com	High Security (APC)	active now, last set by root at 2004-11-26 12:31:17

2. Click the **Edit** button.
3. The policy properties window is automatically displayed for viewing or modifying.

To choose a policy via the Policy Properties window

1. Select **Policy Management > Policies Properties** tab.
2. Select the relevant policy from the Select Policy pull-down list and click the **Go** button.

The policy properties window is automatically updated to the selected policy for viewing or modifying.

Configuration >> Policy Properties Current User: root

Select Policy: PAErrors **GO**

Policy Properties

Policy Name*: PAErrors

Web Application: phpauction.siterequest.com

Policy Description: Flow: [HTTP] /index.php -> (GET) -> [HTTP] /search.php
Parameter Name: q

Security Level: Custom **Edit**

Disable Blocking:

Max HTTP Header Length: Any Length:

Max Cookie Header Length: Any Length: 579

It is also possible to choose and edit a policy from the **Learning** page as described in the following procedure.

To choose a policy via the Learning properties window

1. Select **Policy Management > Real Traffic** tab.
2. From the **Change Policy** drop-down list, select the relevant policy.
3. Click the **Go** button.
The learning properties are automatically updated according to the selected policy.
4. You can view these properties and edit them as required. Upon clicking the **Go** button, the policy becomes the current edited policy.
5. To set the current edited policy, click the **Update TrafficShield** button.

Remove a policy

You can remove a policy, provided that the policy is not active.

To remove a policy

1. Select **Policy Management > Policies List**.
2. Select the relevant policy to remove by checking the radio button at the left of the policy name.

Configuration » Policies List Current User: root

Policies List			
<input type="button" value="Export"/> <input type="button" value="Import"/> <input type="button" value="Copy"/> <input type="button" value="Add"/> <input type="button" value="Edit"/> <input type="button" value="Remove"/>			
Policy	Web Application	Security Level	Last Set Active
<input checked="" type="radio"/> PAErrors	phpauction.siterequest.com	Custom	active now, last set by root at 2004-09-27 12:16:47
<input type="radio"/> PAErrors_2	phpauction2.siterequest.com	Custom	active now, last set by root at 2004-09-26 17:22:02

3. Click the **Remove** button.
You are asked to confirm the policy removal.
4. Click **OK** to remove the policy.

◆ Note

*You cannot remove an policy that is currently active. Since it is not possible to deactivate an already activated policy, you must return to **Administration > Web Application** and make active another policy that belongs to the same Web Application. Only then you can return to the **Policies List** tab and remove the relevant policy. If the policy you want to remove is the only policy related to this Web Application, you will need to remove the Web Application.*

Export/Import a policy

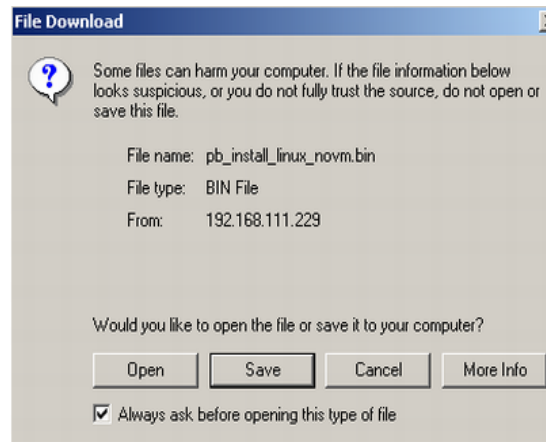
There are different reasons for using the Export/Import policy. The export/import feature can be used to export a policy and then import it, assigning it to a different Web application in the process.

This feature can also be used as a sort of backup and roll-back point in the policy life cycle.

To export a policy

1. In the **Policy Management** tool, select the **Policies List** tab and click the **Export** button.

The Standard File Download dialog box opens.



2. Click the **Save** button and save the policy file.

To import a policy

1. In the **Policy Management** tool, select the **Policies List** tab and click the **Import** button.

The Import Policy page opens.



2. Fill out the **Import Policy** page.

For Web Application

To populate this field, select one of the following:

- Select **Decide Automatically** to assign the imported policy to the Web application from which it was exported.
- Select another Web Application to assign the imported policy

Choose the File

In **Choose the File**, use the browser to select the file to import.

3. Click the **Go** button.

◆ **Note**

The imported policy appears in the Policies List. If the imported policy exists in the current TrafficShield security application environment, it is renamed (a sequential number is added to the end of the policy name).

Copy a policy

The purpose of this option is to quickly duplicate policies or create policies that differ only in a few details.

To copy a policy

1. In the **Policy Management** tool, select the **Policies List** tab and click the **Copy** button.
The Copy Policy page opens.

The screenshot shows the 'Configuration >> Policies List' interface. At the top right, it says 'Current User: root'. Below this, there is a search bar containing 'test' and a 'Select Policy:' dropdown menu with 'test' selected and a 'GO' button. Below the search bar is the 'Copy Policy' section, which has a 'New Policy Name: *' field containing 'test_copy' and a 'GO' button.

2. Verify that the relevant Policy has been selected.
3. Change the selected policy in the **Select Policy** pull-down list.
4. Click the **Go** button to change the selected policy.
5. The **New Policy Name** field in the Copy Policy window is automatically updated accordingly.
You can edit the **New Policy Name** if required.
6. Click the **Go** button to copy the policy.

7. In the **Policies List** tab verify that the newly copied policy is added.

Policy	Web Application	Last Set Active	Security Level	Blocking Mode
A crawler hadash [v3]	www.ana.co.jp	last set by root at 2005-03-24 14:59:31	High Security (APC)	Transparent
M crawler http https [v2]	www.ana.co.jp	last set by root at 2005-03-23 14:44:01	High Security (APC)	Transparent
M crawler https	www.ana.co.jp	N/A	High Security (APC)	Transparent
A M www.kabu.com_default [v1]	www.kabu.com	last set by root at 2005-03-23 14:44:46	Standard	Transparent
A M www.stemakr.ru_default [v10]	www.stemakr.ru	last set by root at 2005-03-24 15:40:34	High Security (APC)	Transparent

◆ Note

A green icon **A** next to a policy indicates that this policy is active. A red icon **M** indicates that the policy has been modified, and you must click the **Update TrafficShield** button to implement the change into the policy.

Roll back to a previous policy version

This feature enables you to revert to previously saved policy versions.

To roll back to a previously saved policy version

1. Select **Policy Management > Policies Properties**.
The list of policies appears.
2. Select a policy.
3. Click the **History** button.
The previously saved versions of the selected policy appear in the History page of the policy.

A Current policy: crawler hadash [v3] Update TrafficShield

History for crawler hadash Restore

Select	Version	Activated At	Active Until
<input type="radio"/>	v1	2005-03-24 08:34:02	2005-03-24 12:52:35
<input type="radio"/>	v2	2005-03-24 12:52:35	2005-03-24 14:59:41
<input type="radio"/>	v3	2005-03-24 14:59:41	N/A

4. Select the appropriate version and click the **Restore** button.
A new policy is created according to the selected policy version. The previously active policy version is not deleted but saved as a version. Upon clicking **Restore**, the selected policy version becomes automatically the active policy.

Flow properties

The following sections describe the flow properties parameters.

Target Object

In simple terms, this is the **to** side for a flow that runs **from** and **to** an object.

Referrer Object

This is the object from which the flow began its path to the Target Object.

Method

This is the action done on the Target Object. For example: GET, POST, PUT and Delete.

Target Frame

The Target Object will be loaded to this frame number. TrafficShield system uses interface frames.

Has QS/PD

This flag indicates whether the HTTP/HTTPS request (for the requested object) has a query-string or a POST-data.

Check QS/PD

This flag indicates whether the TrafficShield security application should verify if the request QS/PD complies with the policy. If the flag is TRUE, it enforces the defined policy of the request's QS/PD; and if the is FALSE, it does not check the QS/PD.

Number of Parameters

Maximum number of parameters in the HTTP/HTTPS request.

Parameter List

This lists the parameters that can appear in the HTTP/HTTPS request.

Policy component editing

This section explains how to manually edit the policy components.

Manual intervention in a policy built by the policy building tools may be needed if you want to override the definitions generated by the Crawler. For example, you may want to remove an object from the policy if you do not want TrafficShield security application to check requests that refer to it, or you can enter regular expressions to enhance the checks.

Most of the modifications made to a policy are typically done through the Learning tables. For example, you can add a missing object through a single click, once the Learning process has determined that the object should be part of the policy.

Refer to the beginning of this chapter for more details on the Learning process.

Adding Object types

The Object Types tab lists the existing file types in the protected Web site. For example, a list of valid object types for a specific policy could be: **GIF**, **JPG** and **HTML** only. If your policy contains the above list, then any request for a **PDF** file would be considered illegal.

The extensions are listed here to enable you to decide how the policy should react to requests that refer to files that have these extensions.

Each entry in the table is composed from the object type, and the object type's set of flags and values. When adding a new object to the policy, this set of flag and values is the default settings applied to the object.

◆ Note

A special entry of “no_ext” file type is created in the object type table to handle the following cases: Objects with no file extension, and Objects with file extensions longer than 8 characters.

The screenshot shows the 'Configuration » Object Types' interface. The current user is 'root'. The selected policy is 'PAErrors'. The 'Object Types' table is displayed with the following columns: Type, Check Objects, Check Flows, Is Referrer, Length URI, Length Request, Query String Included Length, POST Data Included Length, and Check Response. The table contains the following entries:

Type	Check Objects	Check Flows	Is Referrer	Length URI	Length Request	Query String Included Length	POST Data Included Length	Check Response
<input type="checkbox"/> html	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	24	1223	<input checked="" type="checkbox"/>	1	<input checked="" type="checkbox"/>
<input type="checkbox"/> ico	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	12	491	<input type="checkbox"/>		<input type="checkbox"/>
<input type="checkbox"/> log	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	46	733	<input type="checkbox"/>		<input type="checkbox"/>
<input type="checkbox"/> no_ext	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	32	21824	<input type="checkbox"/>		<input type="checkbox"/>
<input type="checkbox"/> php	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	19	5275	<input checked="" type="checkbox"/>	1	<input checked="" type="checkbox"/>
<input type="checkbox"/> zip	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	42	274	<input type="checkbox"/>		<input type="checkbox"/>

Checkboxes

The first column contains checkboxes used to mark the relevant entry.

Type

This is the file extension. Clicking on the object type link leads you to a list of Web objects of this type.

◆ Note

The Type field is case-sensitive; for example you can add both html and HTML and they will be treated as different object types.

Check Objects

If this checkbox next to an object is selected, TrafficShield security application checks requests for this object type to verify that the actual object exists in the Web application or is accessible via the application flow.

If this checkbox is not selected, TrafficShield security application lets through requests for this object type without checking whether the actual object exists in the Web application or is accessible via the application flow. Instead, it applies negative logic checks on the request's contents.

◆ Tip

If the Web application changes frequently, (i.e., a set of objects in the Web application are changed frequently) it is not a good idea to clear this box, in order to avoid massive warnings and rejections. We recommend that you read the Allowed Objects RegExp - Object list relaxation section to learn how to define a less strict set of Web application objects.

Check Flows

The Application Flow (path) is the defined access path leading from one object to another object.

Check this box to instruct the TrafficShield security application to test whether the requested object from a given object type is a legal flow and is based on the flow properties, and to check the parameters' names and values.

If you clear this box, when an object is added to the policy, the check flow flag used in the policy for this object is turned off.

This flag, as well as the other flag in this table, is the default setting for a created object.

Is Referrer

Check this box if objects of this object type may refer to other files. For example, HTML pages containing links to another file are referrers. Pictures and sound files cannot be referrers because these objects never contain links to other objects and are not web pages.

Length URI

This field defines the maximum legal length of the object's full path for this object type.

Length Request

This field defines the maximum legal length of the entire request for this object type.

Query String Included

Check this checkbox if requests for objects of this object type may include user input in the query string part of the request. If this checkbox is not selected, then the length of the Query String for this object type is defined as Zero, and if a request object that belongs to this object type contains any query string, a query string length violation will be generated.

◆ Tip

If the query string is empty, i.e., nothing is written after the question mark, the TrafficShield security application considers the request as an empty query string.

Query String Length

This field defines the maximum legal length of the user input in the query string part of the request. For example: In the following request, `abc.html?Name=John&X=2`, the actual query string length is 13 (`Name=John&X=2`).

POST Data Included

Check this checkbox if requests for objects of this object type may include user input in the POST data part of the request. If this checkbox is not selected, then the length of the POST Data for this object type is defined as Zero and if a request object that belongs to this object type contains any POST Data, a POST data length violation will be generated.

Post Data Length

This field defines the maximum legal length of the post request user input data.

Check-Response

Check this checkbox to activate Server response filtering by the TrafficShield security application. If checked, the html body of the response will be tested vs. the Negative Regular expression applied to the Server response. See the Negative RegExp section in this chapter.

To add an object type manually

If the Web application includes objects of a type not listed here, you can add them manually.

1. In **Object Types**, click the **Add** button.
The Add Object Type popup window opens.



2. Enter the file extension and click **OK**. (Type the extension without the period that appears in front of the extension.)
3. In the **Object Types** page, review the flags and values and set the policy for this object type, as explained above.
4. To save the changes, check the left checkbox next to the relevant entries and click the **Save** button.

◆ **Note**

In order to remove an object type, check the left checkbox next to the relevant entries and click the Remove button. All existing objects of this object type and all relevant flows and parameters will be removed from the policy.

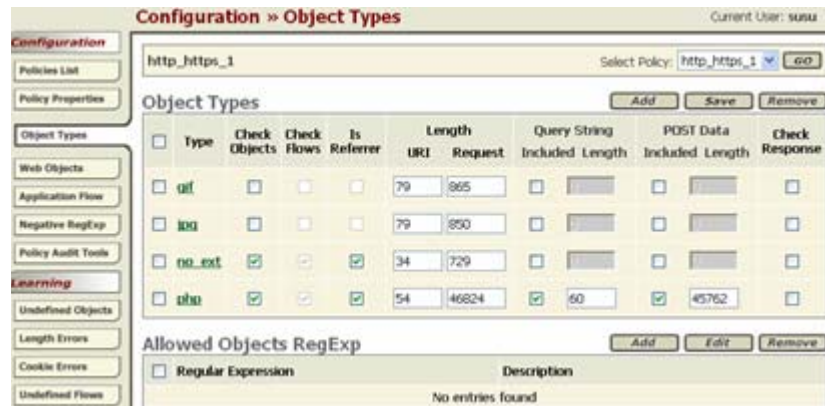
Allowed objects RegExp - Object list relaxation

The object list for a specific object type is enforced by the TrafficShield security application. For instance, if the Check Object flag is set for a specific object, any request containing an object that is not on the list will generate a “non-existent object” violation.

◆ **Note**

This violation will be reported to the user according to the blocking policy.

This situation is inconvenient if the Web application is dynamic and the set of objects of a given object type changes frequently. Adding and editing the object list manually or via the Learning process may become a complicated and endless task.



To resolve this problem, it is possible to define regular expressions describing the set of possible objects.

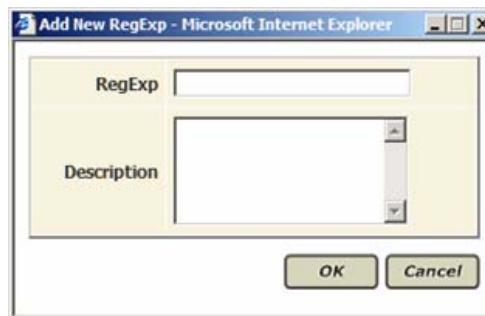
To define expressions as a set of possible objects

In the **Allowed Objects RegExp** section (located at the bottom of the Object Types window) follow these steps:

1. Set **Check Objects** to true.
2. Define regular expression(s) describing the set of possible objects as explained below.

To add a regular expression

1. Click the **Add** button.
The Add New RegExp dialog box opens.



2. In **RegExp**, enter the expression. For example, if the policy contains objects a.gif and b.gif only, the regular expression `*\.gif$` will allow any object of a gif object type.
3. Click **OK**.

If an allowed regular expression is defined in the policy, and TrafficShield security application does not find the requested object in the policy objects list, it checks if the object matches one of the allowed regular expressions. If it matches, then TrafficShield security application performs a negative logic check on the request's contents. If it does not match, then TrafficShield security application generates a non-existent object alert, and performs a negative logic check.

Web objects

After reviewing the object types, you can examine each object separately and fine-tune the security attributes for each of them.

An important policy decision to make at this stage is to decide whether a certain object is an entry point or not (in case you are working in simple mode, then most or all of the objects should be defined as entry points).

An entry point is a page through which a visitor should enter the Web application as designed by the Web Master of the application; for example, by typing its URL in the browser's address box, or by selecting its URL from a favorites list.

Your Web application may have several entry points. By defining objects that are entry points, you prevent an attacker from entering your Web application without passing through the “front door.”

To access the object list relating to a specific object type

1. Choose **Policy Management > Configuration > Web Objects**.
2. Choose the relevant object type in the drop down menu, and click the **GO** button.

The list of objects responding to your choice is displayed.

The screenshot shows the 'Configuration >> Web Objects' interface. At the top, there's a breadcrumb 'Configuration >> Web Objects' and 'Current User: susa'. Below that, there's a 'Configuration' sidebar with buttons for 'Policies List', 'Policy Properties', and 'Object Types'. The main area shows 'http_https_1' selected, with a 'Select Policy: http_https_1' dropdown and a 'GO' button. A 'Filter' section includes a 'URL Include:' text box and an 'Object Type: gf' dropdown with a 'GO' button, indicating '24 objects found'. Below the filter is a 'Web Application Objects (Site Map)' section with 'Show Flows', 'Add', 'Save', and 'Remove' buttons. The main table has columns: 'Is Entry Point', 'Is Referrer', 'Check Flow', and 'Accessible Objects List'. The table contains 5 rows of objects, all with 'Is Entry Point' checked and 'Is Referrer' and 'Check Flow' unchecked. The objects are: [HTTPS] /images/estrella_0.gif, [HTTPS] /images/estrella_0.gif, [HTTPS] /images/estrella_1.gif, [HTTPS] /images/estrella_1.gif, [HTTPS] /images/estrella_2.gif, [HTTPS] /images/estrella_2.gif, [HTTPS] /images/estrella_3.gif, [HTTPS] /images/estrella_3.gif, [HTTPS] /images/estrella_4.gif, and [HTTPS] /images/estrella_4.gif.

Is Entry Point	Is Referrer	Check Flow	Accessible Objects List
<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	[HTTPS] /images/estrella_0.gif
<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	[HTTPS] /images/estrella_0.gif
<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	[HTTPS] /images/estrella_1.gif
<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	[HTTPS] /images/estrella_1.gif
<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	[HTTPS] /images/estrella_2.gif
<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	[HTTPS] /images/estrella_2.gif
<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	[HTTPS] /images/estrella_3.gif
<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	[HTTPS] /images/estrella_3.gif
<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	[HTTPS] /images/estrella_4.gif
<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	[HTTPS] /images/estrella_4.gif

URL Include (Filter bar)

Use this field to view a subset of the object list. For example; type a string to list all the objects containing this string.

◆ Note

Each object in the list has a prefix which indicates the protocol (HTTP/HTTPS) through which this object may be requested. This may cause the same object to be displayed twice in the object list if relevant to both protocols.

◆ Tip

This search is case-sensitive.

Checkboxes

The first column contains checkboxes used to mark the relevant entry.

Is Entry Point

Select this checkbox if the object should be treated as entry point. In simple mode, most or all of the objects should be defined as entry points

Is Referrer

Check this box if this object may refer to other objects. This is important in case there are flows in the policy from this object to other objects. In simple mode most or all of the objects should be defined as non-referrers, since there are no **real** flows in simple mode policy.

Check Flow

Check this box to instruct the TrafficShield security application to test whether there is a legal flow in the policy to the requested object.

Accessible Objects List

Object list that answers the filter criteria. To open the Object Properties Window for a specific object in the list, click the object link. This window is divided into three parts:

- Object Properties
- Flows to Object
- Dynamic Flows from Object

Object properties

This section defines the object flags as displayed in the upper level, Web Objects tab.

Object Properties		Save
<input checked="" type="checkbox"/> Object Is Referrer	<input checked="" type="checkbox"/> Object Is Entry Point	
<input checked="" type="checkbox"/> Check Flows to this Object	<input type="checkbox"/> Object can modify Domain Cookie	
<input type="checkbox"/> Don't Block this Object		

Object is Referrer

Check this box if this object may be treated as a referrer to other objects. This is important in case there are flows in the policy from this object to other objects. In simple mode, most or all of the objects should be defined as non-referrers, since there are no **real** flows in simple mode policy.

Object is Entry Point

Select this checkbox if the object should be treated as entry point. In simple mode, most or all of the objects should be defined as entry points.

Check Flows to this Object

Check this box to instruct the TrafficShield security application to test whether there is a legal flow in the policy to the requested object.

Object can change Domain Cookie


If the object is a referrer, then this box can be checked. If the domain cookie was changed on the client side (i.e., Java script function execution by browser), then the TrafficShield security application will fail any request if this checkbox is not checked for this object and the object is a referrer in the incoming request

Don't block this object

You can use this flag to instruct TrafficShield security application to not block requests for this object even when a violation is detected in the request. This option should be used in case of a new object and when the policy for this object was not yet tested.

Flows to object

This section summarizes the flows to the object.

Flows to Object				Add	Save	Remove
<input type="checkbox"/> From Object	Method	Allow QS/PD	Check QS/PD	Frame Target		
<input type="checkbox"/>  [HTTP] item.php	GET	<input type="checkbox"/>	<input type="checkbox"/>	1		

Checkboxes

The first column contains checkboxes used to mark the relevant entry.

From Object

This column lists the objects from which the object could be accessed.

◆ Note

Click the object link to view the flow properties.

Method

This column specifies the method through which the object should be accessed.

Allow QS/PD

Check this checkbox to define whether Query string and/or POST data are allowed for that flow.

Check QS/PD

If Query string or POST data are allowed for the flow on the request, check this checkbox to enforce parameter and name validation.

Frame Target

This is the index of the HTML frame targeted by the flow. We do not recommend that you change this value unless you know that you want to specifically load this object into a specific frame.

◆ Note

The value 99 is a default frame index which means that the target object is loaded into the same frame as where the referrer object is presented.

Dynamic flows from object

Some flows cannot be defined upfront because the web site involves a constantly changing set of objects. For example: a zone of the application where various users store files that they can access later, involves unpredictable flows from the user personal archive page, since the users remove or add files daily.

In such cases, you can use the Dynamic Flows from Object section to legalize access to the changing sets of files.

Dynamic Flows from Object			Add	Edit	Remove
<input type="checkbox"/> Prefix	RegExp Value	Suffix			
<input type="checkbox"/> 			

After adding a dynamic flow from object, the object becomes a referrer object and the is referrer checkbox is automatically selected.

Checkboxes

The first column contains checkboxes used to mark the relevant entry.

Prefix

This field is a fixed substring of the html source page. It may be a name of a section in combination with html tags; for example: “<h3>Flows2Object</h3 >”.

RegExp Value

This field defines a set of objects in the above mentioned dynamic group.

Suffix

The suffix is similar to the prefix. For example: <form name=”dynamic_flows” >.

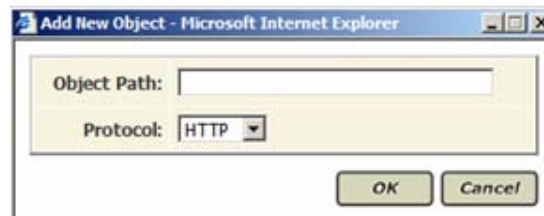
◆ Note

*The **Prefix** and **Suffix** settings instruct the TrafficShield security application of the boundaries that enclose the set of dynamic object links in a page. The TrafficShield system uses the RegExp value as a pattern to evaluate each object in the set between the boundaries.*

Adding a Web object

To add an object manually

1. If you want to manually add an object without running the Crawler again, click the **Add** button and the Add New Object window opens.



2. In the **Object Path** field, enter the full resource path starting with the slash [/].
3. In the **Protocol** field, specify the protocol to be used to access the object.
4. In the **Web Objects** tab, review and edit the flags and values for the new object.
5. Check the **modified entry** checkbox, and click the **Save** button.

Removing a Web object

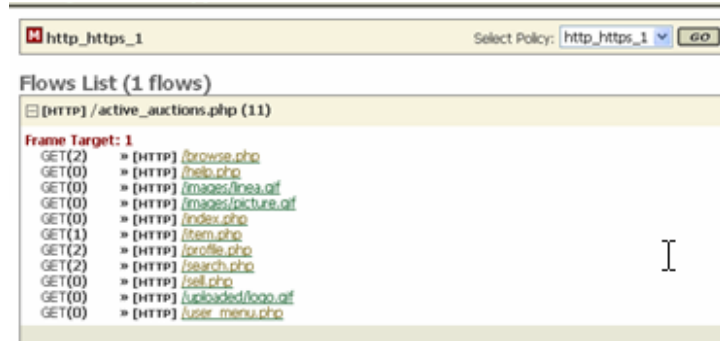
To remove an object

1. In the **Web Application Objects** list, check the relevant objects to be removed.
2. Then click the **Remove** button.
You will be asked to confirm the removal.

Displaying web application flow model

To show the objects' flows

1. Select the checkbox corresponding to the objects you want details on.
2. Click the **Show Flows** button to display a list of flows in the **Flow List** window for the checked objects.
The Flow List window displays the list of checked objects. For each flow, it displays the method (GET/POST), the number of parameters and the target object. Each object can be expanded to display the outgoing flows. For more details, please see the following section on application flow.



Application flow

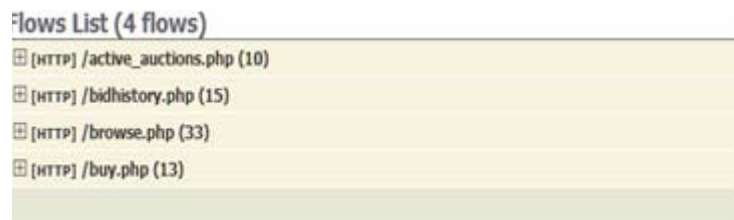
The Application Flow is the defined access path leading from one object to another object.

These flows are populated from various sources: The Crawler generates a map of the flows from within the Web application, by scanning the links and references within the objects. The Learning process results in acceptance of new flows. It is also possible to manually add and edit application flows.

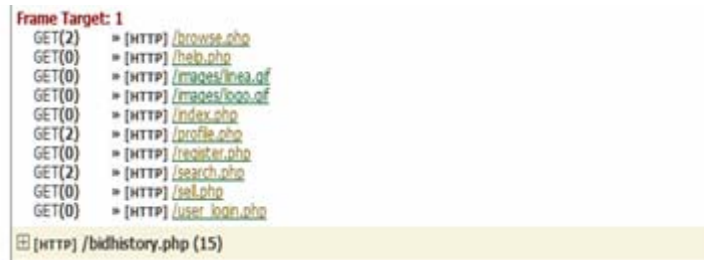
To access the application flow

The Application Flow can be accessed in any of the following three ways:

1. Choose **Policy Management > Configuration > Web Objects** tab.
2. Then click the desired object's URL link.
The **Flows to object** section of the page lists the objects from which the selected file can be reached.
3. Click the **From Object** link to display the Application Flow window.
4. Choose **Policy Management > Configuration > Web Objects** tab.
5. Then check the checkbox to the left of the relevant object (you can check more than one, if you want) and click the **Show Flows** button. This displays, at first, a list of the objects you have just marked.



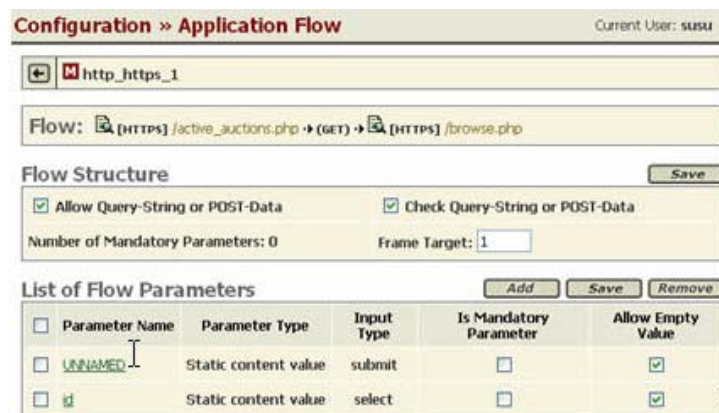
- Click the **+** button to see a list of the actual files that can be reached from the object you selected originally. If the reference targets a frame in a frameset, then the index of the target frame appears at the top of the referenced files.



- Click the **To Object** link to display the Flow window.

Destination Objects are listed under the Frame Target Index into which they should be loaded by the application. Each entry specifies:

- The method used to access the target object.
 - The number of known input parameters in ().
 - A protocol to request the target object.
 - Colorization of the targeted objects is used to differentiate between the Is Referrer flag settings (Brown=flag set to true, Green=flag set to false).
- Click the **Application Flow** tab. You see a list of all flows.



The TrafficShield security application allows the user to view and edit the Query String and the POST Data. The flow parameters configuration is only accessible from these windows.

Flow structure

Allow Query-String or POST-Data

Check this box if a request that accesses the selected object via this specific flow may also carry a query string or POST data.

Check Query-String or POST-Data

Check this box to instruct TrafficShield security application to perform validity checks on the query string and the POST data. This relevant only if you already checked the Allow Query-String or POST-Data checkbox.

Number of Mandatory Parameters

This number represents the number of parameters that must pass from the source to the destination object in this flow. This counter is updated automatically as additional parameters are marked as mandatory.

Frame Target

This is the index of the HTML frame targeted by the flow. We do not recommend that you change this value unless you know that you want to specifically load this object into a specific frame.

◆ **Note**

The value 99 is a default frame index, which means that the target object is loaded into the same frame as where the referrer object is presented.

List of flow parameters

Checkboxes

The first column contains checkboxes used to mark the relevant entry.

Parameter Name

This column displays a list of the flow parameters.

◆ **Note**

The Parameter Name “UNNAMED” is used for actual parameters on the flow that don't have a name.

Parameter Type

This field specifies the parameter type. See the parameter section below for details on the parameter types.

Input Type

This field defines the html input type of the parameter as it appears in the html source page.

Is Mandatory Parameter

Check this checkbox if this parameter must appear in the flow.

Allow Empty Value

Check this checkbox to allow the parameter to contain an empty value.

Adding manually a new application flow

This section explains how to add a new Application flow. Click **OK** after entering the new flow's information, and click the **Save** button to save your changes.

To manually add a flow

1. Choose the **Policy Management > Configuration > Web Objects** tab.
2. Check the relevant object to which you want to add a new flow definition.
3. Click the **Add** button.

The Add New Flow window opens:

Referrer Object

There are two possible referrer object types:

Entry Point

Choose this option if the object to which the flow should be added is an entry point.

Object Path

Choose this option and specify the referrer object path from which the target object should be accessed.

Protocol

Specify the **protocol** type by which the target object should be accessed.

Method

Choose the **method** by which the target object should be accessed.

Frame Target

This is the index of the HTML frame targeted by the flow. We do not recommend that you change this value unless you know that you want to specifically load this object into a specific frame.

◆ Note

The value 99 is a default frame index which means that the target object is loaded into the same frame as where the referrer object is presented.

◆ Tip

In order to decide what to enter to the frame target index field, the html source page should be reviewed for frame set tags.

Defining the Flow parameters

This section describes the parameter properties and its configuration.

1. To access this window, choose the **Policy Management > Configuration > Web Objects** tab.
2. In the **Web Objects** window, choose the “target object.”
3. From the list of **Flows to Object**, choose the “from object.”
4. The Application Flow window appears and displays a List of Flow Parameters.

List of Flow Parameters					Add	Save	Remove
<input type="checkbox"/>	Parameter Name	Parameter Type	Input Type	Is Mandatory Parameter	Allow Empty Value		
<input type="checkbox"/>	UNNAMED	Static content value	submit	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
<input type="checkbox"/>	id	Static content value	select	<input type="checkbox"/>	<input checked="" type="checkbox"/>		

Checkboxes

The first column contains checkboxes used to mark the relevant entry

Parameter Name

Specify the name of the parameter as it appears in the request.

To view and edit the parameter properties, click the Parameter Name link. The Edit Parameter window appears.

Parameter Type

This field specifies the parameter type.

Input Type

This field defines the html input type of the parameter as it appears in the html source page.

Is Mandatory Parameter

Check this checkbox if this parameter must appear in the flow.

Allow Empty Value

Check this checkbox to allow the parameter to contain an empty value.

To add a new parameter to the flow

Click the **Add** button in the **List of Flow Parameters** section in the **Web Application** tab.

◆ Note

The window contains two sections. In the top section, Add Parameter, the Parameter's general information is entered. The selected parameter type automatically changes the appearance and content of the bottom section. For example if you choose to add a parameter of a "static content value" type, the bottom section will display the Parameter Static Values screen.

Optional parameter types

Don't Check Value

Select this option if you do not want TrafficShield security application to check the parameter value at all. If you choose this option, no bottom section appears in the window.

◆ Note

A parameter defined as Don't Check Value must have a value in the request. The TrafficShield security application will not check its validity, but it will check its existence. To disable this functionality, check the Allow Empty Value box: this makes sure that empty parameters are also allowed.

Static Content Value

Select this option if users must select the value from a pre-defined list of values such as values found in a drop-down list or a list of values accessed via radio buttons. When this option is selected, the Parameter Static Values section appears.

Configuration >> Application Flow Current User: **susu**

← http_https_1

Add Parameter Save

Parameter Name: Is Mandatory Parameter

Parameter Type: Static content value Allow Empty Value

Input Type: text-input

Parameter Static Values

Remove All Remove Add

To build a list of pre-defined values

1. In the box next to the **Add** button, enter a value.
2. Click **Add**.
The value moves to the larger box.
3. Repeat this step to define all the values needed.

To remove a value from the list

Select the value and click the **Remove** button.
The **Remove All** button clears the entire list.

◆ Note

If the value list is empty for this parameter type, an illegal static parameter value violation is issued for any value received in this parameter in the request.

Dynamic content value

Use this option if the parameter value changes dynamically and the location of the value in the request cannot be foreseen. In this case, you instruct the TrafficShield security application to actually search for the value in the various sections of the request.

Configuration » Application Flow Current User: root

Current policy: **phpacution.core.com_default_copy** Update TrafficShield

←

Add Parameter Save

Parameter Name:

Parameter Type: Is Mandatory Parameter

Allow Empty Value

Input Type: text-input

Dynamic Parameter Properties

Extract Parameter from Object*:

Search in URL

Search in Form

Form Index:

Parameter Index:

Search in XML

XPath:

Search in Response Body

Find: All Occurrences Limit to Occurrences

Match: Prefix RegExp Value Suffix

Enter the following information (you can run the search in one or more of the sections described below).

Extract Parameter from Object

Define the object on which TrafficShield security application will perform the check of the dynamic parameter. You must define this object in the policy. This object becomes automatically a referrer object.

Search In URL

Check this box to instruct TrafficShield security application to search for the parameter value in the URL section of the request.

Search in Form

Check this box to instruct TrafficShield security application to search for the parameter value in one of the forms.

- In **Form Index**, specify the HTML index of the form that contains the parameter.
- In **Parameter Index**, specify the HTML index of the input parameter in the form that contains it.

Search in XML

Check this box to instruct TrafficShield security application to search for the parameter value in an XML block included in the request.

In the XPath box, specify the XML tag path (e.g., <products><productPrices > <productSalesPrice >) where to look for the value.

Search in Response Body

Check this box to instruct the TrafficShield security application to search for the parameter value between two specific strings in the body of the request.

Enter the following information:

Item	Description
Find:	
All Occurrences	Select this option to search for all occurrences of the value.
Limit to... Occurrences	Select this option to search for the first x occurrences of the value. Specify the number of occurrences to find.
Match	
Prefix	Enter the string that constitutes the starting point of the search in the request body.
RegExpValue	Enter a regular expression that describes the searched value (and parameter name, if necessary).
Suffix	Enter the string that constitutes the ending point of the search in the request body.

Value characteristics for user input values

Select this option if the parameter accepts input from the user. For example it may be applied to html text area, input box, etc.

This option allows you to set the value's data type and to define the characters it may contain.

The screenshot shows a web interface for configuring a parameter. At the top, there is a browser address bar with 'http_https_1'. Below it is the 'Add Parameter' section with a 'Save' button. It includes fields for 'Parameter Name', 'Parameter Type' (set to 'User-input value'), 'Input Type' (set to 'text-input'), 'Is Mandatory Parameter' (checkbox), and 'Allow Empty Value' (checkbox). The 'Parameter Characteristics' section below it has a 'Data Type' dropdown set to 'Alpha-Numeric English'. It contains several checkboxes for validation rules: 'Check Minimum Value', 'Check Maximum Value', 'Check Maximum Length', and 'Regular Expression'. At the bottom, there are two columns of checkboxes for 'Allowed Meta Characters' and 'Allowed Regular Expressions', with some options like '(0x3b)', '(0x7c)', '(0x21)', '(0x26)', and '(0x20)' visible.

Data Type

Select the type of the parameter value. By selecting a type, you instruct the TrafficShield security application to consider as invalid any request that contains data of a different type for this parameter.

Select	To limit the value to
Alpha-Numeric (language)	Any text consisting of letters, digits and the underscore character.
Integer	Whole numbers only (no decimals).
Decimal	Numbers only (including decimals).
E-mail	Text in e-mail address format only.
Phone	Text in telephone number format only.

Select the “Don't check” option if you do not want the TrafficShield security application to check the type of the parameter value.

Check Minimum Value

For numeric parameters of Integer/Decimal types, you can set a minimum value. A request that passes a parameter with a lower value is then considered illegal.

To set the minimum value, check the box and enter the value.

Check Maximum Value

For numeric parameters of Integer/Decimal types, you can set a maximum value. A request that passes a parameter with a higher value is then considered illegal.

To set the maximum value, check the box and enter the value.

Check Maximum Length

This attribute applies to all data types except the Don't check parameter type.

By setting a maximum length for parameters, you prevent unauthorized access via parameter values that have an unexpected length. For example, you can limit the length of an alpha-numeric value to 4 (characters) if it is never expected to contain more than 4 letters, and thus instruct the TrafficShield security application to consider as illegal any requests that contain a longer value.

To set the maximum length, check the checkbox and enter the maximum number of characters the value may contain.

Regular Expression

If the value is non-numeric, you can calculate it via a Regular Expression. To do so, check this checkbox and type the expression in the adjacent field.

This is a positive regular expression that defines what is legal.

Allowed Meta Characters

Use this section for characters defined as C (check) in the **Character Sets table > Parameter values** in the **Administration tool**. The TrafficShield security application will let through requests whose user input includes the characters marked here as valid. That is, C will be treated as Y (true). Please refer to the *Character sets*, on page 2-4 for more details on Character Sets.

Allowed regular expressions

This is a list of regular expressions designed to protect the Web applications from common attacks via user input, like XSS, SQL injections, etc.

The user may allow a specific negative regular expression if the value of the parameter normal input matches the negative regular expression.

Defining negative regular expressions

The Negative Regular Expression tab contains a list of default and user-defined regular expressions. These regular expressions are meant to complete the security policy definitions.

The request/response content that matches at least one negative regular expression should be dropped.

Each regular expression may be modified to apply to one of the following parts of the request/response:

- Request URI
- Request key value pairs
- Request header values
- Server Response data (html body)

To build character sets

1. Click **Policy Management > Configuration**.
2. Click the **Character Sets** tab.
3. In **Select Char. Set** list, open the list and select the application element or input language for which you want to define a valid character set.

The options are:

Option	Allows you to determine the characters allowed in
Object charset	The name of the web object.
Parameter name charset	Parameter names.
Headers charset	The header section of an HTTP request.
Language names	User input in a specific language. For example, if your Web application supports French and you select User Input: French, data typed in by Web application users in form fields is verified against the French character set.
Parameter Value charset	Value section of the parameter value.

4. After you select an option, TrafficShield security application displays an entire character set.

Select Char. Sets: Object Path						Action: Y = YES, N = NO, C = CHECK					
Hex	Char	Action	Hex	Char	Action	Hex	Char	Action	Hex	Char	Action
0	n/a	N	40	●	N	80	n/a	N	c0	n/a	N
1	n/a	N	41	A	Y	81	n/a	N	c1	n/a	N
2	n/a	N	42	B	Y	82	n/a	N	c2	n/a	N
3	n/a	N	43	C	Y	83	n/a	N	c3	n/a	N
4	n/a	N	44	D	Y	84	n/a	N	c4	n/a	N
5	n/a	N	45	E	Y	85	n/a	N	c5	n/a	N
6	n/a	N	46	F	Y	86	n/a	N	c6	n/a	N
7	n/a	N	47	G	Y	87	n/a	N	c7	n/a	N
8	n/a	N	48	H	Y	88	n/a	N	c8	n/a	N

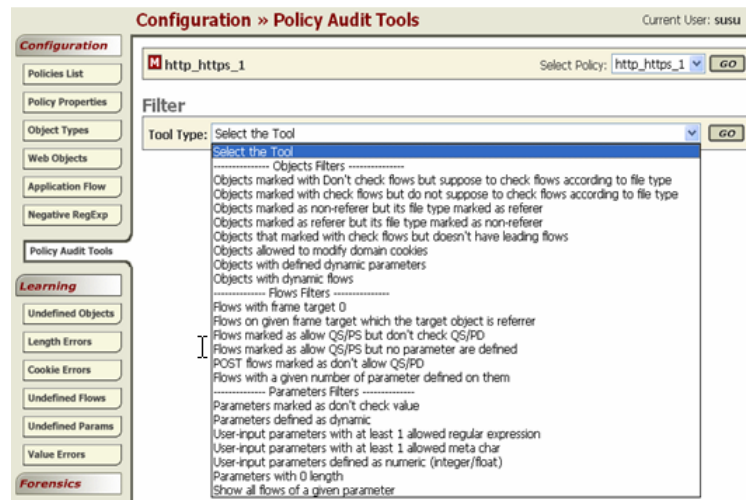
5. In the **Action** field of each character, select one of the following actions.

Action	Means
N	No. The character is invalid. An incoming request that contains this character will be blocked.
Y	Yes. The character is valid. An incoming request that contains this character will be let through.
C	Check, is equal to N, unless its explicitly defined as allowed in the Parameter Characteristics table under Application Flow (Policy Management tool). If the character is allowed there, then the request is valid. <i>C is not available for Header charset, Object charset, Parameter Name charset.</i>

6. To restore to the default character set definitions, click the **Restore Defaults** button.
7. Click **Save** to save the settings.

Policy audit tools

Since viewing all the policy in one screen is quite impossible, TrafficShield security application includes several filters that enable you to query the policy in order to find the information you are looking for. Some of these filters can be used to analyze suspicious policy states (i.e., Object without flows, Parameters with zero length, etc.). Each report isolates a pre-defined state and assists the user in identifying conflicts & errors in the policy.



After having chosen the appropriate filter from the list, click the **Go** button. TrafficShield security application displays the results of the query.



5

Crawler

- Crawler overview
- Populating the policy using the Crawler
- Configuring and launching the Crawler
- Crawler Learning tool

Crawler overview

This chapter explains how to configure, start, and manage the TrafficShield security application Crawler tool.

It provides step-by-step instructions to create an initial policy using the Crawler tool. The Crawler scans your application and builds a preliminary map of your Web application. This chapter also provides instructions on how to use the more advanced Crawler parameters.

Populating the policy using the Crawler

The TrafficShield security application Crawler automatically populates the security policy with the components of the Web application such as the HTML files, the picture files, the form fields, the links, and the flows that lead from one object to the other.

When you run the Crawler for the first time on a policy, it populates the policy with the current objects (application elements). The next time you run the Crawler:

- It collects only the objects that were added after the last run.
- It can be instructed to place the newly-added objects in a series of tables instead of adding them to the policy. This allows you to examine the new objects and decide what to do with them - add them to the policy or reject them. For additional details, please refer to the Data Collection with Policy Browser section in this document.

Configuring and launching the Crawler

The Crawler can be configured in several ways.

We recommend that first time users make use of the Crawler Wizard.


Advanced users may prefer to manually edit the Crawler settings and manually start it.

If your Web application has several entry points, you can instruct the Crawler to scan the application from each entry point separately. This is the advised method if your Web application site is combined from two or more unconnected parts.

Starting the Crawler using the Wizard

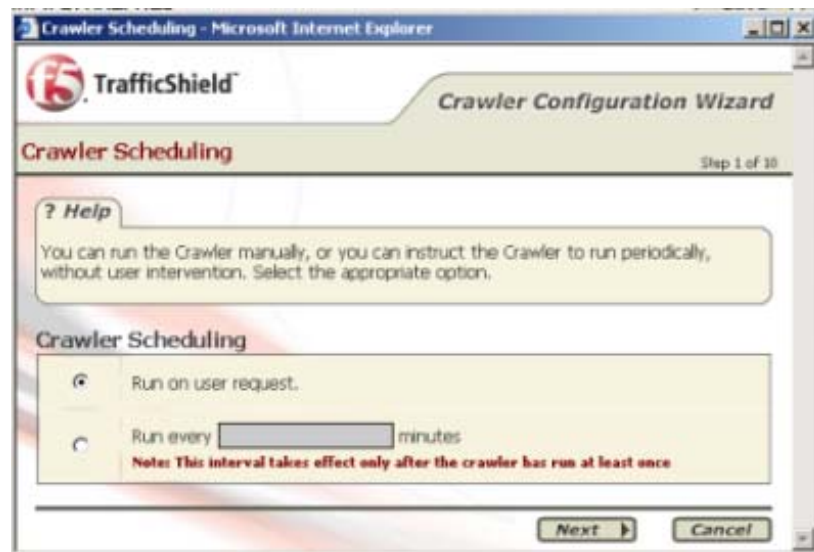
The Crawler Wizard will guide the user through a configuration stage, and enable starting the Crawler.

To access the Crawler Wizard

1. Under **Policy Management**, click **Policy Properties > Build Tools**.
2. Click the Crawler icon  .
The Crawler wizard is launched and Crawler Step 1 page appears.

Crawler scheduling - Step 1 in Crawler Wizard

You can run the Crawler manually, or set it to run periodically. You define this in the Crawler Scheduling section.



To configure and/or start the Crawler

1. Select the relevant policy for which the Crawler settings will apply, **Policy Management > Policies List**.
2. Click the **Policy Properties** tab or the **EDIT** button to open the policy in order to edit it.
3. Go to the **Build Tools** section and, per your desired work mode, begin to work with the Crawler.

To set a schedule

1. Select one of the following options:
 - **Run on user request**
Use this option if you want to run the Crawler at your command. You can run the Crawler at any time you choose, you just click its **Start** button in the Build Tools section.
 - **Run every... minutes**
Use this option to automatically run the Crawler every X minutes. Click the button, and in the **Run every... minutes** box, type the number of minutes you want between Crawler cycles. (For instance, if you want the Crawler to run every 10 minutes, type **10**.)
2. In the Crawler Scheduling window, click the **Save** button to save your settings, and continue.
Or you can click the **Cancel** button to exit the Wizard without saving your selections.

Start points - Step 2 in Crawler Wizard

The Crawler starts the data collection process from a URL. This is the start point.



The screenshot shows the 'Start Points' configuration window in the TrafficShield Crawler Configuration Wizard. The window title is 'Crawler Configuration Wizard' and the current step is 'Step 2 of 10'. A help box explains that the crawler enters the web application and follows links, and that multiple entry points can be specified. Below the help box, there is a section for 'Start Points' with 'Add' and 'Remove' buttons. A table lists the start points, with the first one being 'Start Point URL' and the second one being 'http://phpauction.core.com/'. At the bottom, there are 'Back', 'Next', and 'Cancel' buttons.

Start Points
<input type="checkbox"/> Start Point URL
<input type="checkbox"/> Select domain http://phpauction.core.com/

The start point is usually the Web application's home page. You can instruct the Crawler to start scanning sections of the application from other points as well, in case the application includes sub-applications that cannot be accessed through the home page, but only directly from a sub-URL.

To add Crawler start points

1. Click **Add**.
A new line is added to start points list.
2. In the **Domains** drop down list, select the domain to which the start point belongs.
A start point can be specified either as part of this Web application's Fully Qualified Domain Name, or as part of one of its aliases. Select the domain or the alias to use. You must make a selection.
The selected domain or alias appears in the Start Point text field.
3. Add the start point (a file name) to the end of the domain or alias string in the Start Point text field.
The resulting string must be a valid path specification, or it will be rejected.
4. Repeat this procedure to define all relevant starting points.

Form Filler - Step 3 in Crawler Wizard

Since the Crawler emulates user behavior, it submits data, in Web application pages, in the same way users do.

Each time the Crawler is activated, it populates the Form Filler Parameters Table with previously undefined parameter names.

If this is the first time you start the Crawler, all parameters are new to the Crawler and therefore it will most likely fail to submit any forms.

The next logical stage is to enter the crucial values needed to properly submit forms, for example: user name, passwords, etc. Sometimes the fields' names are not self-explanatory and you will need to consult the web application programmer.

If you know what crucial parameters and values should be defined before running the Crawler the first time, you can enter them to help the Crawler utilize the Web application on the first run.

To use this feature, you specify the names and data types of the fields as well as the values the Crawler should enter in them.

Form Fillers Step 3 of 10

? Help
As the crawler emulates user behavior, it may be required to enter data in Web application forms in the same way users do. For each form field the crawler will encounter, enter the appropriate information.

Form Fillers Add Remove

<input type="checkbox"/> Parameter Name	Parameter Type	Parameter Value
<input type="checkbox"/> TPL_address	text-input	
<input type="checkbox"/> TPL_birthdate	text-input	
<input type="checkbox"/> TPL_city	text-input	
<input type="checkbox"/> TPL_email	text-input	
<input type="checkbox"/> TPL_feedback	text-input	
<input type="checkbox"/> TPL_name	text-input	
<input type="checkbox"/> TPL_nick	text-input	
<input type="checkbox"/> TPL_password	password	
<input type="checkbox"/> TPL_phone	text-input	
<input type="checkbox"/> TPL_prov	text-input	

Back Next Cancel

To add a customized parameter

1. In the **Custom Parameters** section, click **Add**.
An empty line is displayed.
2. In **Parameter Name** and **Parameter Type**, specify the name of the field and its data type.
3. In **Parameter Value**, specify the value you want the Crawler to enter in the field.
4. Click **OK**.
The Page not found criteria page appears.

Page not found criteria - Step 4 in Crawler Wizard



When a request to a non-existing page comes in, Web applications return the standard HTTP 404 error page. This page may be exploited to stage attacks. To prevent this, some Web applications may use error pages of their own that don't return the HTTP 404 status code. They do this so that their content can be controlled and verified.

If your Web application uses such custom-tailored error pages, you need to supply a text string that the pages contain, so that the Crawler can identify them as a valid error message page and add it to the policy. If the “page not found” criteria is not defined, the Crawler will attempt to identify it by itself.

When an error occurs, the policy makes sure that only an error page whose content is recognized is returned to the request's sender.

TrafficShield security application can recognize an error page by its filename or by text included in its <TITLE> or <BODY>.

◆ Tip

In re-direct cases: The Crawler always follows the re-direct link. The Crawler identifies the page behind the link and avoids the link if the identified page is included in the Page Not Found list.

To identify a customized error page

1. Click the **Add** button.
A new empty line of page not found criteria is added.
2. In **Apply to**, select one of the following options to identify the error page:
 - Full Object Name
Its full file name. In **Search Item**, enter the file name.

- **HTML Title**
The text entered in its <TITLE> section.
In **Search Item**, enter the text.
 - **HTML Body**
Any string of text that appears in its <BODY> section.
In **Search Item**, type the string.
3. In **Search Item**, enter the indicated value and click **OK**.

Logout pages - Step 5 in Crawler Wizard

If the Web application contains a page designed to log the Web application visitor out, you need to instruct the Crawler not to follow the logout link as this will cause the Crawler to log out of the application before has fully scanned the application. In fact, many Web applications have an “exit” or “logout” link right in their home page, which would cause the Crawler to exit as soon as it enters the application. To prevent this, use the Logout Pages section to identify the logout points that the Crawler should avoid.

◆ Note

The logout page will be added to the policy.

To define a logout point

1. Click the **Add** button.
A new empty line of Logout Pages is added.
2. In **Logout Pattern (URL)**, enter the relative path of the logout page.
3. Click **OK**.

Properties - Step 6 in Crawler Wizard

The Properties section provides additional instructions to the Crawler. For example, you can instruct the Crawler to analyze Java Script code included in the Web Application or to skip it.

To specify properties

1. Enter the information described in the sections below.
2. Upon completion, click the **Save** button in the **Properties** window to save your entries.



Analyze JavaScript

Check this box to instruct the Crawler to analyze the JavaScript code included in the Web application. This is useful if the scripts contain links that can be followed, or if they include fields that need to be filled.

Clear the box if JavaScript analysis is not necessary.

Accept un-trusted SSL certificates

An un-trusted SSL certificate is used by the Web application and this checkbox option is checked, the Crawler accepts the SSL certificate and continues scanning.

Clear this box to instruct the Crawler to accept only trusted certificates.

Create back flows

As the Crawler runs, it always registers the page that follows a certain page over a link, thus adding the application flows to the policy. You can access each such flow definition and further configure it in order to establish rules of passage from one page to another.

By checking this box, you instruct the Crawler to also register in the policy all flows in the opposite direction, in which case you can also impose rules on navigating backwards (which occurs when the visitor uses the Back button).

Create cache flows

Cache flows are created around cacheable objects. The flow is created from the first non-cacheable referrer object around the cacheable object. The parameters of the incoming flow will be added to the newly created cache flow.

When no previous non-cacheable referrer object is found, the cacheable object itself becomes the entry point and the flow is added.

Min. delay between worm requests to web application (in sec.)

The Crawler is a mechanism that can be likened to a central unit sending out multiple probes to the different areas of the Web application in order to register Web application components simultaneously. Each probe behaves as if it were a real user, following links and filling in forms, and therefore increases traffic.

The probes can be sent in quick or slow succession. Quicker bursts create more traffic. A burst is measured in terms of the number of seconds to wait before sending the next probe. If your Web application is active and currently serving visitors, consider increasing this value in order to slow down the Crawler.

Number of threads to be used by the Crawler

This parameter also relates to simultaneous probe activity. A smaller number decreases the Crawler's bandwidth consumption, leaving more bandwidth to actual visitors.

Number of times the Crawler fetches requests with the same structure

Applications usually have many identical structures where only the parameter values differ. The following examples illustrate identical links passing different parameter values:

`http://www.myapp.htm?par=111`

`http://www.myapp.htm?par=222`

`http://www.myapp.htm?par=333`

To reduce crawling time and traffic you can instruct the Crawler to scan only a few of such identical structures and not all of them, assuming that all others behave in the same way.

Specify the number of samples you deem it sufficient for the Crawler to scan. A higher value yields a more accurate policy with longer crawling times.

Maximum number of requests generated for each form by the form iterator

When the Crawler encounters a form, it processes it as many times as the number of pre-defined parameter values included in it. For example, a drop-down list containing ten values causes the Crawler to process the form ten times, each time with a different value. However, you can reduce crawling time and traffic by instructing the Crawler to process only a few of the values and not all of them.

Specify the number of samples you deem it sufficient for the Crawler to process from the same form with different values. A higher value yields a more accurate policy with longer crawling times.

Emulate browser

If your Web application is set to work only with a given Internet browser, set the relevant browser name.

This name will be used to select the user-agent header data.

Default character set for user input fields

Select the character set in which data is normally entered in the form fields of the scanned application. This value will be used as the default value for all new policy fields added by the Crawler.

HTTP authentication - Step 7 in Crawler Wizard

Use this option only if your Web application uses HTTP authentication.



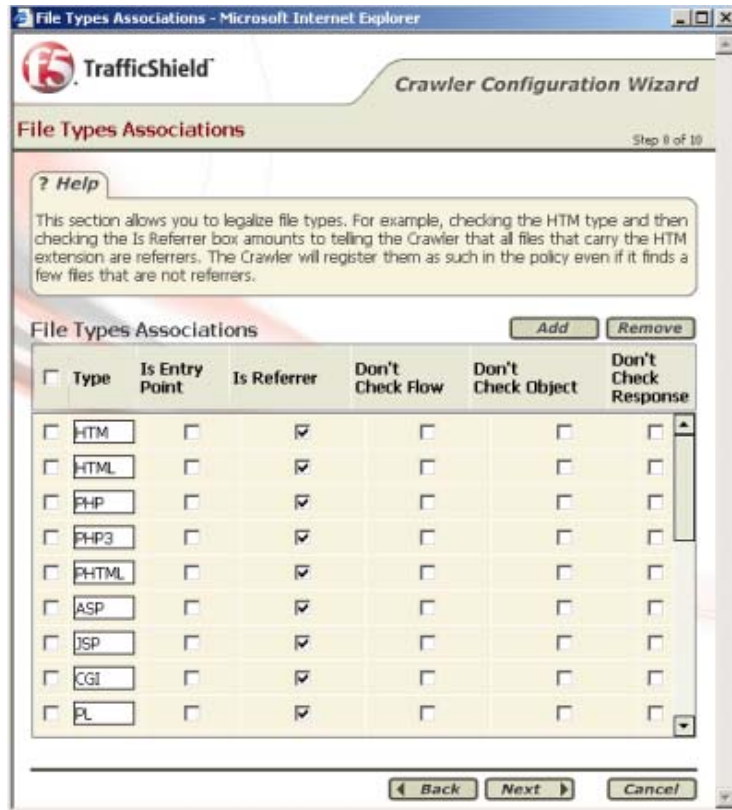
The screenshot shows a web browser window titled "HTTP Authentication - Microsoft Internet Explorer". The page displays the "TrafficShield" logo and the "Crawler Configuration Wizard" interface. The current step is "HTTP Authentication", labeled as "Step 7 of 10". A help box contains the text: "Use this option only if your Web application uses HTTP authentication. Specify the username and password the Crawler should supply in order to access the server where the Web application resides." Below the help box, there are two input fields: "HTTP Authentication Username:" with the value "root" and "HTTP Authentication Password:" with masked characters "****". At the bottom of the form, there are three buttons: "Back", "Next", and "Cancel".

Specify the user name and password the Crawler should supply in order to access the server where the Web application resides.

File type associations - Step 8 in Crawler Wizard

This section provides a list of file types frequently used in a Web application and their most common usage in the Web application.

It allows you to configure file types globally, thus saving tedious manual configuration in the policy. For example, you can instruct the Crawler to define all BMP files as files that do not have a flow.



If the list does not include a file type, you need to configure it.

- ◆ Click the **Add** button, add a file extension, and click **OK**.

The defaults provided in this page cover the most plausible eventualities, but you can adapt them to your needs by checking or clearing boxes.

A description of the file type configuration parameters follows.

Is Entry Point

Check this box if all files of this type can be entry points to the Web application.

Is Referrer

Check this box if objects of this object type may refer to other files. For example, HTML pages containing a link or CGI files calling another file, are waverers. Pictures and sound files cannot be waverers because these objects never contain links to other objects and are not web pages.

Don't Check Flow

Check this box if you don't want the system to check the flows to objects of this file type.

Don't check object

Check this box to if you don't want the system to check the requests referring to files of this type.

◆ Note

This will also be applied to files that do not exist in the application.

Crawler configuration settings - Step 9 in Crawler Wizard

This page displays the Crawler settings you defined in previous pages.

To modify the configuration, click the Back button until you reach the relevant step, and modify the data.



If the settings are correct, at this stage you should click the **Finish** button and run the Crawler.

To manually configure the Crawler

1. Click the **Settings** button.
The Crawler settings window appears. Each group of parameters is displayed in a separate box.
2. Enter the Crawler settings as described in the previous sections.

- Return to **Policy Properties** by clicking the **Back** button, located on the upper left side of the policy properties window.

Data collection with policy browser

The Policy Browser collects data that the Crawler can later use as a sort of fine-tuning input. The Policy Browser also overcomes browsing obstacles.

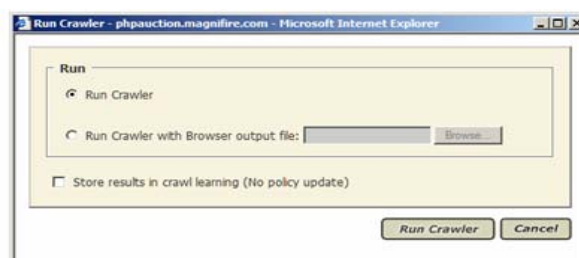
The data is collected by simply browsing the application as you would browse it with a regular browser. The browsing information processed by the browser is stored in a file. It is advisable to use the Policy Browser extensively and let it collect as much data as possible to later help the Crawler create a more accurate policy.

For instructions on how to download the policy browser and how to create the input file refer to the *Downloads* section in Chapter 6 *Administration*, of the *TrafficShield® Installation and Configuration Manual Version 3.2*.

Running the Crawler Manually

To manually start the Crawler

- Select the relevant policy for which the Crawler settings will apply, from the **Policy Management > Policies List**.
- Open the policy for editing by selecting the policy you want to work on and clicking on the **Policy Properties** tab or the **EDIT** button.
- In **Build Tools**, click the Crawler's **Start** button. The Run Crawler dialog box opens.



- Select the appropriate options and click **Run Crawler** to run the Crawler, or click **Cancel** to exit without running the Crawler.

Run Crawler

Choosing this radio button runs the Crawler as is, without the additional information supplied by the Policy Browser.

Run Crawler with policy browser output file

Run the Crawler and also use Web application details pre-recorded by the Policy Browser. Click the **Browse** button and select the

Policy Browser's output file. For additional information on how such a file is created please refer to the *Data Collection* section in this chapter.

Store results in crawl learning (no policy update)

Select this checkbox to activate the Crawler Learning process. For more details, please refer to *Crawler Learning tool*, on page 5-15.

5. Click the **Run Crawler** button.
The Crawler starts collecting data.
While the Crawler is running, you can click the **Status** button to open a window where you can see how the operation is progressing.



The message **Running** appears at the top of the window while the Crawler is still running. During this time, the dialog box displays the number of objects and flows that have been scanned and identified. Click the **Status** button to display the current status, without waiting for the next automatic refresh operation. The status window title changes to “Finished” when the operation ends. You can also monitor the process by accessing the other tabs in the navigation bar on the left.

Crawler Learning tool

This section explains how to use the Crawler Learning tool and how to adapt the policy using the Crawler Learning tool's output.

The Crawler Learning tool enables the user to scan the Web application in a learning mode.

When we use the Crawler in a non-learning mode, the Crawler populates the policy with the new items.

When the Crawler is set to work in a Learning mode, it populates the crawler learning tables with the new items instead of directly populating the policy tables.

You can then review the data and accept object types, objects and flows that were found by the Crawler and then add or reject them.

Crawler Learning tabs are identical to the Learning tabs. Both Learning and Crawler Learning populate the forensics section.

First-time usage: Crawler Learning can be used to update an existing policy or to initialize a policy. When updating a policy, the Crawler works in update mode and writes all the incrementally new items to the Crawler Learning tables. It doesn't change the existing policy items. When populating an empty policy, all items appear in the Crawler learning tables. In both cases you need to accept the item if you want to add it to the policy.

Second time usage: Unlike the regular Learning, once the Object is accepted and added to **Configuration > Web Objects** tab, all relevant flows are not automatically added to the policy. In order to add the relevant flows, you will need to re-run the Crawler or the Crawler learning.

◆ Tip

*If an item is rejected permanently, it is moved to Forensics > Ignore Items. This affects the Learning stage as well. For more details, please refer to **Ignored requests**, on page 6-48.*



6

Learning - Testing & Fine Tuning the Policy

- Overview
- Access violations
- Length violations
- Input violations
- Negative security violations
- Cookie violations
- Forensics

Overview

After automatically generating a policy using the Crawler and making any manual changes needed, you are ready to test and refine the policy in real-life conditions, through the Learning tool and the Policy editing tools.

This chapter explains how to use the Learning tool to adapt the policy to real-life traffic requirements. It also covers the Policy editing feature, which allows you to view and manually adjust the entire security policy.

Learning tool

The Learning tool was created so you could fine tune the Crawler-created security policies. This is relevant both for the first activation of the TrafficShield security application and as an ongoing tool as well.

In each case, the Learning screens are actually suggesting changes to the policy which would include all future requests of this nature. You can accept objects or flows that were rejected by the TrafficShield security application, and reject changes to the policy that were caused by actual attacks which were screened out.

◆ Tip

Customize your blocking definitions to temporarily allow some violations to go through until the Learning fine-tuning is more complete.

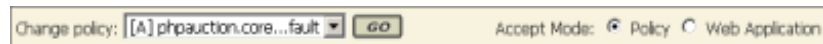
First-time usage: As the Web application is new; you may prefer to run an initial test in safe conditions. Such conditions can be created by opening the Web application to a limited number of visitors like Quality Assurance (QA) and employees of your organization (persons who are not potential hackers). Initially, the alarms help you adjust policy attribute values until you are sure that the policy is usable. Any invalid request that might come after the Learning stage can justifiably be considered illegal and treated as such. In fact, after the initial testing period you can use the Learning tool to track real attacks.

Ongoing usage: If the Web application to be protected is already in use, a portion of the live traffic can be diverted through the TrafficShield security application to the Learning tool.

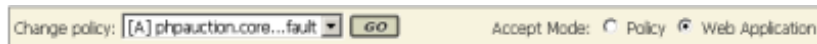
As visitors move through the Web application, the TrafficShield security application captures requests that contradict your current policy settings, and posts alarms to the Learning tool pages.

The Learning tool checks that all objects that are supposed to exist in your Web application are indeed present (for example, all links lead to objects that exist in the Web application). It also checks that the attributes specified for policy objects, such as URI lengths or allowed meta characters, are realistic.

In all the Learning windows, the fine-tuning changes can be applied to a specific policy:



The Learning fine-tuning changes can also be applied to a Web application which will affect all related policies:



In the Learning tool, the System displays recommendations to the user on how to fine tune the policy. Learning is not an analysis tool. There are situations that will be recorded in the Forensic that will not be converted into a policy. Note that the Learning tool saves Learning recommendations in the Learning tables at account level for the whole account. If a policy is deleted, the learning recommendations will be saved and displayed.

If you have several policies that are related to the same web application, in order to build a policy, you must first ensure that the policy is active, and then select its radio button in the Policies List screen.

Policies List					Export	Import	Copy	History	Add	Edit	Remove
Policy	Web Application	Last Set	Active	Security Level	Blocking Mode						
<input type="radio"/> High Secur1	phpauction.core.com	N/A		High Security (APC)	Transparent						
<input checked="" type="radio"/> [A] M phpauction.core.com_default [v2]	phpauction.core.com	last set by root	at 2005-04-17 17:32:29	Standard	Transparent						

Learning duration

The aim of the Learning process should be to generate traffic on all pages, to click all links, to fill all form fields, and so on. For new web applications, standard customer workflow routines can be used for Learning. For live applications, even a 15 minute test might supply valuable information that will help you fine-tune the policy. Obviously, the longer the test, the greater the opportunities to capture information that may help you establish a safer policy.

Auto Accept build tool

The Auto Accept Build tool enables the Security Manager to adapt the policy to accept automatically specific illegal requests recorded in the Forensics and make them legal.

◆ Note

The Auto Accept tool must be handled with ultimate care due to its immediate and comprehensive impact on the policy, as it automatically includes the selected violations into the policy, making them legal. In this

aspect it is distinguished from the Learning procedure, which provides only hints about the violations and requests the user to accept each of them manually into the policy.

To access the Auto Accept tool

1. From the **Policy Management** tool, click **Policy Properties > Build Tools**.



2. Click **Settings** to open the Settings screen.

3. Select the appropriate **Request**, **source IP**, **Request Time Range** and **Requested Objects**. These are the filters according to which the requests will be filtered. In the **Request Object** section you can limit the filtering by **Mask** and **Regular Expressions**.
4. In the **Accept Types** section, define the objects that you wish the policy to accept as entry points.
5. Once the settings are completed, click **Save** to save the settings.
6. Click the **Back** button at the top left side of the screen. You are returned to the previous screen.
7. Click the **Start** button. You are required to confirm the Auto Accept run.

8. Click **Run Auto Accept** button.

The Auto Accept process starts running and upon completion, an information message appears, providing information about the process.

Started at: 2004-12-21 17:55:15
Finished at: 2004-12-21 17:55:16

- Objects Types found: 8
- Objects found: 4
- Flows found: 8

Accessing the Learning data

To access the Learning data

In the **Policy Management** module, select **Learning** > **Real Traffic**.
The Real Traffic screen opens and a comprehensive list of violations groups appears.

Learning » Real Traffic
Current User: susu

Learning for Policy: crawl_copy

Change policy: crawl_copy
Accept Mode: Policy Web Application

Select all violations

Access Violations	
Learning of	Occurrences
<input type="checkbox"/> Illegal object type	0
<input type="checkbox"/> Non existent object	0
<input type="checkbox"/> Illegal flow to object	1
<input type="checkbox"/> Illegal method	0

Length Violations	
Learning of	Occurrences
<input type="checkbox"/> Length Errors	0

Input Violations	
Learning of	Occurrences
<input type="checkbox"/> Illegal Query-String or POST-Data	0
<input type="checkbox"/> Illegal Parameter	0
<input type="checkbox"/> Illegal static parameter value	0
<input type="checkbox"/> Illegal empty parameter value	0
<input type="checkbox"/> Illegal parameter value length	0
<input type="checkbox"/> Illegal parameter numeric value	0
<input type="checkbox"/> Illegal parameter data type	0
<input type="checkbox"/> Illegal meta character in parameter value	0
<input type="checkbox"/> Malicious parameter value	0

Negative Security Violations	
Learning of	Occurrences
<input type="checkbox"/> Illegal meta character in header value	0
<input type="checkbox"/> Illegal meta character in object	0
<input type="checkbox"/> Illegal meta character in parameter name	0
<input type="checkbox"/> Illegal meta character in parameter value	0
<input type="checkbox"/> Illegal pattern in object	0
<input type="checkbox"/> Illegal pattern in response	0
<input type="checkbox"/> Illegal pattern in header value	0
<input type="checkbox"/> Illegal pattern in user input	0

Cookie Violations	
Learning of	Occurrences
<input type="checkbox"/> Modified Domain Cookies	0

◆ **Note**

*The **M** that appears in the top menu next to the Policy name indicates that a modification has been done to the policy. Although all changes made to the policy were recorded in the database, they are not yet implemented until you activate the policy by clicking the **Update TrafficShield** button. Until then, the policy will act according to its previously defined parameters.*

If the TrafficShield system Learning process detected requests that were generated, then those detected violations appear in green and are underlined and linked.

- Select the policy violation you wish to review.

Violation grouping

Violations detected by the TrafficShield Security module are grouped as follows:

- Access Violations
- Length Violations
- Input Violations
- Negative Security Violations
- Cookie Violations

Access violations

This section is divided into five parts:

- Illegal object type
- Non existent object
- Illegal flow to object
- Illegal entry point
- Illegal method

Illegal object type

The Illegal object type window lists information about requests that referenced object types not found in the Web application. The object type is considered undefined unless you define it in the **Configuration > Object types** section.

<input type="checkbox"/>	Type	Occurrences	Check Objects	Check Flows	Is Referrer	Max. Request Length	Max. Object Length	Max. Query-String Length	Max. POST-Data Length	Check Response
<input type="checkbox"/>	gif	1	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="text" value="596"/>	<input type="text" value="22"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="checkbox"/>
<input type="checkbox"/>	jpg	1	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="text" value="566"/>	<input type="text" value="46"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="checkbox"/>
<input type="checkbox"/>	php	14	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="text" value="692"/>	<input type="text" value="15"/>	<input type="text" value="36"/>	<input type="text" value="0"/>	<input type="checkbox"/>

It is possible to manually change the value of some of the parameters. If the parameter is editable, it will appear as a user input box.

Checkboxes

The first column contains checkboxes used to mark the relevant entry.

Type

Check the checkbox for the relevant Object (file) type that you want to add to the policy.

Occurrences

This number indicates the number of request occurrences that were rejected for this type of violation.

- If you click the linked occurrence number, a **View requested objects** window appears containing a list of all the objects that caused this violation.
- If you click an object link, the **View full request information** window appears showing all the technical details of all the violations related to the specific request.

For more details please refer to *View Full Requests Information window*, on page 6-47.

Check Objects

Select this checkbox to instruct the TrafficShield security application to check the requests for this object type in order to verify that the actual object exists in the Web application, or is accessible via the application flow. The non-existent object window is automatically populated, and displays with all the objects belonging to all the requests for this object type.

If this checkbox is not selected, TrafficShield security application lets through requests for this object type without checking whether the actual object exists in the Web application or is accessible via the application flow.

Check Flows

Check this checkbox to instruct the TrafficShield security application to test whether the requested object from a given object type is a legal flow.

Is Referrer

Select this checkbox if objects of this object type may refer to other files.

Max. Request Length

The maximum request length received from all the requests for this object type.

Max. Object Length

The maximum object length received from all the requests for this object type.

Max. Query String Length

The maximum Query string length received from all the requests for this object type.

Max. POST Data Length

The maximum Post data length received from all the requests for this object type.

Check Response

Select this checkbox to activate the Server response filtering by the TrafficShield security application.

Available actions for Illegal Object Type

Accept

Clicking the **Accept** button adds the changes to the policy.

Accept means that you have decided that the request reflects a real-life situation that warrants a change in the policy.

The undefined objects types will appear under the **Configuration > Object Types** section.

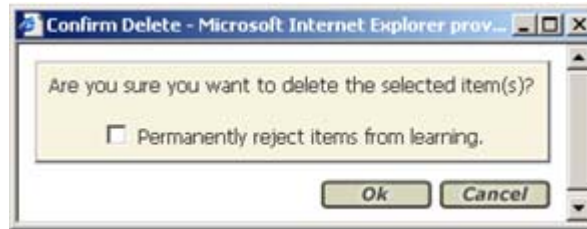
When you accept an Object type, the non-existent object window is automatically populated and displayed with all the objects belonging to all the requests for this object type. For example; if you accepted an HTML object type, all HTML requests' objects will now appear in the non-existent object window. See the next section to learn more about how to accept a non-existent object.

◆ **Note**

Requests with the accepted object types will still not be allowed by the TrafficShield security application until all the request's components have been "learned".

Clear

Clicking **Clear** deletes the selected entries in this learning window without changing the policy. The confirmation window appears.



Permanently reject items from learning

Select the **Permanently reject items from learning** checkbox to delete the request and instruct the TrafficShield security application not to register again identical requests. The deleted request is stored in the **Forensics > Ignored Items**.

◆ Note

After transferring the requests to Ignored Items, all similar requests for all policies that belong to this Web application will ignore these requests.

◆ WARNING

If you only want to apply this clear to this specific policy - don't check this checkbox. For example: if you checked this checkbox for HTML requests, all HTML requests (even rejected requests coming in for other policies will be ignored).

◆ Tip

*To change this decision after clicking Ok, you can go to Policy Management > Forensics > Ignored Items tab to unset the ignore decision. For more details, see **Forensics**, on page 6-45.*

Non-existent object

The Non Existent Object window lists information about requests that referenced objects that are not found in the policy.

Non-existent object						Accept	Clear
<input type="checkbox"/> Object	Occurrences	Entry Point	Is Referrer	Check Flow	Cookie Change		
<input type="checkbox"/> [HTTP] /help.php	1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
<input type="checkbox"/> [HTTP] /andrew.aristo.html	1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
<input type="checkbox"/> [HTTP] /browse.php	2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
<input type="checkbox"/> [HTTP] /sell.php	2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
<input type="checkbox"/> [HTTP] /index.php	1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
<input type="checkbox"/> [HTTP] /images/transparent.gif	1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
<input type="checkbox"/> [HTTP] /	3	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		

Checkboxes

The first column contains checkboxes used to mark the relevant entry.

Object

This column displays the name of the non-existent object.

Occurrences

This number displays the number of requests and values that caused this violation.

Entry Point

An entry point is a page through which a visitor enters the Web application, for example, by typing its URL in the browser's address box or by selecting its URL from a favorites list.

By checking this checkbox you instruct the TrafficShield security application to consider this object as a valid entry point.

Is Referrer

Check this box if files of this type may refer to other files. For example, HTML pages containing a link or CGI files calling another file, are referrers. Pictures and sound files cannot be referrers because they do not link to any other pages.

Check Flow

The Application Flow (path) is the defined access path leading from one object to another object. For example, a list of valid flows would be:

from abc.html to abc.gif, OK

from abc.html to def.html, OK

If your policy contains the above list, then any request that tries to access abc.gif from def.html would be considered illegal.

Check this checkbox to instruct the TrafficShield security application to verify that the object was accessed by a legally defined flow.

Some of these checkboxes are checked by default and cannot be cleared by the user.

If you clear the checkbox, the object can be requested from any place in the Web application or even when the user is outside the scope of the application.

Cookie Change

Select this checkbox if the client-side code of the object modified one of the Web application cookies in order to prevent false positive alarms on cookie poisoning.

Available actions for non-existent objects

Accept

Clicking the **Accept** button adds the changes to the policy. Accept means that you have decided that the request reflects a real-life situation that warrants a change in the policy.

Clear

Clicking **Clear** deletes the checked entries from this learning window without changing the policy. A confirmation window is displayed.



Permanently reject items from learning

Select the **Permanently reject items from learning** option to delete the request and instruct the TrafficShield security application not to register again identical requests in the Learning tables. The deleted request goes to **Forensics > Ignored Items**.

◆ Note

After transferring the requests to ignored items, all similar requests for all policies that belong to this Web application will ignore these requests.

◆ WARNING

If you only want to apply this clear to this specific policy, don't check this checkbox.

◆ Tip

To change this decision after clicking **Ok**, you can go to **Policy Management > Forensics > Ignored Items** tab to unset the ignore decision. For more details, see **Forensics**, on page 6-45.

Illegal flow to object

The Illegal Flow to Object screen is divided into two sections:

- Illegal Flow to Object
- Illegal Entry Point.

The Illegal Flow to Object screen lists the flows that were requested but were not found in the policy. In this case too, you can configure the query string and POST data settings of the Illegal flow to object flow and include them in your policy by clicking the **Accept** button.

Illegal flow to object						Accept	Clear
<input type="checkbox"/>	Flow	Method	Occurrences	Frame Target	Allow QS/PD	Check QS/PD	
<input type="checkbox"/>	[HTTP] / → [HTTP] /index.php	GET	1	@ 99	<input type="checkbox"/>	<input type="checkbox"/>	
<input type="checkbox"/>	[HTTP] /search.php → [HTTP] /sell.php	GET	1	@ 1	<input type="checkbox"/>	<input type="checkbox"/>	
<input type="checkbox"/>	[HTTP] /sell.php → [HTTP] /sell.php	GET	1	@ 1	<input type="checkbox"/>	<input type="checkbox"/>	
<input type="checkbox"/>	[HTTP] /sell.php → [HTTP] /sell.php	POST	2	@ 1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
<input type="checkbox"/>	[HTTP] /user_login.php → [HTTP] /index.php	GET	2	@ 99	<input type="checkbox"/>	<input type="checkbox"/>	


Checkboxes

The first column contains checkboxes used to mark the relevant entry.

Flow

This is the name of the Application Flow (path) which defines the access path leading from one object to another object.

◆ Note

The  indicates that the object is not a referrer. If the object should be defined as a referrer, go to the Policy Management > Configuration > Web Object window, and modify the definition of the object so that it is defined as referrer. Only after this operation is completed, it is possible to accept the violation.

Method

This is the HTTP method used in the Request. For more details refer to RFC-2610 (HTTP).

Occurrences

This field displays the number of illegal flow to object violation occurrences.

- If you click the linked occurrence number, a **View requested objects** window appears containing a list of all the objects that caused this violation.
- If you click an object link, the **View full request information** window appears showing all the technical details of all the violations related to the specific request.


Frame Target

This is the index of the HTML frame targeted by the flow. It is not recommend that you change this value unless you know that you want to specifically load this object into a specific frame.

User Input

User Input fields allow the user to enter a valid value that overrides the defaults.

The value 99 is a default frame index which indicates that the target object is loaded into the same frame as where the referrer object is presented. An empty value in the Frame target is allowed and accepting this empty value accepts it automatically under the 99 value.

Click the magnifying glass icon  next to the frame target value to open its screen

Allow QS/PD

Check this box if a request that accesses the selected object via this specific flow may also carry a query string or POST method.

Check QS/IPD

Check this box to instruct the TrafficShield security application to perform validity checks on the query string or POST data if allowed in the previous step.

If you clear the checkbox, the object can be requested from any place in the Web application or even when the user is outside the scope of the application.

Available actions for illegal flow to object

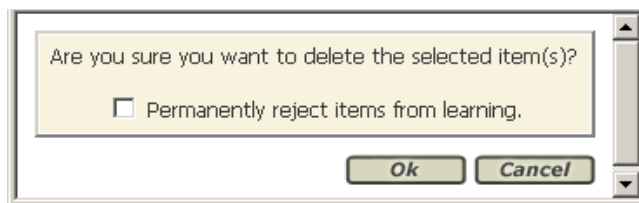
Accept

Clicking the **Accept** button adds the changes to the policy.

Accept means that you have decided that the request reflects a real-life situation that warrants a change in the policy.

Clear

Clicking **Clear** deletes the checked entries from this learning window without changing the policy. The confirmation window is displayed.



Permanently reject items from learning

Check the **Permanently reject items from learning** checkbox to delete the request and also instruct the TrafficShield security application not to register again identical requests in the Learning tables. The deleted request goes to **Forensics > ignored items**.

◆ Note

After transferring the requests to ignored items, all similar requests for all policies that belong to this Web application will ignore these requests.

◆ WARNING

If you only want to apply this clear to this specific policy, don't check this checkbox.

◆ Tip

*To change this decision after clicking **Ok**, you can go to Policy Management > Forensics > Ignored Items tab to unset the ignore decision. For more details, see **Forensics**, on page 6-45.*

Illegal entry point

Illegal entry point				Accept	Clear
<input type="checkbox"/> Flow	Method	Occurrences	Frame Target	Allow QS/PD	Check QS/PD
<input type="checkbox"/> Entry Point →  [HTTP] /browse.php	GET	1	@ 1	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> Entry Point →  [HTTP] /sell.php	GET	1	@ 1	<input type="checkbox"/>	<input type="checkbox"/>

Checkboxes

The first column contains checkboxes used to mark the relevant entry.

Flow

This is the entry point access to the object.

Method

This is the HTTP method used in the Request. For more details refer to RFC-2610 (HTTP).

Occurrences

This field displays the number of illegal flow to object violation occurrences.

- If you click the linked occurrence number, a **View requested objects** window appears containing a list of all the objects that caused this violation.

- If you click an object link, the **View full request information** window appears showing all the technical details of all the violations related to the specific request.


Frame Target

This is the index of the HTML frame targeted by the flow. It is not recommend that you change this value unless you know that you want to specifically load this object into a specific frame.

User Input

User Input fields allow the user to enter a valid value that overrides the defaults.

An empty value in the Frame target is allowed and accepting this empty value accepts it automatically under the 1 value.

Click the magnifying glass icon  next to the frame target value to open its screen.

Allow QS/PD

Check this box if a request that accesses the selected object via this specific flow may also carry a query string or POST method.

Check QS/PD

Check this box to instruct the TrafficShield security application to perform validity checks on the query string or POST data if allowed in the previous step.

Available actions for illegal entry point

Accept

Clicking **Accept** adds the changes to the policy.

Accept means that you have decided that the request reflects a real-life situation that warrants a change in the policy.

Clear

Clicking **Clear** deletes the checked entries from this learning window without changing the policy. The confirmation window is displayed.

Permanently Reject items from Learning

Select **Permanently reject items from learning** to delete the request and also instruct the TrafficShield security application not to register again identical requests in the Learning tables. The deleted request goes to Forensics > ignored items.

◆ Note

After transferring the requests to ignored items, all similar requests for all policies that belong to this Web application will ignore these requests.

◆ WARNING

If you only want to apply this clear to this specific policy, don't check this checkbox.

◆ Tip

To change this decision after clicking Ok, you can go to the Policy **Management > Forensics > Ignored Items** tab to unset the ignore decision. For more details, see **Forensics**, on page 6-45.

Illegal method

Illegal method				Accept	Clear
<input type="checkbox"/>	Method Name	Occurrences	Act As Method	Check trusted IPs for allowed methods	
<input type="checkbox"/>	COPY	1	POST	<input type="checkbox"/>	
<input type="checkbox"/>	SEARCH	1	GET	<input checked="" type="checkbox"/>	

Checkboxes

The first column contains checkboxes used to mark the relevant entry

Method Name

Describes the Method name

Occurrences

Displays the number of illegal methods occurrences detected.

- If you click the linked occurrence number, a **View requested objects** window appears containing a list of all the objects that caused this violation.
- If you click an object link, the **View full request information** window appears showing all the technical details of all the violations related to the specific request.

Act as Method

For each Method, set a corresponding GET or POST option.

Check Trusted IPs for Allowed Methods

Selecting this checkbox instructs the TS Security Mechanism to check for Trusted IP numbers that are allowed to use the method.

Length violations

Length violations are detected as Length Errors.

Length Violations	
Learning of	Occurrences
<input type="checkbox"/> Length Errors	3

This section lists the requests that exceeded a length setting.

This section is divided into two categories:

- Object Type Length Errors
- Header Length errors

◆ Note

*When you accept an object type, the length of the object is added as part of this object's properties. You can view the added length for an object under **Policy Management > Object types**. Whenever TrafficShield security application identifies a longer length for the same object, it signals a violation, and the new detected length appears under Length Errors in the Learning tool.*

Object type lengths errors

This section lists the requests that exceeded a length setting.

Object Type Length Errors				
<input type="checkbox"/> Object Type	Total Request Length Occurrences	URI Length Occurrences	Query-String Length Occurrences	POST-Data Length Occurrences
<input type="checkbox"/> gif	19	7	0	0
<input type="checkbox"/> no_ext	2	1	0	0
<input type="checkbox"/> php	2	0	3	2

Checkboxes

The first column contains checkboxes used to mark the relevant entry.

Object Type

Select the checkbox for the relevant Object (file) types that you want to clear. If you want to define and accept this object type length, you will need to click the relevant Object type link, and the Requests Lengths for “object” type window will be displayed. For more details, see *Actions available for accept requests lengths*, on page 6-20.

Total Request Length Occurrences

The Total Request Length is the sum of the URI, Query string and POST data lengths in a specific request.

URI Length Occurrences

The maximum URI length received from all the requests for this object type.

Query-String Length Occurrences

The maximum Query string length received from all the requests for this object type.

POST-Data Length Occurrences

The maximum Post data length received from all the requests for this object type.

Available actions for object type length errors

Clear

Deletes the checked entries from this learning window without changing the policy. A confirmation window is displayed.

Clear All

Deletes all entries from this learning window without changing the policy, regardless of whether their checkbox is selected or not. A confirmation window is displayed.

Object Type link

Clicking the **Object Type** link displays the object type length window.

«php» object type lengths Accept All

Length Type	Current Max Length	Detected Max Length	Detected Average Length	Occurrences	Accept
Total Request Length	1075	2764	2764.0	1	<input type="text" value="3593"/> <input type="button" value="Accept"/>
URI Length	14	16	15.5	2	<input type="text" value="20"/> <input type="button" value="Accept"/>
Query-String Length	5	67	22.2	6	<input type="text" value="87"/> <input type="button" value="Accept"/>
POST-Data Length	0	2016	2016.0	1	<input type="text" value="2620"/> <input type="button" value="Accept"/>

Checkboxes

The first column contains checkboxes used to mark the relevant entry.

Length Type

There are four length types. The Total Request Length is the sum of the other three types.

Current Max Length

The length set in the policy. For example, the Current Max Length (column) for URI Length (row) indicates the valid length defined in the policy for the URI section of the request.

Detected Max Length

This value indicates the highest length value that has been detected for a specific policy object.

Detected Average Length

This value indicates the average length value that has been detected for a specific length object. If the average length is very different from the Max length, this could indicate a problem that requires further investigation.

Occurrences

This is the number of requests that have been rejected for violating the length constraints.

Clicking the number opens the Full Request Information window that contains all the technical details of all the violations related to the longest request.

User Input

User Input fields allow the user to enter a valid value that overrides the defaults.

Actions available for accept requests lengths

Accept

Choose the **Accept** button on the relevant length type row if you decide that the returned statistics reflect a real-life situation that warrants a change in the policy. You can also decide to manually define the new length in the user input field in the Accept column. The decision should be based on an in-depth understanding of your Web application.

Accept All

Choose the **Accept All** button if you decide that all the length types displayed reflect a real-life situation that warrants a change in the policy. You can also decide to manually define all new lengths in the user input fields in the Accept column. The decision should be based on an in-depth understanding of your Web application.

To return to the Real Traffic tab

Click the arrow button at the top left corner.

Header length errors

Header Length Errors						Accept	Clear
<input type="checkbox"/> Header Type	Current Max Length	Detected Max Length	Detected Average Length	Occurrences	Set Max Length Value		
<input type="checkbox"/> Cookie Header	1	270	267.2	72	<input type="radio"/> Any	<input checked="" type="radio"/> Length: <input type="text" value="351"/>	
<input type="checkbox"/> HTTP Header	1	278	63.4	791	<input type="radio"/> Any	<input checked="" type="radio"/> Length: <input type="text" value="361"/>	

There are two possible Header length violations:

- HTTP Header
- Cookie Header

Checkboxes

The first column contains checkboxes used to mark the relevant entry.

Header Type

Check the checkbox for the relevant Header type that you want to clear or accept.

Current Max Length

The valid length defined in the policy for the Header length.

Detected Average Length

This value indicates the average Header length that violated the Header length constraint.

Occurrences

This number displays the number of requests that caused this violation.

Set Max Length Value

You can manually change the maximum length allowed for the Header Type or select the Any option to allow any length.

Actions available for Header Length

Accept

Clicking the **Accept** button adds the changes to the policy. Accept means that you have decided that the request reflects a real-life situation that warrants a change in the policy. It is possible to manually change the value of some of the parameters.

Clear

Clicking **Clear** deletes the entry from this learning window without changing the policy. A warning message appears asking to confirm the deletion.

Input violations

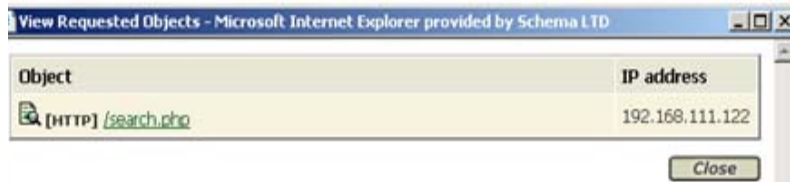
Input violations are classified by the TrafficShield Security Module as follows:

- Illegal Query-String or POST-Data
- Illegal Parameter
- Illegal static parameter value
- Illegal empty parameter value
- Illegal parameter value length
- Illegal parameter numeric value
- Illegal parameter data type
- Illegal meta character in parameter value
- Malicious parameter value

Input Violations	
Learning of	Occurrences
<input type="checkbox"/> Illegal Query-String or POST-Data	1
<input type="checkbox"/> Illegal Parameter	0
<input type="checkbox"/> Illegal static parameter value	1
<input type="checkbox"/> Illegal empty parameter value	1
<input type="checkbox"/> Illegal parameter value length	1
<input type="checkbox"/> Illegal parameter numeric value	1
<input type="checkbox"/> Illegal parameter data type	1
<input type="checkbox"/> Illegal meta character in parameter value	1
<input type="checkbox"/> Malicious parameter value	1

The Occurrences row displays the number of illegal flow to object violation occurrences for each violation type.

- If you click the linked occurrence number, a **View requested objects** window appears containing a list of all the objects that caused this violation. (See the graphic below.)
- If you click an object **link**, the **View full request information** window appears showing all the technical details of all the violations related to the specific request.



Illegal query-string or POST-data

Illegal Query-String or POST-Data		Accept	Clear
<input type="checkbox"/> Flow		Check QS/PD	Occurrences
<input type="checkbox"/>	[HTTP] /index.php → (GET) → [HTTP] /search.php	<input checked="" type="checkbox"/>	3
<input type="checkbox"/>	[HTTP] /index.php → (GET) → [HTTP] /help.php	<input checked="" type="checkbox"/>	1

Checkboxes

The first column contains checkboxes used to mark the relevant entry.

Flow

This is the name of the Application Flow (path) which defines the access path leading from one object to another object.

Check QS/PD

Check this box to instruct the TrafficShield security application to perform validity checks on the query string or POST data.

Occurrences

This number displays the number of requests that caused this violation.

Available actions for illegal query-string or POST-data

Accept

Clicking the **Accept** button adds the changes to the policy. Accept means that you have decided that the request reflects a real-life situation that warrants a change in the policy. It is possible to manually change the value of some of the parameters.

Clear

Clicking **Clear** deletes the entry from this learning window without changing the policy. A warning message appears asking to confirm the deletion.

Illegal parameter

Illegal Parameter				Clear	Clear All
<input type="checkbox"/> Flow	Parameter Name	Occurrences (Values) Number	Accept		
<input type="checkbox"/>	[HTTP] /index.php → (GET) → [HTTP] /search.php	city	1(1)	<input checked="" type="checkbox"/>	Accept

The Illegal Parameter window lists the parameters that can appear in the request but are not defined for a specific flow.

Checkboxes

The first column contains checkboxes used to mark the relevant entry.

Flow

This is the name of the Application Flow (path) which defines the access path leading from one object to another object.

Parameter Name

This is the name of the undefined parameter.

Occurrences (Values) Number

This field displays the number of illegal parameter occurrences.

- If you click the linked occurrence number, a **View requested objects** window appears containing a list of all the objects that caused this violation.
- If you click an object link, the **View full request information** window appears showing all the technical details of all the violations related to the specific request.

Available actions for Illegal parameter

Clear

Click Clear to clear the specific entry from the learning window without changing the policy

Accept

To accept the violation case and make it legal for future occurrences, click Accept. The Accept Parameter window appears.

Illegal static parameter value

This screen shows static parameters that carried a value not included in the value list defined in the policy.

Illegal static parameter value (1)			Clear	Clear All
<input type="checkbox"/> Parameter Name	Parameter Flow	Occurrences (Values) Number		
<input type="checkbox"/> id	[HTTP] /index.php → (GET) → [HTTP] /browse.php	2(2)		

Checkboxes

The first column contains checkboxes used to mark the relevant entry.

Parameter Name

This is the name of the illegal static parameter value.

Parameter Flow

This is the name of the parameter flow (path) which defines the access path leading from one object to another object.

Occurrences (Values) Number

This field displays the number of illegal flow to object violation occurrences.

- If you click the linked occurrence number, a **View requested objects** window appears containing a list of all the objects that caused this violation.
- If you click an object link, the **View full request information** window appears showing all the technical details of all the violations related to the specific request.

Available actions for illegal static parameter value

Clear

Click Clear to clear the specific entry from the learning window without changing the policy.

Clear All

Click Clear All to clear all the entries from the learning window without changing the policy.



Click the parameter name link to accept the violation, this will open a new window.

Illegal empty parameter value

This window displays the list of parameters that violated the not null value definition. (The field was empty when it should have contained a value.)

The decision whether a specific parameter can be left empty or not is dependent on the web application.

Illegal empty parameter value (2) Accept Clear Clear All

<input type="checkbox"/> Parameter Name	Parameter Flow	Occurrences (Values) Number
<input type="checkbox"/> UNNAMED	 [HTTP] /index.php → (GET) →  [HTTP] /search.php	2(1)
<input type="checkbox"/> q	 [HTTP] /index.php → (GET) →  [HTTP] /search.php	3(1)

Checkboxes

The first column contains checkboxes used to mark the relevant entry.

Parameter Name

Lists the parameters where Value Error was found.

Parameter Flow

This is the name of the parameter flow (path) which defines the access path leading from one object to another object.

Occurrences (Values) Number

This number displays the number of requests and values that caused this violation.

- If you click the linked occurrence number, a **View requested objects** window appears containing a list of all the objects that caused this violation.
- If you click an object link, the **View full request information** window appears showing all the technical details of all the violations related to the specific request.

Available actions for illegal empty parameter value

Clear

Click **Clear** to clear the specific entry from the learning window without changing the policy.

Clear All

Click **Clear All** to clear all the entries from the learning window without changing the policy.

Illegal parameter value length

◆ **Note**

This violation is relevant only for the Parameter Type: User Input Value.

Illegal parameter value length (2)				Clear	Clear All
<input type="checkbox"/>	Parameter Name	Curr. Max Value Length	Detected Max Value Length	Occurrences (Values) Number	Accept
<input type="checkbox"/>	id	3	8	1(1)	8 <input type="text"/> <input type="button" value="Accept"/>
<input type="checkbox"/>	qi	4	6	1(1)	6 <input type="text"/> <input type="button" value="Accept"/>

Checkboxes

The first column contains checkboxes used to mark the relevant entry.

Parameter Name

Lists the parameters where the illegal parameter Value length error occurred.

Current Max Value Length

The maximum length value permitted for this parameter.

Detected Max Value Length

The maximum length value permitted for this parameter

Occurrences (Values) Number

This number displays the number of requests and values that caused this violation.

- If you click the linked occurrence number, a **View requested objects** window appears containing a list of all the objects that caused this violation.
- If you click an object link, the **View full request information** window appears showing all the technical details of all the violations related to the specific request.

User Input

User Input fields allow the user to enter a valid value that overrides the defaults.

Available actions for illegal parameter value length

Accept

To accept the violation case and make it legal for future occurrences.

Clear

To clear the specific entry/entries from this learning window without changing the policy.

Clear All

To clear all entries from this learning window without changing the policy, regardless of whether they are checked or not. You will be asked to confirm the operation.

Illegal parameter numeric value

The Illegal Parameter Numeric Value screen lists the errors that may occur for the request parameters' values if the defined parameter is decimal or integer in the policy. This window provides statistical information regarding the types of parameter numeric value problems that have been detected.

Illegal parameter numeric value (1)							Clear	Clear All	
<input type="checkbox"/>	Parameter Name	Current		Detected		Occurrences (Values) Number	Min	Max	Accept
		Min	Max	Min	Max				
<input type="checkbox"/>	q	-5	123	-8	234	2(2)	-8	234	Accept

Checkboxes

The first column contains checkboxes used to mark the relevant entry.

Parameter Name

This is the name of the illegal parameter value length.

Current Min Max

The minimum and maximum numeric values permitted for this parameter.

Detected Min Max

The detected minimum and maximum numeric values detected in the violation.

Occurrences (Values) Number

This number displays the number of requests and values that caused this violation.

- If you click the linked occurrence number, a **View requested objects** window appears containing a list of all the objects that caused this violation.
- If you click an object link, the **View full request information** window appears showing all the technical details of all the violations related to the specific request.

User Input - Min

This is the minimum value received from all the request parameters with this violation type. It is possible to manually change the value of this field.

User Input - Max

This is the maximum value received from all the request parameters with this violation type. It is possible to manually change the value of this field.

Available actions for illegal parameter numeric value

Accept

To accept the violation case and make it legal for future occurrences.

Clear

To clear the specific entry/entries from this learning window without changing the policy.

Clear All

To clear all entries from this learning window without changing the policy, regardless of whether they are checked or not. You will be asked to confirm the operation.

Illegal parameter data type

This screen shows parameters whose data type is different from the data type defined for them in the policy.

Illegal parameter data type (1) [Clear](#) [Clear All](#)

<input type="checkbox"/> Parameter Name	Parameter Flow	Occurrences (Values) Number	Data Type in Policy	Accept
<input type="checkbox"/> q	 [HTTP] /index.php → (GET) →  [HTTP] /search.php	8(8)	Alpha-Numeric English	Accept

Checkboxes

The first column contains checkboxes used to mark the relevant entry.

Parameter Name

Lists the parameters where the illegal parameter data length error occurred.

Parameter Flow

This is the flow where the parameter value error occurred.

Occurrences (Values) Number

This number displays the number of requests and values that caused this violation.

- If you click the linked occurrence number, a **View requested objects** window appears containing a list of all the objects that caused this violation.
- If you click an object link, the **View full request information** window appears showing all the technical details of all the violations related to the specific request.

Data Type in Policy

This field displays the data type that was detected by the value error.

Available actions for illegal parameter data type

Accept

To accept the violation case and make it legal for future occurrences.

Clear

To clear the specific entry/entries from this learning window without changing the policy.

Clear All

To clear all entries from this learning window without changing the policy, regardless of whether they are checked or not. You will be asked to confirm the operation.

Illegal meta character in parameter value

The parameter name contains a character that is set to “N” (false) or “C” (check) in the **Administration > Character Sets > User Input: Defaults**.

Illegal meta character in parameter value (1)			Clear	Clear All
<input type="checkbox"/> Parameter Name	Parameter Flow	Occurrences (Values) Number		
<input type="checkbox"/> id	[HTTP] /index.php → (GET) → [HTTP] /browse.php	3(3)		

Checkboxes

The first column contains checkboxes used to mark the relevant entry.

Parameter Name

Lists the parameters where the illegal character in parameter value error occurred.

Parameter Flow

This is the flow where the parameter value error occurred.

Occurrences (Values) Number

This number displays the number of requests and values that caused this violation.

- If you click the linked occurrence number, a **View requested objects** window appears containing a list of all the objects that caused this violation.

- If you click an object link, the **View full request information** window appears showing all the technical details of all the violations related to the specific request.

Available actions for illegal meta character in parameter value

Accept

To accept the violation case and make it legal for future occurrences.

Clear

To clear the specific entry/entries from this learning window without changing the policy.

Clear All

To delete all entries from this learning window without changing the policy, regardless of whether they are checked or not. You will be asked to confirm this.

Click the **Parameter name** to open the following screen.



The color legend at the top of the screen is applicable all through the Security Policy tool.

- Red - blocks the character.
- Blue - flags the character.
- Black - allows the character.

Click the parameter name link to open the **Edit Parameter window**.

Edit Parameter: id Global Update Save

Parameter Name: id Is Mandatory Parameter

Parameter Type: User-input value Allow Empty Value
Input Type: text-input

Parameter Characteristics

Data Type: Alpha-Numeric Hebrew

Check Minimum Value: Check Maximum Value:

Check Maximum Length: 3 Regular Expression:

Allowed Meta Characters:	Allowed Regular Expressions:
<input type="checkbox"/> ; (0x3b)	<input type="checkbox"/> .*(?<).*SCRIPT.*>.*
<input type="checkbox"/> (0x7c)	<input type="checkbox"/> .*(?<).*SELECT.*FROM.*
<input type="checkbox"/> ! (0x21)	<input type="checkbox"/> .*(?<).*exec.*xp_.*
<input type="checkbox"/> & (0x26)	<input type="checkbox"/> .*(?<).*exec.*.dbo.*
<input type="checkbox"/> Space (0x20)	<input type="checkbox"/> .*(?<).*sys.*
<input type="checkbox"/> EOT (0x04)	<input type="checkbox"/> .*(?<).*DBCC.*
<input type="checkbox"/> LF (0x0a)	<input type="checkbox"/> .*(?<).*OR.*1=1.*
<input type="checkbox"/> CR (0x0d)	<input type="checkbox"/> .*(?<).*OR.*'1'='1'.*
<input type="checkbox"/> ESC (0x1b)	<input type="checkbox"/> (?<)%(?<)
<input type="checkbox"/> BS (0x08)	<input type="checkbox"/> 2c 26 27 22 2b 2d 26 20 25 2f 21 3f 28 29 40 3a)
<input type="checkbox"/> DEL (0x7f)	<input type="checkbox"/> [0-9a-f]{2}
	<input type="checkbox"/> (?<).*META.*>.*

Edit the parameter as required.

Malicious parameter value

This is the parameter that contains a regular expression value that is not defined as an allowed regular expression for this parameter.

Malicious parameter value (1) Clear Clear All

<input type="checkbox"/> Parameter Name	Parameter Flow	Occurrences (Values) Number
<input checked="" type="checkbox"/> a	Entry Point → (GET) → [HTTP] /search.php	3(2)

Checkboxes

The first column contains checkboxes used to mark the relevant entry

Parameter Name

Lists the parameters where the malicious parameter value error occurred.

Parameter Flow

This is the flow where the parameter value error occurred.

Occurrences (Values) Number

This number displays the number of requests and values that caused this violation.

- If you click the linked occurrence number, a **View requested objects** window appears containing a list of all the objects that caused this violation.
- If you click an object link, the **View full request information** window appears showing all the technical details of all the violations related to the specific request.

Available actions for malicious parameter value

Clear

Click **Clear** to clear the specific entry/entries from this learning window without changing the policy.

Clear All

Click **Clear All** to delete all entries from this Learning window without changing the policy, regardless of whether they are checked or not. You will be asked to confirm this.

To edit the parameter definitions

Click the parameter name link, and the Values for the specific parameter window are displayed.

Parameter Value	Occurrences
<input type="checkbox"/> Parameter Value	
<input type="checkbox"/> <meta>	1
<input type="checkbox"/> <script>	1

Parameter Value

The parameter value where the violation occurred

Occurrences

This number displays the number of requests that caused this violation.

Available actions for editing the malicious value parameter definition

Accept

To accept the violation case and make it legal for future occurrences.

◆ WARNING

Be aware that accepting this violation instructs TrafficShield system to allow values whose contents might be, in certain cases, attacks.

Clear

To clear the specific entry/entries from this learning window without changing the policy.

Clear All

To clear all entries from this learning window without changing the policy, regardless of whether they are checked or not. You will be asked to confirm the operation.

To open the Edit parameter window, click the specific parameter name.

Negative security violations

Negative Security Violations are generated whenever a character or regular expression that is not allowed in the TrafficShield security application policy is detected.

Negative Security Violations	
Learning of	Occurrences
<input type="checkbox"/> Illegal meta character in header	2
<input type="checkbox"/> Illegal meta character in object	2
<input type="checkbox"/> Illegal meta character in parameter name	2
<input type="checkbox"/> Illegal meta character in parameter value	2
<input type="checkbox"/> Illegal pattern in object	0
<input type="checkbox"/> Illegal pattern in response	0
<input type="checkbox"/> Illegal pattern in header	0
<input type="checkbox"/> Illegal pattern in user input	0

The Negative Security Violations are classified as follows:

- Illegal meta character in header
- Illegal meta character in object
- Illegal meta character in parameter name
- Illegal meta character in parameter value
- Illegal pattern in object
- Illegal pattern in response
- Illegal pattern in header
- Illegal pattern in user input

Illegal meta character in header

This violation is detected whenever a meta character is detected in the Header.

The list of legal meta characters can be found in the Character Set tab of the configuration section.

Accepting a header that contains an illegal meta character or more modifies the Action for all the illegal meta characters found in the header from **No** to **Yes** in the Configuration > Character Sets > Header Charset list in the Policy Management tool.

Illegal meta character in header		Accept	Clear
<input type="checkbox"/> Header	Metachars	Occurrences	
<input type="checkbox"/> Test		1	
<input type="checkbox"/> Length'	'	1	

Checkboxes

The first column contains checkboxes used to mark the relevant entry.

Header

The Header name that has been detected as containing a meta character.

Metachars

The illegal meta characters that were detected.

◆ Note

Once TrafficShield enforcer detects the first meta character, it does not continue to check the rest of the value.

Occurrences

This number displays the number of requests and values that caused this violation.

- If you click the linked occurrence number, a **View requested objects** window appears containing a list of all the objects that caused this violation.
- If you click an object link, the **View full request information** window appears showing all the technical details of all the violations related to the specific request.

Available actions for illegal meta character in header

Accept

To accept the violation case and make it legal for future occurrences.

◆ Note

Accepting a meta character does not eliminate from the Learning tool other Learning records that include this meta character. In such cases, you must clear those other Learning records as well.

Clear

To clear the specific entry/entries from this learning window without changing the policy.

Illegal meta character in object

This violation is detected whenever a meta character is detected in the Object.

Illegal meta character in object		Accept	Clear
<input type="checkbox"/>	Object		Occurrences
<input type="checkbox"/>	/index^\.php		1
<input type="checkbox"/>	/register.php		2

The list of legal meta characters can be found in the **Character Set** tab of the **Configuration** section.

Accepting an object that contains an illegal meta character modifies the Action for all the illegal meta characters found in the header from **NO** to **Yes** in the Configuration > Character Sets > Object Charset list in the Policy Management tool.

Checkboxes

The first column contains checkboxes used to mark the relevant entry.

Object

The object in which the illegal meta character was detected.

Occurrences

The number of occurrences of the violation.

Available actions for Illegal meta character in object

Accept

To accept the violation case and make it legal for future occurrences.

Clear

To clear the specific entry/entries from this learning window without changing the policy.

Illegal meta character in parameter name

Illegal meta character in parameter name		Accept	Clear
<input type="checkbox"/>	Parameter Name		Occurrences
<input type="checkbox"/>	param&V		1
<input type="checkbox"/>	index\$		1

This violation is detected whenever a meta character is detected in the Parameter name.

Accepting a parameter that contains an illegal meta character modifies the Action for all the illegal meta characters found in the header from **NO** to **Yes** in the Configuration **Character Sets > Parameter name list** in the Policy Management tool.

Checkboxes

The first column contains checkboxes used to mark the relevant entry.

Parameter Name

The name of the parameter in which the illegal meta character was detected.

Occurrences

The number of occurrences of the violation.

Available actions for illegal meta character in object

Accept

To accept the violation case and make it legal for future occurrences.

Clear

To clear the specific entry/entries from this learning window without changing the policy.

Illegal meta character in parameter value

This violation is detected whenever a meta character is detected in the parameter value.

Illegal meta character in parameter value		Accept	Clear
<input type="checkbox"/> Parameter Value		Occurrences	
<input type="checkbox"/>	name_car=Jafuar/Jeep	2	
<input type="checkbox"/>	city=~London	1	

Checkboxes

The first column contains checkboxes used to mark the relevant entry.

Parameter Value

The parameter value in which the error value occurred.

Available actions for illegal meta character in parameter value

Accept

To accept the violation case and make it legal for future occurrences.

Clear

To clear the specific entry/entries from this learning window without changing the policy.

Illegal pattern in object

This violation is displayed whenever an illegal pattern is detected in the Object.

Accepting an object that contains an illegal pattern deletes the regular expression (the pattern) detected in that object from the list of regular expressions defined for checking objects in the **Configuration > Negative RegExp Default** screen in the Policy Management section.

Illegal pattern in object		Accept	Clear
<input type="checkbox"/> Object		Occurrences	
<input type="checkbox"/>	/../\W\..metaindex.php		1

Checkbox

The first column contains checkboxes used to mark the relevant entry

Object

The name of the object in which the illegal pattern occurred.

Occurrences

The number of occurrences of the violation.

Available actions for illegal pattern in object

Accept

To accept the violation case and make it legal for future occurrences. For example, the regular expression that generates the violation will not be checked on objects for this policy.

Clear

To clear the specific entry/entries from this learning window without changing the policy.

Illegal pattern in response

This violation is signaled whenever an illegal pattern is detected in the Response.

Accept
Clear

Response	RegExp Name	Occurrences
<input type="checkbox"/> <pre><? /* <php> dddd </php> Copyright (c), 1999, # 2000 - phpauction.org This program is free # software; you can redistribute it and/or modify it under the # terms of the GNU General Public License as published by the Free # Software Foundation (version 2 or later). # # This program is distributed in the hope that it will be # useful, but WITHOUT ANY WARRANTY; without even the implied # warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR # PURPOSE. See the GNU General Public License for more # details. # You should have received a # copy of the GNU General Public License along with this # program; if not, write to the Free Software Foundation, # Inc., 675 Mass Ave, Cambridge, MA 02139, USA. # */ require("../includes/config.inc.php"); require("../includ # es/messages.inc.php"); require("../header.php"); # /* If subscription to information service # requested update INFO table */ \$TPL_info_err = # ""; if(\$user_email){ if(!ereg("^[_a-z0-9-]+(\.[_a- # -z0-9-]+)*@[a-z0-9-]+(\.[_a-z0-9-]+)+\$", \$user_email)){ \$TPL_ # info_err = \$ERR_026; }else{ \$query = "insert into info # values('\$user_email')"; \$result = # mysql_query(\$query); if(!\$result){ \$TPL_info_err = # \$ERR_001; }else{ \$TPL_info_err = # \$MSG_512; } } } /* prepare data for # templates/template */ /* prepare categories list for # templates/template */ \$TPL_categories_value = # ""; \$query = "select * from categories WHERE # parent_id=0 order by sub_counter desc"; \$result = # mysql_query(\$query); if(!\$result) { \$TPL_categories_value # .= \$ERR_001; } else { \$num_cat = # mysql_num_rows(\$result); \$i = 0; \$TPL_categories_value .= # "<</pre>	Server side code disclosure 1	1

The list of legal patterns for the response can be found in the **Configuration > Negative RegExp** default list referring to responses in the Configuration section of the Policy Management tool.

Accepting a response that contains an illegal pattern deletes the regular expression found for the response from the list in the **Configuration > Negative RegExp default** screen referring to responses in the Policy Management section.

Illegal pattern in header

This violation is signaled whenever an illegal pattern is detected in the Header.

Accept
Clear

Header value	RegExp Name	Occurrences
<input type="checkbox"/> <pre>Referer:http://phpauction.core.com/help.php?<script>=cross-site- # scripting</pre>	Cross-Site Scripting 3	3

The list of legal patterns that is used to check the Header request in the **Configuration > Negative RegExp list** in the **Configuration** section of the **Policy Management** section.

Accepting a header that contains an illegal pattern deletes the regular expression found in the header from the list for the Configuration > Negative RegExp default list in the Policy Management tool.

Checkboxes

The first column contains checkboxes used to mark the relevant entry

Object

The name of the header in which the illegal pattern was detected.

Occurrences

The number of occurrences of the violation.

Available actions for illegal pattern in header

Accept

To accept the violation case and make it legal for future occurrences.

Clear

To clear the specific entry/entries from this learning window without changing the policy.

Illegal pattern in user input

This violation is signaled whenever an illegal pattern is detected in the user input.

Illegal pattern in User input			Accept	Clear
<input type="checkbox"/>	User input	RegExp Name	Occurrences	
<input type="checkbox"/>	<script>=cross-site-scripting	Cross-Site Scripting 1	1	

The list of legal patterns that is used to check the user input request is in the **Configuration > Negative RegExp** under **Configuration** in the Policy Management section.

Accepting a user input that contains an illegal pattern deletes the regular expression found in the user input from the list for the **Configuration > Negative RegExp** default list in the Policy Management section.

Checkboxes

The first column contains checkboxes used to mark the relevant entry.

User Input

The parameter key and value pair in which the illegal pattern was detected.

Occurrences

The number of occurrences of the violation.

Available actions for Illegal pattern in user input

Accept

To accept the violation case and make it legal for future occurrences.

Clear

To clear the specific entry/entries from this learning window without changing the policy.

Cookie violations

Cookie Violations	
Learning of	Occurrences
<input type="checkbox"/> Objects That Modified Domain Cookies	2

This category contains one section:

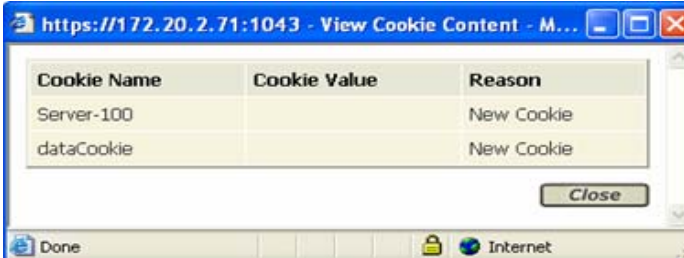
- [Objects that modified domain cookies.](#)

Modified domain cookies

This violation category is divided into two cookie violation sub-categories: **Modified Domain Cookies** and **Objects That Modified Domain Cookies**.

Modified Domain Cookies		<input type="button" value="Accept"/>	<input type="button" value="Clear"/>
<input type="checkbox"/> Cookie Name		Occurrences	
<input type="checkbox"/> PHPAUCTION_SESSION		1	
<input type="checkbox"/> Server-100		2	

Click the cookie name. The cookie content is displayed.



Cookie Name	Cookie Value	Reason
Server-100		New Cookie
dataCookie		New Cookie

Close

Checkboxes

The first column contains checkboxes used to mark the relevant entry.

Cookie Name

This is the attribute part of the cookie name value pair (name=value).

Occurrences

The number of occurrences of the violation.

Available actions for Modified domain cookies

Accept

To accept the violation case and make it legal for future occurrences.

Clear

To clear the specific entry/entries from this learning window without changing the policy.

Objects that modified domain cookies

This screen lists the objects that modified domain cookies.

Objects That Modified Domain Cookies		Accept	Clear
<input type="checkbox"/>	Object	Occurrences	
<input type="checkbox"/>	[HTTP] /index.php	1	
<input type="checkbox"/>	[HTTP] /search.php	2	

Checkboxes

The first column contains checkboxes used to mark the relevant entry.

Object

This is the name of the Object that modified the Domain Cookie.

Occurrences

This is the number of times that the object modified the Domain Cookie.

Available actions for Objects that modified domain cookies

Accept

To accept the violation case and make it legal for future occurrences.

Clear

To clear the specific entry/entries from this learning window without changing the policy.

Object link

Click the Object link to view the cookie contents.

Forensics

This section explains how the user can review all the requests that caused at least one violation error. Each request is mapped to the learning tables, and the user can locate the full content of the specific request in order to do further investigation and to have a better understanding of the problem.

All the requests that violate the policy settings always go to the Illegal Requests table in the Forensics section. The other Forensic tables store requests that meet specific criteria (i.e., defined as containing components that were defined by the user as illegal).

You can select multiple forensic entries using the Forensic filters tool located at the top of all the Forensics windows.



Illegal requests

You can view requests that contradict the policy in the Illegal Requests window. In addition, these requests are automatically categorized according to their content and registered in the appropriate Learning tables as well.

For example, a request for an illegal flow is registered in **Forensics - Illegal Requests** and also in **Learning - Undefined Flows**.

	Time	Type	Requested Object	Response	Source IP
<input type="checkbox"/>	2004-12-08 18:58:57	HTTP	/index.php	200	192.168.111.122
<input type="checkbox"/>	2004-12-08 18:58:55	HTTP	/search.php	200	192.168.111.122
<input type="checkbox"/>	2004-12-08 18:58:52	HTTP	/index.php	200	192.168.111.122
<input type="checkbox"/>	2004-12-08 18:58:45	HTTP	/help.php	200	192.168.111.122
<input type="checkbox"/>	2004-12-08 18:58:45	HTTP	/user_login.php	200	192.168.111.122
<input type="checkbox"/>	2004-12-08 18:58:43	HTTP	/bid.php	200	192.168.111.122
<input type="checkbox"/>	2004-12-08 18:58:35	HTTP	/register.php	200	192.168.111.122
<input type="checkbox"/>	2004-12-08 18:58:34	HTTP	/browse.php	200	192.168.111.122
<input type="checkbox"/>	2004-12-08 18:58:31	HTTP	/sell.php	200	192.168.111.122
<input type="checkbox"/>	2004-12-08 18:58:30	HTTP	/	200	192.168.111.122

4 Pages: [1] 2 3 4 =

Checkboxes

The first column contains checkboxes used to mark the relevant entry.

Blocked column

The second column may contain a red X which indicates that this request was blocked.

Time

This box displays the date and time of the request.

Type

This shows the protocol of the request (HTTP/HTTPS).

Requested Object

This field displays the requested URI.

Response

This field is the server HTTP response status.

Source IP

This is the IP address of the client machine that issued the request.

◆ Note

Click a specific object to view the full contents of the request, and the View Full Request Information window is displayed.

Available actions for requests in Forensics

Clear

Clicking **Clear** deletes the checked entries from this window without changing the policy. A confirmation window is displayed.

◆ Note

When a request is deleted from Forensics, then it is also deleted from the Learning tables.

Clear All

Clicking **Clear All** deletes all entries from this learning window without changing the policy, regardless of whether they are checked or not. You will be asked to confirm this.

Support ID

This is the unique identifier of the illegal request.

Cookie Name

This is the attribute part of the cookie name value pair (name=value).

Cookie Value

This is the value part of the cookie name value pair (name=value).

Reason

This is the reason that caused the violation.

Ignored requests

This section deals with illegal requests that match the ignored criteria defined by the user (for instance, in case the request's object type and/or the the request's object and/or the request's flow are defined by the user explicitly as illegal). As a result, they are not used for Learning and are displayed as ignored.

<input type="checkbox"/>	Time	Type	Requested Object	Response	Source IP
<input type="checkbox"/>	2004-12-06 11:18:06	HTTP	/uploaded/logo_of	304	192.168.111.122
<input type="checkbox"/>	2004-12-06 11:17:52	HTTP	/uploaded/logo_of	304	192.168.111.122
<input type="checkbox"/>	2004-12-06 11:17:20	HTTP	/uploaded/logo_of	304	192.168.111.122
<input type="checkbox"/>	2004-12-06 11:17:18	HTTP	/uploaded/logo_of	304	192.168.111.122
<input type="checkbox"/>	2004-12-06 11:17:15	HTTP	/uploaded/logo_of	304	192.168.111.122
<input type="checkbox"/>	2004-12-06 11:17:14	HTTP	/uploaded/logo_of	304	192.168.111.122
<input type="checkbox"/>	2004-12-06 11:11:11	HTTP	/images/info_of	200	192.168.111.122
<input type="checkbox"/>	2004-12-06 11:11:11	HTTP	/images/estrella_3_of	200	192.168.111.122
<input type="checkbox"/>	2004-12-06 11:11:11	HTTP	/uploaded/f69d3cc26844d4fae02dbb6af5a53254_of	200	192.168.111.122
<input type="checkbox"/>	2004-12-06 11:09:03	HTTP	/forgotpassword.php	200	192.168.111.122

8 Pages: [First](#) ... [4](#) [5](#) [6](#) [7](#) **[8]**

Checkboxes

The first column contains checkboxes used to mark the relevant entry.

Blocked column

The second column may contain a red X which indicates that this request was blocked.

Time

Date and Time of request

Type

Protocol of the request (HTTP/HTTPS)

Requested Object

This field displays the requested URI.

◆ Note

Click a specific object to view the full contents of the request, and the View Full Request Information window is displayed.

Response

This field is the server HTTP response status.

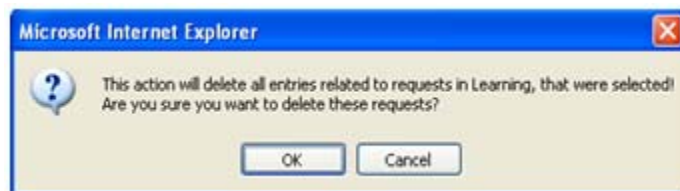
Source IP

This is the IP address of the client machine that issued the request.

Available actions for ignored requests

Clear

Clicking Clear deletes the checked entries from this learning window without changing the policy. The following confirmation window is displayed.

**Clear All**

Clicking Clear All deletes all entries from this learning window without changing the policy, regardless of whether they are checked or not. You will be asked to confirm this.

Viewing the Full Request Information window

You can access this window from both the Learning and the Forensic areas.

Following is an example of the type of information displayed when you choose to open this window.

Ignored items

This section explains the origin of the items listed in the Ignored Items window.

Filter

Ignored Items for Web Application:

Ignored Object Types

Type

gif

Ignored Objects

Object

[HTTP] /..../meta.php

[HTTP] /forgotpasswd.php

Ignored Flows

Flow

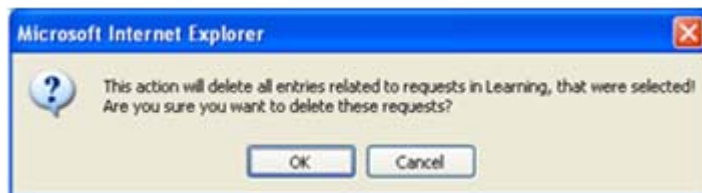
[HTTP] /sel.php → (POST) → [HTTP] /sel.php

[HTTP] /sel.php → (GET) → [HTTP] /sel.php

While using the Learning capabilities to fine-tune the policy object types, objects and flows may be either accepted or cleared. When a user chooses to clear any of the items in the above list, the user is asked whether he would like to “permanently reject the item from learning.” This instructs the TrafficShield security application not to register duplicate identical requests in the Learning tables. The deleted request goes to the Forensics > Ignored Items screen.

To change the permanent ignore decision, select the checkbox next to the relevant item, and click the relevant **Clear** button.

The next time a new request causes a violation it will not be ignored, and will appear in the corresponding Learning windows, and the full request contents will be viewable in the Illegal Requests window.



Available actions for ignored items

Close

Close the window and return to the previous screen.

Accept

To accept the violation case and make it legal for future occurrences.

Run Auto-Accept

Run the **Auto-Accept** function.



Glossary

ARP

Address Request Protocol: (a networking protocol). A method for finding a host's IP address from its Ethernet address. The sender broadcasts an ARP packet containing the IP address of another host and waits for it (or some other host) to send back its Ethernet address. Each host maintains a cache of address translations to reduce delay and loading. ARP allows the IP address to be independent of the Ethernet address, but it only works if all hosts support it.

ARP is defined in RFC 826.

The alternative for hosts that do not do ARP is constant mapping.

Check Object

Indicates whether TrafficShield security application should check the Object requested in the HTTP/HTTPS request against the list of its known objects before it forwards the request to the server. In case it doesn't find the requested object in the list, it generates a violation that, based on the blocking policy, can cause the request to be blocked

Cookie

A packet of information sent by an HTTP server to a World-Wide Web browser and then sent back by the browser each time it accesses that server. Cookies can contain any arbitrary information the server chooses and are used to maintain state between otherwise stateless HTTP transactions. Typically this is used to authenticate or identify a registered user of a Web application without requiring them to sign in again every time they access that Web application. Other uses are maintaining a "shopping basket" of goods you have selected to purchase during a session at a Web application, Web application personalization (presenting different pages to different users), and tracking a particular user's access to a Web application.

DELETE

An HTTP request type that requests to delete a resource on the web server.

Domain Name

A series of alphanumeric strings separated by periods, such as **www.siterequest.com**, that is an address of a computer network connection, and that identifies the owner of the address.

Dynamic Parameter

A dynamic parameter is a parameter in a request where the set of legal values this parameter can have is changing dynamically, and usually depends of the user session. For example, in a banking application the account number is a dynamic parameter, since each user has its own set of legal account numbers that this parameter can have. This set of legal account numbers is dynamically generated by the server and embedded in the web page sent to user. TrafficShield security application extracts this list of legal values from the web page that is sent to the user, and uses them to verify that the value sent in the request for the dynamic parameter is legal.

Dynamic Value

See *dynamic parameter*

Entry Point

A web page that could be the first requested page in the Web application: an end-user could get to the Entry Point by typing a URL in the browser window, opening a favorites menu, be linked from a different Web application or e-mail client. The end user could also get to the Entry Point by clicking a back button of the browser.

Flow

The defined access path for a browser to get from one object to another specific object.

GET

A type of HTTP request that does not have a content body

Learning

A process of making a policy more accurate by verifying how the policy complies with the traffic requests, and if there are discrepancies between the policy and the traffic requests, then translating these discrepancies into a suggestion for modifying the policy. The learning phase also enables the system administrator to verify that the policy is not generating any false positives before turning on the blocking feature. The learning process can be used to fine-tune any policy component such as requests length, parameters, and values. In case new objects are added in the Web application, TrafficShield security application can learn those objects and their flows using the learning engine.

Length-Cookie

The length of the cookie.

Length-Post Data

The length of the Data that comes with a POST request.

Length-Query String

The length of the Query string.

Length-Request

See *Request Length*.

Length-URI

The length of the URI in characters.

Meta character

A character or a sequence of characters that has a special meaning (<SCRIPT >, \, SELECT, INSERT, ;, `, <).

Method

The HTTP/HTTPS request method, e.g. GET, POST, HEAD, PUT, and DELETE.

Non Existent Object

The flow did not match the defined flows.

Object

A file or a script that generates web pages on the web server that can be requested by a user,

Object is Allowed to modify domain Cookie

In case an Object (i.e., a web page) includes a JavaScript/java applet/flash as part of the client-side and can change a domain cookie value, the object should be defined as "Object is allowed to modify Cookie."

Path Traversal

An HTTP Attack that uses patterns like ../../ to get access to files not intended to be viewed above the WWW root, or in order to cross directories on the server.

Policy

A set of rules that enables TrafficShield security application to understand if a request is valid.

POST

A type of HTTP request, in which a query is put into a content body and possibly compressed or encoded.

PUT

An HTTP request type that requests a content change on the web server.

Query String

Part of an HTTP request that specifies a list of parameters and values into a CGI script. For instance:

`http://www.siterequest.com/index.cgi?param1=value1¶m2=value2`

Anything that comes after the question mark in the example above is a query string.

Referrer

A web page that requests other objects An HTML page could request picture files and other html objects to be downloaded, but pictures cannot cause other objects to be downloaded. For example, HTML, asp, php pages are usually Referrers, while gif and jpeg images are not.

Regular Expression

Used by UNIX utilities such as grep, sed and awk, and by editors such as vi and Emacs. A regular expression (regexp) is a sequence of characters which provides the user with a powerful, flexible and efficient test processing tool. For more details on how to write regular expressions please refer to the many books written on this subject; for example: Mastering Regular Expressions, by Jeffrey E.F. Friedl, Published by O'Reilly & Associates, Inc.

Request Length

The total Length of the HTTP request (in characters) which includes the request line, all headers, cookies, and post data.

Server IP

The IP address of the Web Server that TrafficShield security application is protecting (usually this is an internal IP address).

Service IP

The external IP address on which TrafficShield security application is listening for http requests. (Usually this is the IP address that the DNS A record of the Web Server is mapped to.)

Shield Unit

The on-line enforcing mechanism responsible for TCP session termination, requests parsing, and analyzing.

Static Parameter

A parameter in the request where its values are chosen from a known set of values: Name of a Country, Yes/No, etc.

Static Value

See *static parameter*.

Target Frame

The frame to which the object is loaded.

Undefined Flow

The flow did not match the defined flows.

Undefined Object

The object did not match any objects on the list of allowed objects.

URI

Part of the URL that specifies the name of the object requested: in **http://www.siterequest.com/index.html**, **index.html** is the URI.

