SECURITY code

Continent WAF Version 2

User Guide

Mailing address:	**115127, Russian Federation, Moscow, 1st Nagatinsky proezd, 10/1**
Phone:	**+7 (495) 982-30-20**
E-mail:	**info@securitycode.ru**
Web:	**https://www.securitycode.ru**

Table of contents

Introduction

This manual is designed for users of Continent WAF, Version 2 (hereinafter — Continent WAF). It contains information about the installation and configuration of Continent WAF.

This document contains links to the document [**1**].

Website. Information about SECURITY CODE LLC products can be found on https://www.securitycode.ru/.

Technical support. You can contact technical support by phone: 8 800 505-30-20 or by email: support@securitycode.ru.

Training. You can learn more about hardware and software products of SECURITY CODE LLC in authorized education centers. The list of the centers and information about learning environment can be found on https://www.securitycode.ru/company/education/training-courses/. You can contact a company's representative for more information about trainings by email: education@securitycode.ru.

Chapter 1
Overview

Continent WAF features

Continent WAF is a smart firewall designed to protect web applications. Continent WAF ensures the protection of critical web resources from external attacks and makes it possible to monitor web applications according to allowed scenarios.

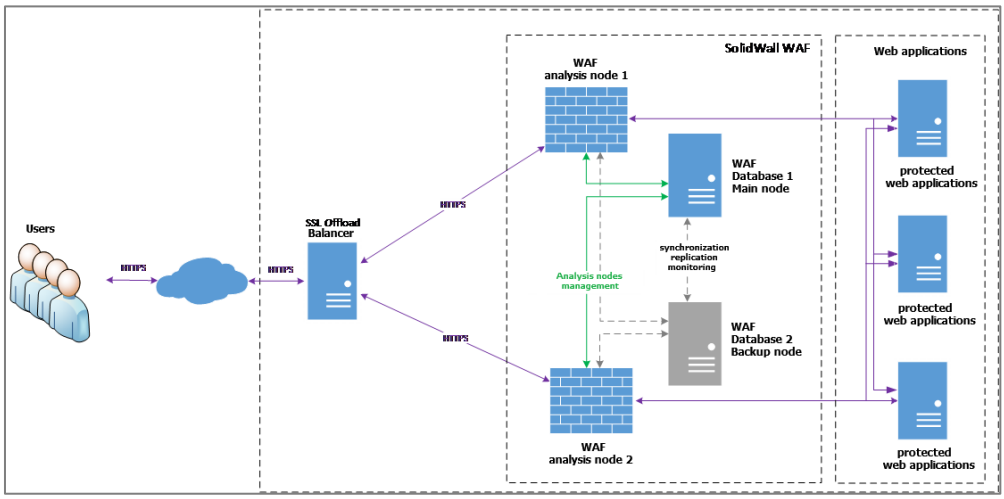
Continent WAF performs the following:

- control and filtering;
- user identification and authentication;
- security event registration (audit);
- continuous operation and recovery;
- testing and integrity control;
- management;
- interoperability with other security tools.

Continent WAF includes:

- analysis modules (software tools for traffic analysis);
- message queue service;
- decision module;
- module for logging actions taking place in the Continent WAF interface;
- module for creating log files;
- backend for the management web application interface.

You can see the physical boundaries of Continent WAF, links between its components and environment in the figure below.



Continent WAF is a firewall of the application level that protects web applications from Internet threats. It performs:

- web application traffic analysis and attack (intrusion) detection;
- blocking network attack attempts when working with web applications;
- protecting web applications from the main threat types:
  - various injection types (SQL injection, OS injection, RCE, XPath-injection, XXE);
  - Directory traversal, Remote/Local File Inclusion attacks;
  - XSS;
  - CSRF;
  - attacks exploiting security misconfigurations;

  - brute force attacks;
  - application level DoS;
  - attacks exploiting authentication system weaknesses (session fixation, session theft, missing timeout, etc.);
  - authorization mechanism attacks (Insecure Direct Object References, Missing Function Level Access Control);
  - web scraping, automation;
- detecting suspicious activity of web application users;
- automatic configuration of Continent WAF according to a specific web application (learning);
- access control to functions and logging user actions for a web application;
- integration with SIEM and issue tracking systems.

Users

Continent WAF includes the built-in user groups presented in the table below.

Group	Purpose
Administrator	All members of this group have unlimited rights and full access to all web interface functions
Analyst	All members of this group have restricted rights to: • view and edit data in the **Overview**, **Events**, **Rules**, **Webapps** sections except for permanent deletion of security events, and only within an application to which they are granted access; • create reports and configure notifications for their account
User (read only)	All members of this group have access to a limited interface (the **Settings** section is not available) and have the right to view data related to a web application but cannot edit it

Access control

Firewall administrator

Responsibilities:

- install and configure Continent WAF;
- change the Continent WAF operation mode;
- restart Continent WAF analyzers;
- add and edit Continent WAF rules;
- add and remove web applications;
- analyze security events;
- decide on a response to detected events;
- check Continent WAF operation when deploying new versions of web applications;
- interact with the department that uses the web application when analyzing error messages, especially when it comes to false positives.

Qualifications:

- knowledge of information security;
- knowledge of protected corporate system design basics;
- computer network administering skills;
- knowledge of web technologies;
- knowledge of TCP/IP protocols;
- Ubuntu OS administering skills;
- Continent WAF hands-on skills;
- knowledge of Continent WAF architecture, operation and administering principles.

Analyst

Responsibilities:

- monitor web applications' state;
- configure and maintain the configuration of a web application considering its features;
- monitor and analyze security events;
- decide on response to detected events;
- check Continent WAF operation when deploying new versions of web applications;
- interact with the department that uses the web application when analyzing error messages, especially when it comes to false positives.
- interact with the Continent WAF administrator.

Qualifications:

- knowledge of Continent WAF configuration procedures considering web application features;
- knowledge of web application behavior (input/output, encoding, etc.).

User

Responsibilities:

- monitor web applications' state;
- interact with the analysts and/or the administrator in the case of contingencies.

Qualifications:

- technical education;
- information security knowledge;
- knowledge of web technologies.

A user account is assigned the read only role.

System requirements

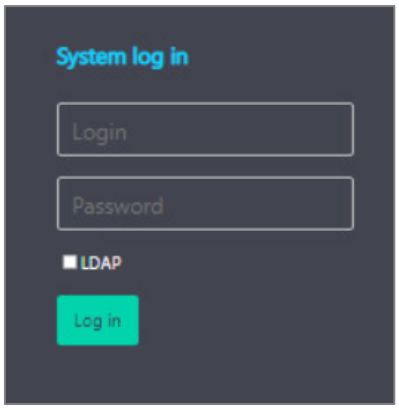
You can see the recommended system requirements in the table below:

Component	Recommended requirement
Operating system	• Ubuntu 20.04 Server; • Astra Linux Special Edition 1.6
RAM	at least 16 GB
CPU	x86_64 with 4 cores, at least 2.2 GHz
Hard drive	at least 500 GB
Network interfaces	• at least 2x Gigabit Ethernet for active mode; • 1x Gigabit Ethernet for passive mode
Web browser	• Google Chrome 88 or later; • Mozilla Firefox 85 or later

Chapter 2
Administration interface

You can administer Continent WAF using the management program (console), which looks like a web interface. To connect to Continent WAF, launch a web browser and enter **https://IP:8443** in the address bar, where IP is the address of the node with Continent WAF.

A welcome screen appears.

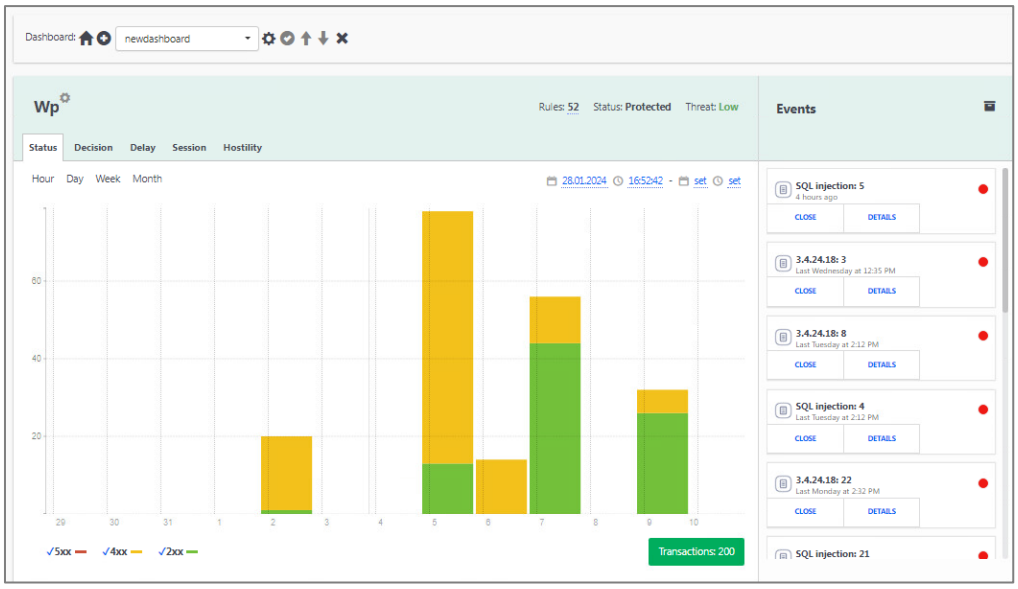


Enter your credentials and click **Log in**. An account with the name *admin* is created by default. Its initial password is specified in the **/etc/waf/dashboard/config.yml** configuration file.

After the first login to the admin account, the system requires you to change the initial password to a new one.

In case of successful authorization, the **Overview** window of Continent WAF opens.

An example of the **Overview** section interface is shown in the figure below.

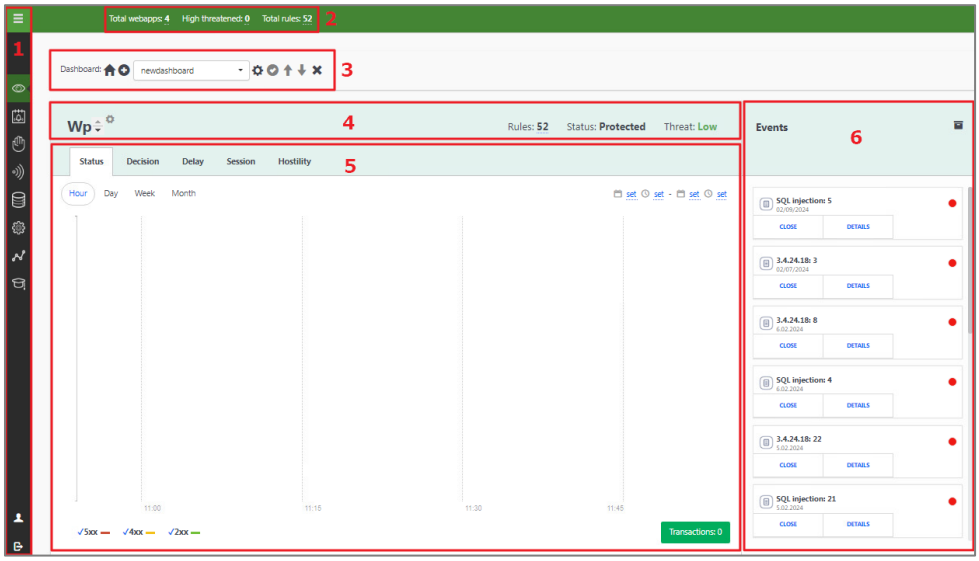


Overview

The **Overview** section allows you to view activity charts for each application and assess how the load on the application and the number of events of different threat levels change over the specified time period.

The **Overview** section is a homepage and opens by default after authorization. Click  on the Navigation panel to open the **Overview** section.

An example of the **Overview** section interface is shown in the figure below.

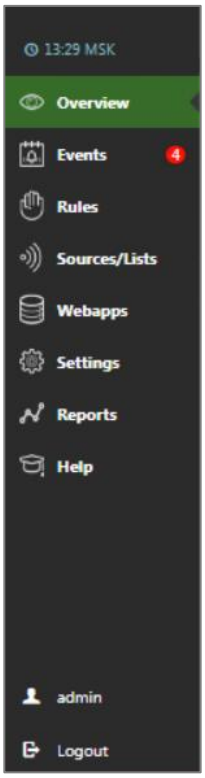


The **Overview** section shown in the figure above has the following elements:

1. Navigation panel.

   The Navigation panel is displayed for all roles.

   On the Navigation panel, click  to view the list of the available sections:



2. **General**.

   The **General** subsection contains data about the number of protected applications, applications with a high threat level and rules created.

3. **Dashboard**.

   The **Dashboard** subsection allows you to switch to the default set of applications (dashboard) and create a new set of applications for Continent WAF.

4. **Application title**.

   The **Application title** subsection contains the application title, data about the number of rules applied to this application, its status, as well as the threat level.

   There are the following statuses for the application:

   - **Monitored** — active protection mode is disabled;

- **Protected** — active protection mode is enabled.

There are the following threat levels possible:

- High;
- Medium;
- Low.

5. **Event charts**.

The **Event charts** subsection contains five tabs, as well as event charts with legends and data about the total number of transactions.

The purpose of the tabs is described in the table below.

Tab	Purpose
Status	Opens by default if you go to the **Overview** section. The charts on this tab display the number of transactions (vertical axis) per unit of time (horizontal axis), broken down by HTTP response codes. There are the following groups of HTTP transactions: • **2xx** — transactions with the 1xx, 2xx and 3xx response codes corresponding to successful operations. Displayed in green on the chart; • **4xx** — transactions with the 4xx response codes — client errors (for example, 404 — a non-existent page request or 403 — a request that requires authorization). Displayed in yellow on the chart; • **5xx** — transactions with the 5xx response codes — server errors. The presence of such errors indicates the incorrect operation of the web server, you must pay special attention to them. Displayed in red on the chart
Decision	Displays the decisions made by the system about transactions. The vertical axis shows the number of transactions, while the horizontal axis shows the time of the transaction registration. There are the following groups of transactions: • **Allowed** — the transaction is allowed to pass; • **Blocked** — the transaction is blocked; • **Modified** — the request is passed with changes and/or the response is changed before being shown to the client
Delay	Displays the average delay in processing a transaction by Continent WAF, as well as the average delay in processing a request by a protected application
Session	Displays the number of active sessions of the protected application at a given time
Hostility	Displays the degree of the environment's hostility at a given time

In all tabs with multiple charts, you can highlight the required chart by moving the cursor to the required parameter in the chart legend. When moving the cursor to a specific diagram column or a chart point, a window appears with the chart parameters at a specific point.

You can change the period for which information is provided in all tabs. To quickly change the period, click **Hour**, **Day**, **Week** or **Month** in the upper-left corner of subsection 5. To set a concrete display period, use the fields in the upper-right corner of subsection 5.

The **Transactions** button contains data about the number of transactions. Click it to go to the **Applications** section. For detailed information on transaction details, see p. 26.

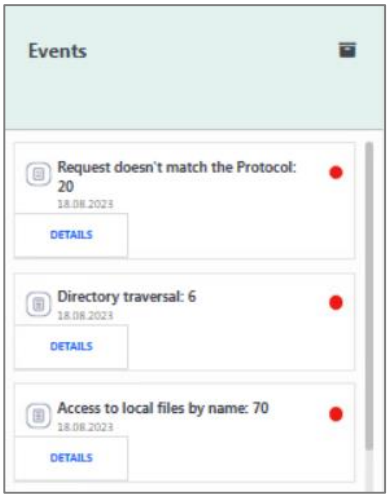
6. **Events**.

The **Events** subsection contains information about the latest events: the name of the triggered rule, the time of the event and the threat level indicator (high — red, medium — yellow, low — green). The number indicated in the red oval above the **Events** label corresponds to the number of new events.

Click  to go to the **Archive** section. To view detailed information about each event, click **Details**.

Click **Close** to archive the event (mark it as closed). This button is also available for users with administrator and analyst rights.

An example of the **Events** subsection interface is shown in the figure below.

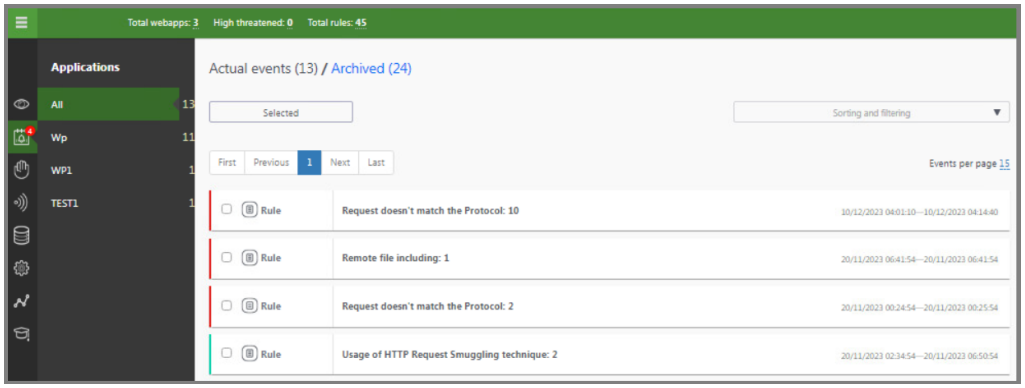


Events

You can view the list of events related to all or specified applications and sort them by their severity, threat levels and their lifetime in the **Events** section.

If you go to the **Events** tab, a second-level menu appears. This menu is a list of applications available for users. To the right of the application, the number of relevant events is shown.

An example of the **Events** section interface is shown in the figure below.



The **Actual events** section opens by default. This section includes security events that are not resolved or seen by a user. The **Archived** section includes security events resolved by a user.

To switch between the **Actual events** and **Archived** sections, click the respective section name.

By default, all events are sorted by the following criteria: view state, severity, update time, rule, start time, status. This means that new, severe, rule-based events are displayed on top. They are sorted by the time the event is changed by the user or the occurrence of a new incident related to this event, as well as by the start time and status.

To change the sorting rules, click **Sorting and filtering**. A dialog box appears as in the figure below.

To change the order of sorting by a specific parameter, click  or  next to the parameter. To change the priority of sorting parameters, move the required parameter using the  arrows to the right of the parameters.

To filter the events, specify the filtering criteria on the right side of the **Sorting and filtering** menu. Specify the rule name or key words for the search. If necessary, specify the date and time of events and select one of the parameters **All/With severity/New**.

To minimize the **Sorting and filtering** dialog box, click  in the upper right corner.

Events are displayed in the form of a sorted list. Each event has a severity indicator on the left:

- red — for events with high severity;
- yellow — for events with medium severity;
- green — for events with low severity;
- gray — for events whose severity is not set.

An example of the above-mentioned indicators usage is shown in the figure below.

Every event line contains a check box, a name of the triggered rule, a number of transactions and a time of the event. Click an event line to view general information about the event.

An example of the event general information is shown in the figure below.

To preview the event information, click the required item:

- **Top IP addresses** — displays top 10 IP addresses with the number of transactions, in descending order;
- **Top actions** — displays top 10 IP addresses with the number of transactions, in descending order (if there are no transactions in the event, the **Not found** message is displayed);
- **Anomalies** — displays top 10 anomalies detected in the event transactions;
- **Modsecurity rule ids**— displays top 10 modsecurity signature IDs detected in the event transactions;
- **Anomaly locations** — displays top 10 anomaly locations of the transactions with detected anomalies;
- **Description** — displays a rule description that caused the creation of the event.

Click **Details**, to view event detailed information. It is available in the administrator, analyst and operator interfaces.

The **Delete** button is available only in the administrator interface. Click **Delete** to delete an event. The window prompting you to confirm the deletion appears, click **OK**. You cannot recover the deleted event.

The **Close** button is available only in the administrator and analyst interfaces. Click **Close** to archive the event as completed.

To perform group operations, click **Selected**. It is available only to the administrator and analyst. It is displayed below the **Actual events** or **Archived** sections.

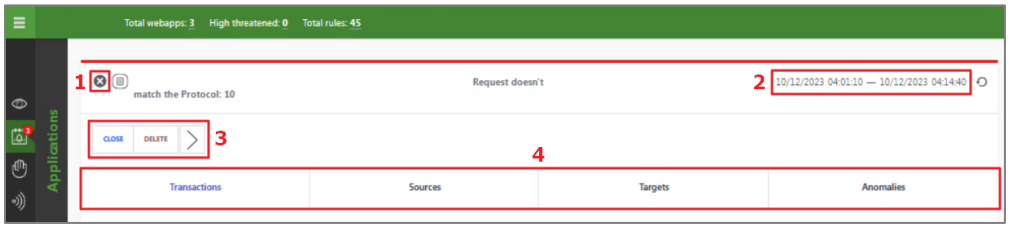
You can manage events in groups by clicking the following buttons:

- **Close** — moves the application to the archive;
- **Mark as read** — moves the event to the end of the list;
- **Delete** — deletes the event (deleted events are not moved to the archive);
- **Check all** — selects all actual events;
- **Remove selection** — removes all check marks.

Detailed information about event

To view detailed information about an event, click the event and click **Details**.

A window with detailed information appears as in the figure below.

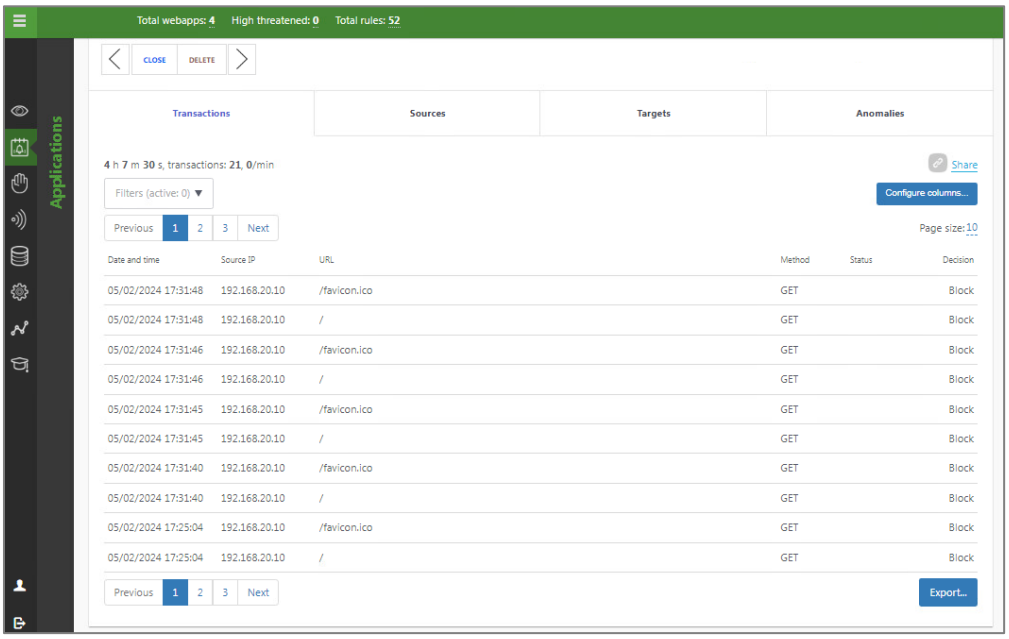


The **Details** window of each event includes the following elements (see the figure above):

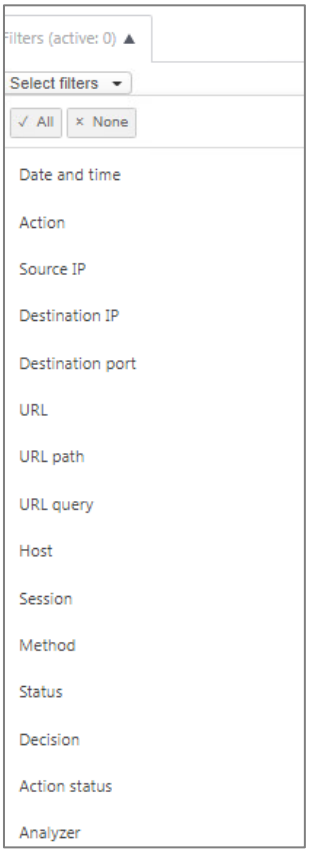
1. **Exit** (icon) — click it to exit the **Details** window and go to the **Event** section. The viewed event is placed at the end of the list of current events and marked as read.
2. **Event period** — displays the period for which event transactions are collected. To view the latest changes to the event, click .
3. **Event view management** — there are 4 buttons to configure the view:
   - / — previous/next security event;
   - **Close** — moves the selected event to the archive. This button is available only in the administrator, analyst and operator interfaces;
   - **Delete** — deletes an event without the possibility of recovering it. This button is available only in the administrator interface.
4. **Analytics** — there are 4 tabs available to analyze transactions:
   - **Transactions** (see p. 15);
   - **Sources** (see p. 17);
   - **Targets** (see p. 18);
   - **Anomalies** (see p. 18).

Transactions tab

The **Transactions** tab contains a list of all transactions of the event, the duration of the event, and the frequency of transactions. An example of the **Transactions** tab interface is shown in the figure below.



You can use the specified filters to search for transactions. To do that, click **Filters** and fill in the required fields for each selected filter. To cancel filtering, click **None** in the **Select filters** drop-down list.



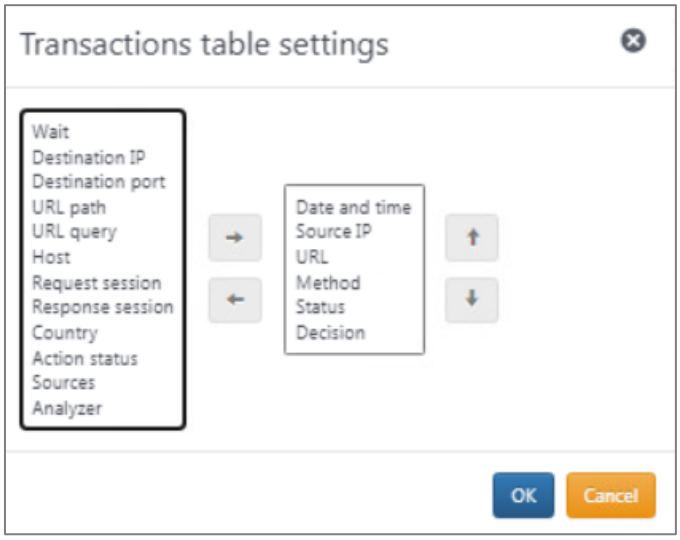
The filters and their descriptions are listed in the table below.

Filter	Purpose
Date and time	Selection of the period for filtering, for which you want to output transactions included in the event

Filter	Purpose
Action	Selection of one or more actions that transactions from the selected event are included in. The action is selected from a drop-down list with a search function
Source IP	Display of transactions of the selected event in which the src_ip header matches the manually entered source (client) IP address. After specifying one port, click to display an additional field specifying another port. If you click , the line against which the icon is located will be deleted
Destination IP	Display of transactions of the selected event in which the dst_ip header matches the manually entered recipient (server) IP address. After specifying one port, click to display an additional field specifying another port. If you click , the line against which the icon is located will be deleted
Destination port	Display transactions of the selected event in which the dst_port header matches the manually entered port of the recipient (server). Click / to increase or decrease the value entered in the field. After specifying one port, click to display an additional field specifying another port. If you click , the line against which the icon is located will be deleted
URL	Display of transactions for the selected event in which the URL header (the URL header contains all PATH and QUERY header values) contains a manually entered value for this filter field
URL path	Display of transactions of the selected event in which the PATH header contains a manually entered value for this filter field
URL query	Display of transactions of the selected event in which the QUERY header contains a manually entered value for this filter field
Host	Display of transactions of the selected event in which the HOST header matches a manually entered value for this filter field
Session	Display of transactions of the selected event in which the session ID matches a manually entered value for this filter field. This filter works only if the session model is configured
Method	Selection of one or more methods by which transactions from the selected event match. The method is selected from the drop-down list
Status	Selection of one or more status options that came in response to transactions from the selected event
Decision	Selection of one or more of the suggested solutions for the transactions from the list to filter on
Action status	Selection of one or more of the proposed success options for action on transactions from the list
Analyzer	Selection of the analyzer that provided the transaction solution for the selected event

In the **Transactions** tab, you can change the information display view by clicking **Configure columns**.

The **Transactions table settings** window appears.



The list on the right displays parameters available for displaying in the **Transactions** tab. The list on the left displays parameters already displayed in the section. To move a parameter to another list, click it and click  or  . To change the parameter's order, click  and  .

Descriptions of the possible parameters are listed in the table below.

Filter	Description
Wait	Lag time of the Continent WAF response
Source IP	src_ip header (client IP address) value
Destination IP	dst_ip header (server IP address) value
Destination port	dst_ip header (server port) value
URL	URL header value
URL path	PATH header value
URL query	QUERY header value
Host	HOST header value
Request session	128-character request session ID (requires a configured session model)
Response session	128-character response session ID (requires a configured session model)
Country	Country flag of the client IP address
Action status	(requires configured action success)
Source	All sources found in the transaction
Analyzer	Selecting the analyzer that produced the transaction solution from the selected event
Date and time	Date and time when a transaction was received by Continent WAF
Method	Method used for sending the transaction
Status	Server response status
Decision	Decision upon the transaction by Continent WAF

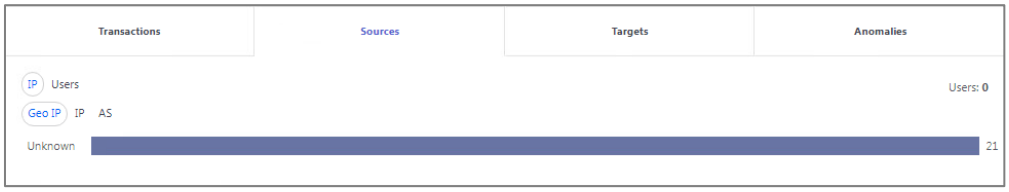
On the **Transactions** tab, click **Share** to copy the link to the **Transactions** tab for the event.

Double-click a transaction in the list to open the **Transaction details**. For detailed information on transaction details, see p. 39.

Sources tab

On the **Sources** tab of the window with detailed information about an event, information about the IP addresses and users of the sources is available.

An example of the **Sources** tab interface is shown in the figure below.



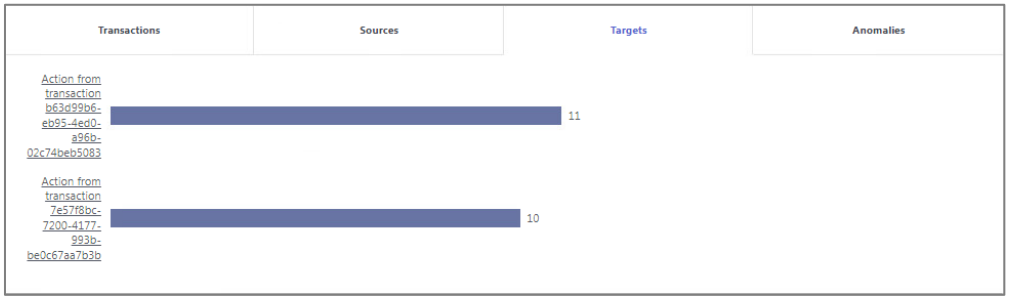
You can view the following information on the tab:

- Geographical location — top countries of the IP addresses of transaction sources;
- IP addresses — top IP address of the event;
- Autonomous systems — top autonomous systems to which IP addresses belong;
- Users — top users of the transaction (to extract user IDs from HTTP transactions, configure the **User name extraction** session ID).

Targets tab

On the **Targets** tab, information about the top actions of the positive model of the business logic is displayed.

An example of the **Targets** tab interface is shown in the figure below.



Note. Some values in these statistics are underlined, such as IP addresses (**Sources** tab) or actions (**Targets** tab). If you click on a value, the transaction log opens filtered by the selected value.

Anomalies tab

On the **Anomalies** tab, information about detected anomalies is available by analysis modules and by type.

An example of the **Anomalies** tab interface is shown in the figure below.

The following information is available for viewing on this tab:

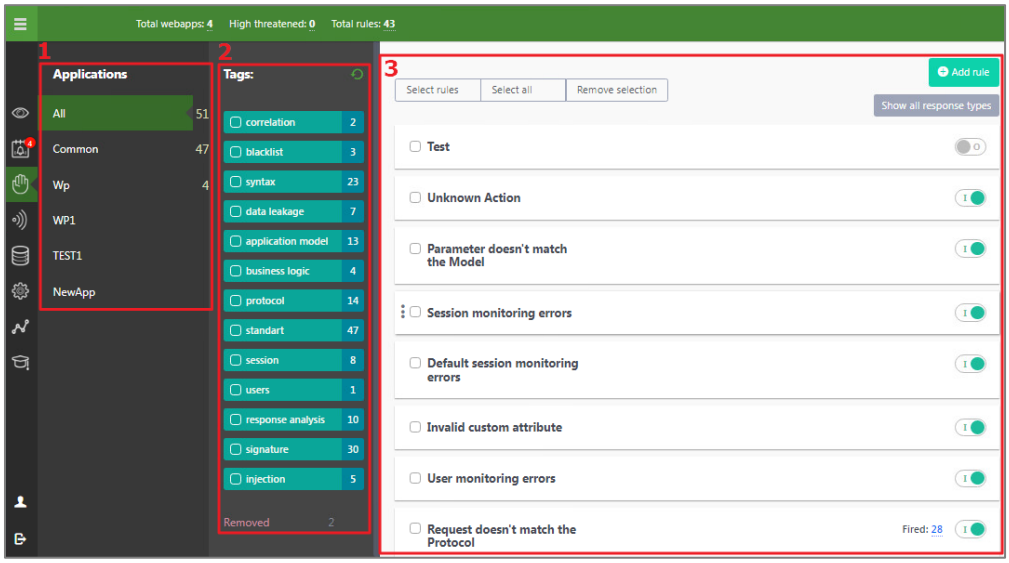
- statistics on anomalies from different analyzer modules, under which transactions from this event fell;
- statistics on the types of anomalies under which transactions from the given event fell.

Rules section

Reaction rules manage responses and requests based on their attributes and anomaly sets. Continent WAF has a minimum set of rules by default. A rule consists of a condition (what event should happen) and an action (what to do in this case).

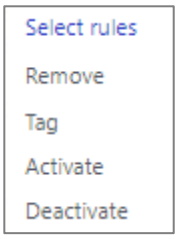
You can view, create, edit, enable and disable rules for all protected applications or for a particular application. You can also filter them by categories (tags) or by actions (block, allow or mark).

An example of the **Rules** section interface is shown in the figure below.



The **Rules** section includes the following panels:

1. **Applications** — contains a list of all applications available to the user. To the right of the application name is the number of rules created for it. Click on an application to view the application rules.
2. **Tags** — contains a list of keywords (tags) used in the rules. To the right of the tag is the number of rules for which it is used. When you select a tag, all rules for which it is selected are displayed. To select a tag, click the tag name; to reset a tag selection, click it again or click  to reset selection.
3. **Rules** — contains a list of created rules. Multiple actions are available for rules. To select all rules, click **Select all**. To reset the selection, click **Remove selection**. Hover your mouse over **Select rules**. A submenu appears as shown in the figure below.



You can select the following commands in this submenu:

- **Remove** — deletes all selected rules;
- **Tag** — adds a tag to all selected rules;
- **Activate** — enables all selected rules;
- **Deactivate** — disables all selected rules.

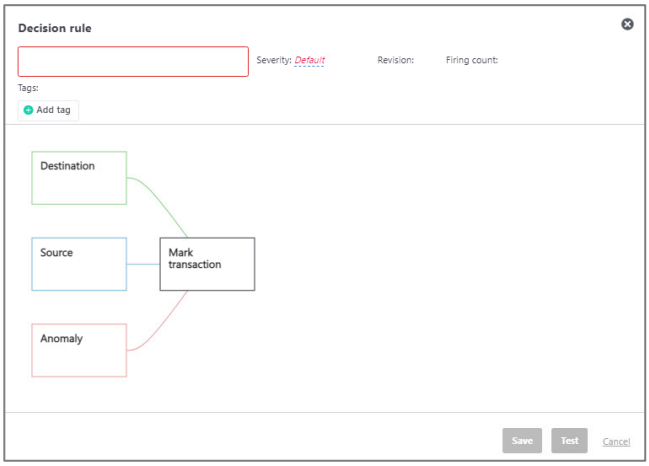
Click **Show all response types**, to view rules grouped by transaction actions (block, mark, allow). By default, all actions are displayed but a submenu with a selection of actions appears when you hover your mouse over a filter.

An administrator and analyst can create new rules.

To add a new rule:

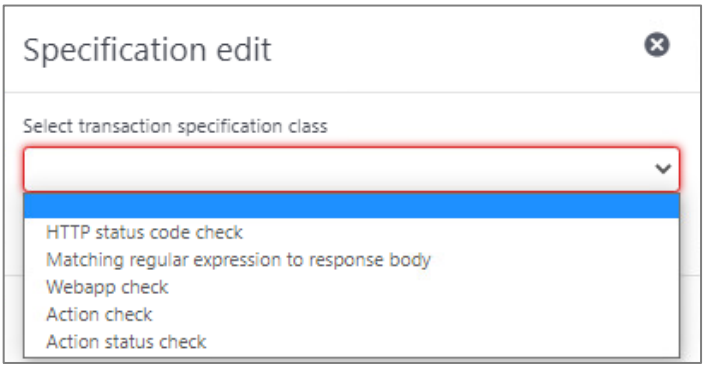
1. Click **Add rule** in the right corner of the **Rules** panel.

The **Decision rule** dialog box for creating a new rule appears.



2. Specify the rule in the respective field. If necessary, add one or several tags by clicking **Add tag**. Click **Default** to change the rule severity.
3. Specify the condition that triggers the rule:
   - Hover your mouse over the **Destination**, **Source** or **Anomaly** item.
   - Click the **+** button that appears.

The **Specification edit** dialog box appears.



   - Select the specification class from the respective drop-down list.
   - Click **Mark transaction**.
   - Select the required action type (**Block**/**Allow**/**Mark transaction**) from the **Change action** drop-down list.
4. Click one of the following buttons when you finish editing the rule:
   - **Save** — saves the settings.

     The created rule appears in the **Rules** list.
   - **Test** — displays the operation of the created rule on transactions previously registered for this application.
   - **Cancel** — closes a dialog box without saving the settings.

All possible parameters used when creating rules are listed in the table below.

Specification	Possible transaction specification classes	Possible parameters	Note
Destination	**HTTP status code check**	Specify response codes	The rule only applies to the listed response codes

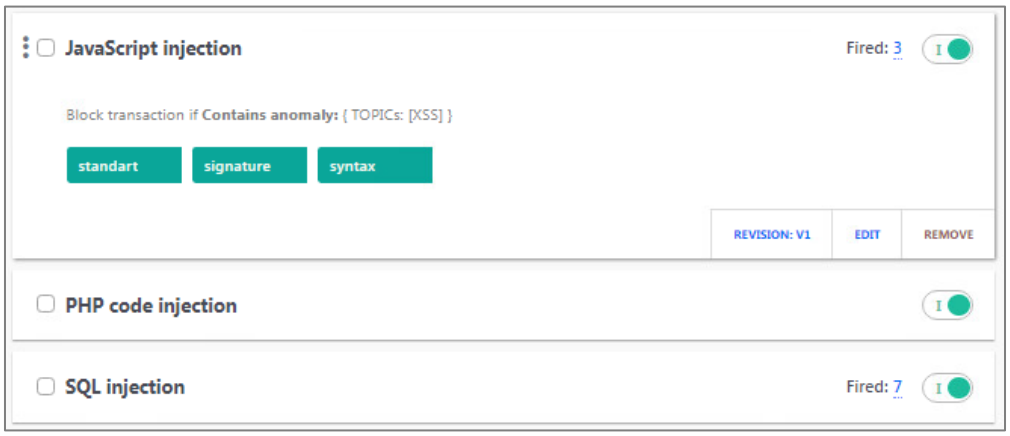
Specification	Possible transaction specification classes	Possible parameters	Note
		Use specification negation	The rule applies to all response codes except for the listed ones
	Matching regular expression to response body	Regular expression	The rule applies to responses that contain the specified regular expression
		Use specification negation	The rule applies to responses that do not contain the specified regular expression
	Webapp check	Specify a web application	The rule applies to the traffic of a specified web application
		Use specification negation	The rule applies to traffic of all applications except the specified one
	Action check	Specify an action	It is used only after the **Webapp check** parameter. The rule applies only to the application traffic corresponding to the selected action
		Use specification negation	It is used only after the **Webapp check** parameter. The rule applies to application traffic, except for the traffic corresponding to the selected action
Source	**Session identifier check**	Specify a session ID	The rule applies to requests whose session ID matches the specified
		Use specification negation	The rule applies to requests whose session ID does not match the specified
	Request without a session identifier	—	The rule applies to requests without a session ID
		Use specification negation	The rule does not apply to requests without a session ID
	Username check	Specify a user name	The rule applies to requests related to the specified user
		Use specification negation	The rule applies to all requests related to users except for the specified one
	Unauthorized request	—	The rule applies to requests from unauthorized users
		Use specification negation	The rule applies to all requests except those from unauthorized users
	User IP address check	Specify an IP address	The rule applies to requests from the specified IP address with the specified subnet mask
		Subnet mask	
		Use specification negation	The rule applies to all requests except those from the specified IP address with the specified subnet mask
	User country check	Specify a country	The rule applies to requests from the specified IP address of the specified country
		Use specification negation	The rule applies to all requests except those from the specified IP addresses of the specified country
	User ASN check	Specify an ASN	The rule applies to requests from the specified ASN

Specification	Possible transaction specification classes	Possible parameters	Note
		Use specification negation	The rule applies to all requests from ASNs except for the specified one
Anomaly	**Any anomaly existence check**	—	The rule applies to transactions with the detected anomaly
		Use specification negation	The rule applies to traffic without detected anomalies
	Check for anomaly with specifies parameters	Select anomaly analyzer (**Lightweight session tracker**; **User tracker**; **Session tracker**; **Open redirect detector**; **Csrf detector**; **Modsecurity analyzer**; **Action sequence anomaly detector**; **Bruteforce detector**; **Integration with signature-based analysis on ICAP-server**; **Action determiner**; **Action param validator**; **Session anomaly counter**; **Libinjection detector**; **Decision tree request parser**; **Decision tree response parser**; **Nginx zmq adapter**)	The rule applies to transactions with an anomaly with specified parameters. Required filed
		Select anomaly topics	The rule applies to transactions with anomalies detected by a specified module
		Select anomaly location type (**Any**; **Message**; **Start line**; **Header**; **Raw body**; **Body**; **URL**; **Query**; **Cookie**; **Parse tree**; **Session data**; **Source**; **Target/Web application object**)	The rule applies to transactions with an anomaly detected at the specified location by a specified module
		Select anomaly location ID	The rule applies to transactions with a detected anomaly with the specified location and the location name
		Use specification negation	The rule applies to all transactions except those with the specified parameters
Action	**Mark transaction**	—	A transaction that matches the **Destination**, **Source** and **Anomaly** specifications is marked for further analysis. Used as an action by default
	Allow transaction	—	A transaction that matches the **Destination**, **Source** and **Anomaly** specifications is allowed to pass as legitimate

Specification	Possible transaction specification classes	Possible parameters	Note
	Block transaction	—	A transaction that matches the **Destination**, **Source** and **Anomaly** specifications is blocked

In order to search for rules, select the required rule tags in the **Tags** section and specify the type of action in the **Show all response types** drop-down list. There are the following response types: all actions, allowing, blocking and marking. Click to reset the tags.

To view the general rule information, click the required rule. The general information includes the rule name, its description and tags, revision (a rule version) and trigger count. An administrator and analyst can enable or disable, edit, delete or move a rule in the list. An operator can only view general rule information.

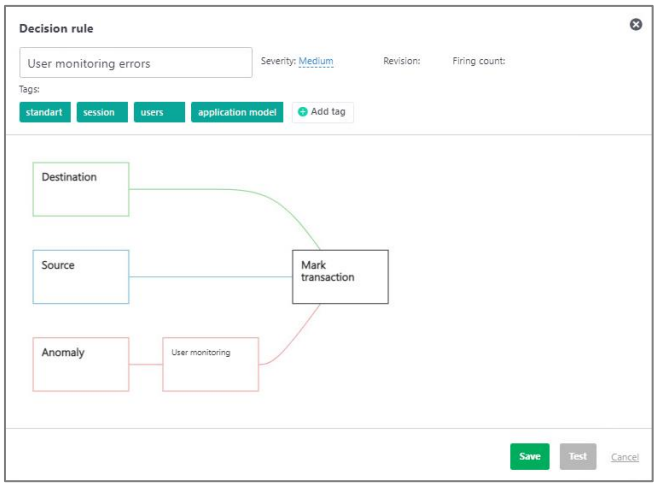


To enable or disable a rule, turn on the toggle in the upper-right corner of the rule subsection.

If you click **Delete**, a rule is moved to the **Removed** section under the tag list. To restore a rule, go to the **Removed** section by clicking the **Removed** link. Then select the required rule and click **Restore**.

Attention! A restored rule is disabled by default. If necessary, enable it.

If you click **Edit**, the **Action** window appears as shown in the figure below. You can add and delete tags, destinations, sources, anomalies and actions and configure the rule severity in the section.



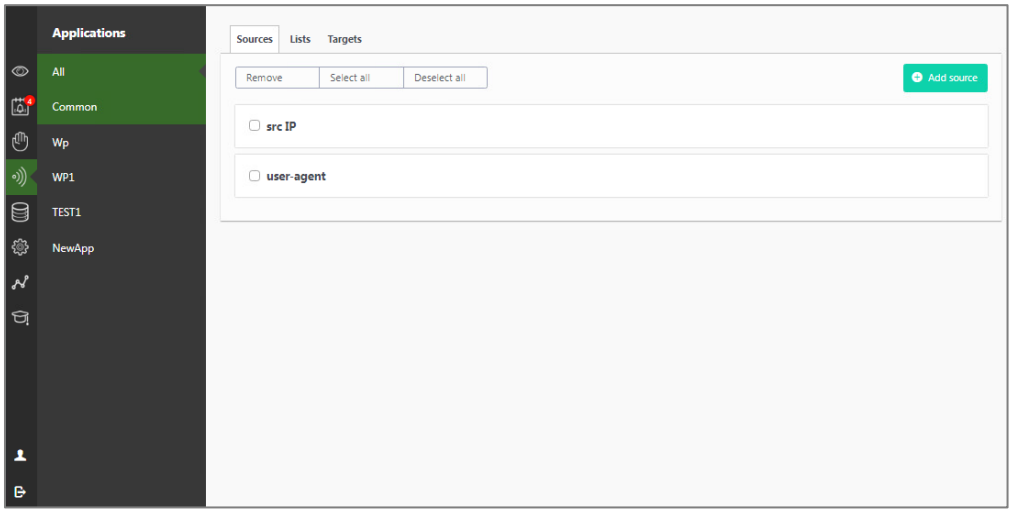
To move a rule, hover the mouse over the left corner of the rule subsection. Then click three vertical points that appear and hold the left mouse button while moving the rule. A rule's priority depends on its position in the list: the higher a rule is in the list, the higher its priority is.

Sources and lists section

You can view, add, edit and delete sources and lists in this section. All users can view, add and delete sources and lists. An operator can only view the tabs data.

Sources tab

The **scr IP** list is provided by default in the **Sources** tab. An administrator and analyst can create new sources. An example of the **Source** tab interface is shown in the figure below.



Note. The default source named **src_ip** is typically used to create brute-force detectors. This source cannot be deleted.

Lists tab

You can create, edit, filter and delete the lists in the **Lists** tab. An example of the **Lists** tab interface is shown in the figure below.

You can filter lists by purpose. By default, all lists are displayed on the tab. To view a particular list, hover your mouse over **Show all tabs**. The view of the menu is given in the figure below.

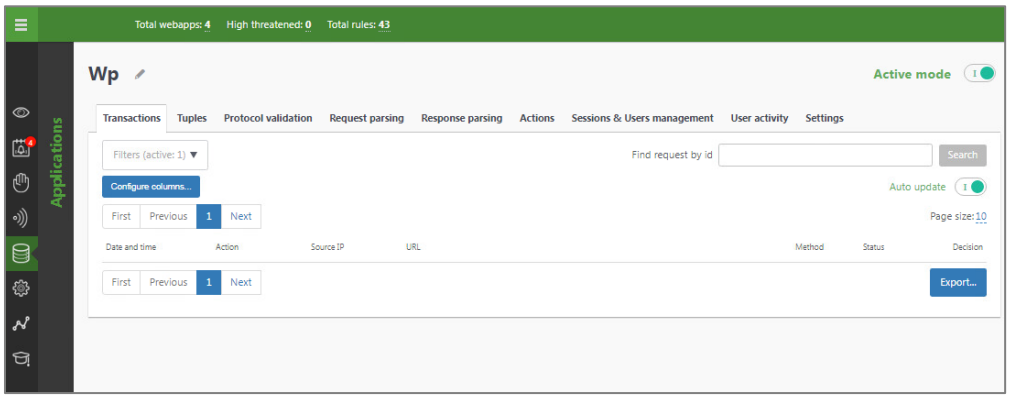
Click the list to view a list information. The window contains the following information:

- a brief list description;
- **Edit** — opens the list editing window;
- **Remove** — permanently deletes the selected list.

Applications section

The **Applications** section is used to monitor traffic and configure web application security models. You can add and delete applications and view application information in this section. You can enable and disable active protection mode of an application.

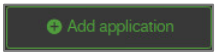
The analyst and administrator interface in the **Applications** section is shown in the figure below.



In the operator interface of the **Applications** section, you can view only application transactions as shown in the figure below.

This section includes the tabs listed in the table below.

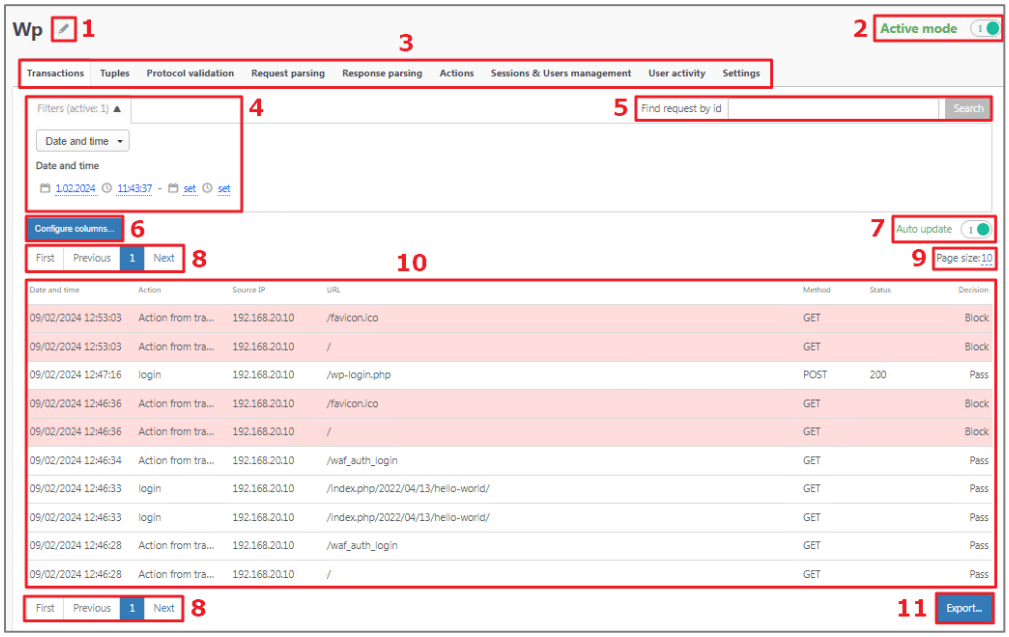
Tab	Purpose
Transactions	View real-time events related to the selected application. This tab is opened by default
Tuples	View the web application domain name, IP address, and the Continent WAF port on which traffic is received. The **Tuples** tab allows you to map traffic coming to Continent WAF to a specific protected application
Protocol validation	Edit positive model for protocol validation
Request parsing	Edit positive model for parsing trees
Response parsing	
Actions	View and edit an action model
Sessions & Users management	View and edit a model for tracking sessions and users of the protected application
User activity	View users of a protected application registered by Continent WAF
Settings	Add headers specific to applications and delete applications

An analyst can add new applications. To do that, click  or 

. For detailed information on how to add applications, see [**1**].

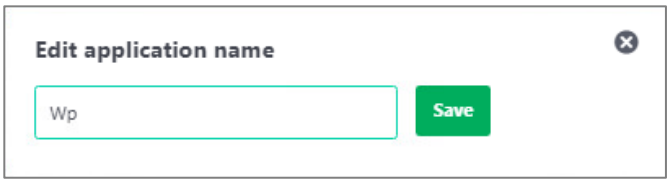
Transactions tab

The **Transactions** tab provides a list of transactions of the selected web application. This tab is used to analyze the traffic passing through Continent WAF. By default, the sampling for transactions is set to 10. This means that every transaction is processed on Continent WAF but only every 10 of the usual transactions are displayed on the transactions tab. In contrast to usual transactions, blocked and marked transactions are not sampled and all transactions are saved.



The **Transaction** tab consists of the following elements:

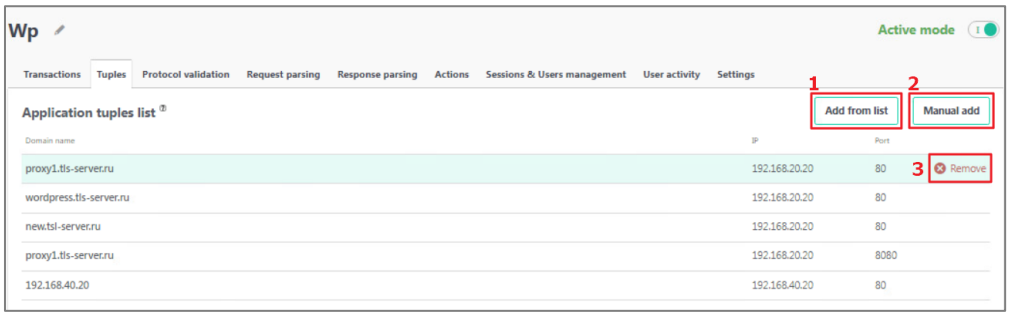
1. **Edit application name** button — click it to edit an application name.



2. Active protection — application protection mode toggle. There are two possible modes:
   - **Monitored** — active protection is disabled. The decision to block transactions is displayed in the interface but is not applied.
   - **Active mode** — active protection is enabled. Accepted blocking decisions are applied to transactions.
3. Tab switching area.
4. **Filters** field — click it to select the required filters from the drop-down list. For more information on how to edit transaction filters, see p. 15.
5. **Find request by id** field — each transaction is assigned its own unique identifier, which can be viewed in the transaction data, blocking message. Specify the transaction identifier in the field and click **Search**.
6. **Configure columns** button — click it to view the **Transactions table settings** window. For more information, see p. 15.
7. **Auto update** toggle — switch for automatic updating of displayed transactions.
8. Pages and moving between pages elements — displays the number of pages, considering the specified number of transactions per page. You can move between pages by clicking **Previous**/**Next** or by selecting a page number.
9. **Page size** — the number of displayed transactions per page. It can be changed by clicking the number in the **Page size** line, then entering the required number of transactions per page and applying the changes.
10. **Transactions** area — this area displays transactions, according to the specified filters, configured columns and the number of transactions on the page. For more details on working with transactions, see p. 39.
11. **Export** button — click it to export transactions based on the specified filters and configured columns. Click **Export** and an additional field appears with the number of transactions to be exported (you can also select all by selecting the **All** check box). Click **Export** again, a dialog box appears where you can download the CSV file.

Tuples tab

This tab includes a list of tuples. A tuple includes the domain name of a web application, IP address and port of Continent WAF which receives traffic. Tuples allow comparing traffic going through Continent WAF with particular web applications.



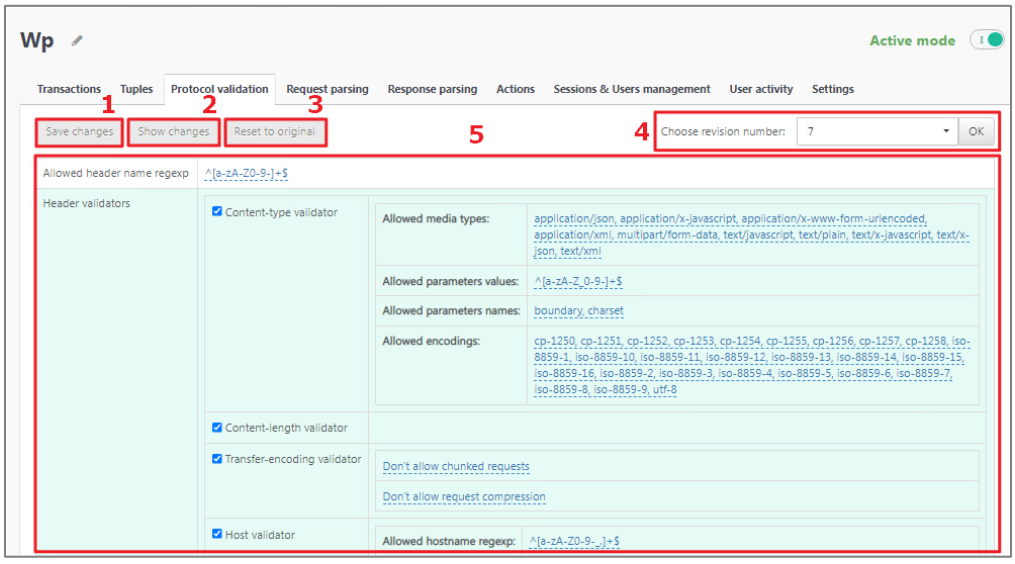
The **Tuples** tab contains the following commands:

1. **Add from list** — opens a dialog box with the IP addresses/names, traffic that is detected by Continent WAF.
2. **Manual add** — opens a dialog box with the domain name, IP address and parameters to specify.
3. **Remove** — permanently deletes a tuple. This button becomes active when you hover your mouse over a tuple.

Tuples cannot be edited. For detailed information on how to add tuples, see [**1**].

Protocol validation tab

The **Protocol validation** tab contains the settings of the positive model protocol validation. An example of the **Protocol validation** tab interface is shown in the figure below.



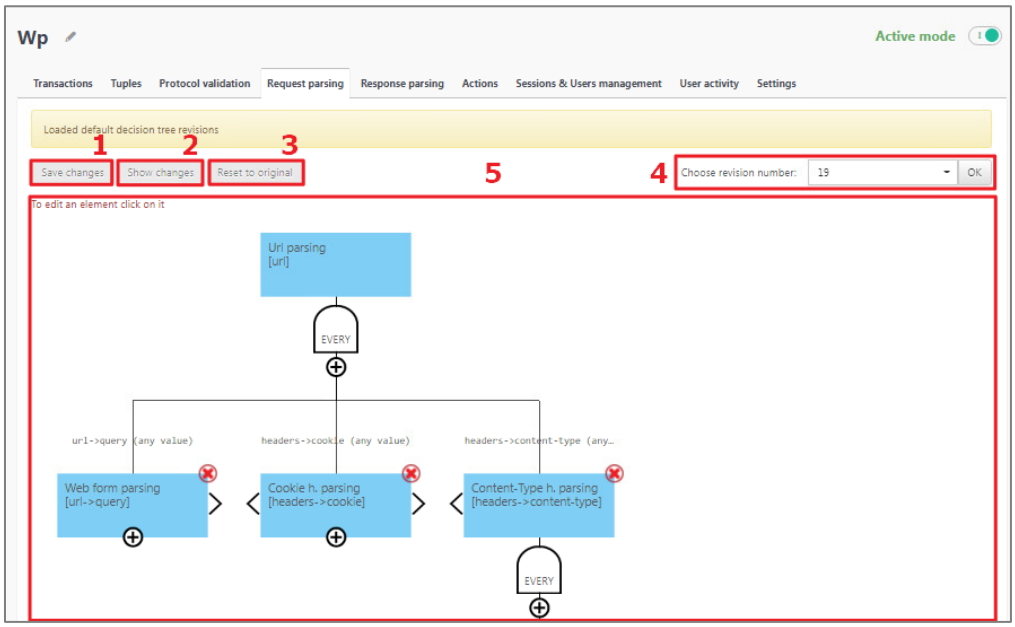
The **Protocol validation** tab includes the following sections:

1. **Save changes** — save the changes made and create a new revision. The button becomes active when you make changes to the protocol parameters (section 5).
2. **Show changes** —compares the entered values with the current revision and shows the changes in a dialog box. The button becomes active when you make changes to the protocol parameters (section 5).
3. **Reset to original** — discards the changes. The button becomes active when you make changes to the protocol parameters (section 5).

4. **Choose revision number** — switches between revisions of the protocol settings. Specify the required revision number and click **OK**. To apply the selected revision, click **Revert**.
5. This section includes the protocol validation setting. For detailed information on protocol validation settings, see p. 49.

Request parsing and Response parsing tabs

You can view a request decision trees in the tab. An example of the tab interfaces is shown in the figure below. The tabs are identical in terms of possible settings.

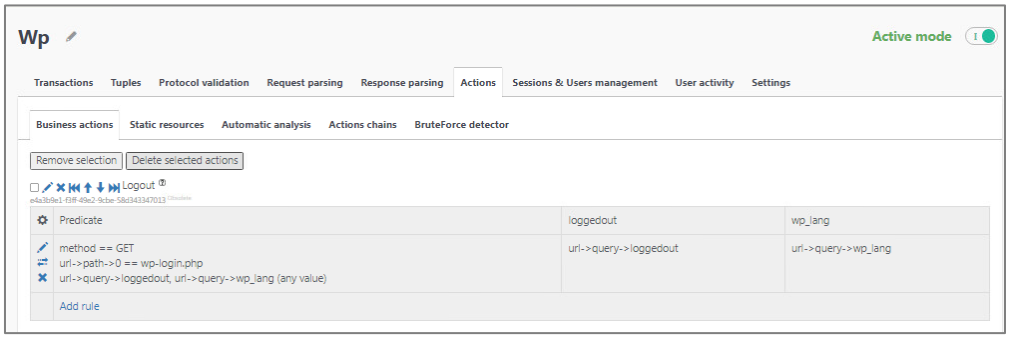


The **Request parsing** and **Response parsing** tabs include the following sections:

1. **Save changes** — save the changes made and creates a new revision. The button becomes active when you make changes to the decision tree parameters (section 5).
2. **Show changes** — compares the entered values with the current revision and shows the changes in a dialog box. The button becomes active when you make changes to the decision tree parameters (section 5).
3. **Reset to original** — discards the changes. The button becomes active when you make changes to the decision tree parameters (section 5).
4. **Choose revision number** — switches between revisions of the decision tree settings. Specify the required revision number and click **OK**. To apply the selected revision, click **Revert**.
5. This section includes the decision tree setting. For detailed information on protocol validation settings, see p. 51.

Actions tab

You can add new actions and rules, view and edit the existing ones in this tab. An example of the **Actions** tab interface is shown in the figure below.



This tab contains settings of the following tabs:

- **Business actions** — the tab is used for creating, editing and deleting actions;
- **Static resources** — the tab is used for creating and deleting settings for filtering requests to static resources;
- **Automatic analysis** — the tab with setting tasks for self-learning of some modules of Continent WAF;
- **Actions chains** — the tab for creating, editing and deleting action chains;
- **Bruteforce detector** — the tab contains two brute-force detection modules (Action/Source and Action/Source/Target).

Business actions

The **Business actions** tab opens by default when you go to **Actions** tab. You can create and edit a positive business logic model. The model consists of actions and their parameters. It allows you to describe user actions in web applications on a logical level. For example, authentication, registration, search, purchase payment, form submission, etc.

Actions are the matching of HTTP requests with the logic of the web application. When creating an action, you can use conditions not only on the request URL, but also on any other request parameters.

The business logic model can be used to detect and block requests that do not match this model. In addition, the created actions can be used for point suppression of false positives, as a parameter in a rule, for the tweaking of the brute-force attack detector. Actions can also be used as elements of the action chain model, as parameters when configuring the session model, as well as other related modules.

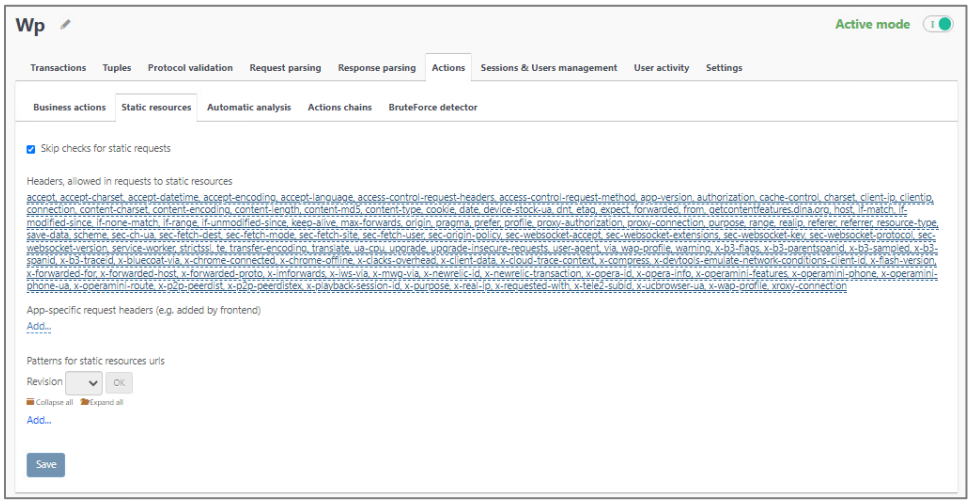
For detailed information on business action settings, see p. 63.

Static resources

The page load of the protected web application also results in a number of requests to download files such as icons, scripts and stylesheets. These requests are also displayed in Continent WAF, which prevents correct traffic analysis.

Filtering of requests to static resources should be configured to reduce the load on Continent WAF analyzers and remove uninformative traffic. By default, each transaction is dropped into a common pool and analyzed (a parse tree is built, signatures are checked, etc.). In order not to analyze transactions to static resources, the **Static resources** tab is used.

An example of the **Static resources** tab interface is shown in the figure below.



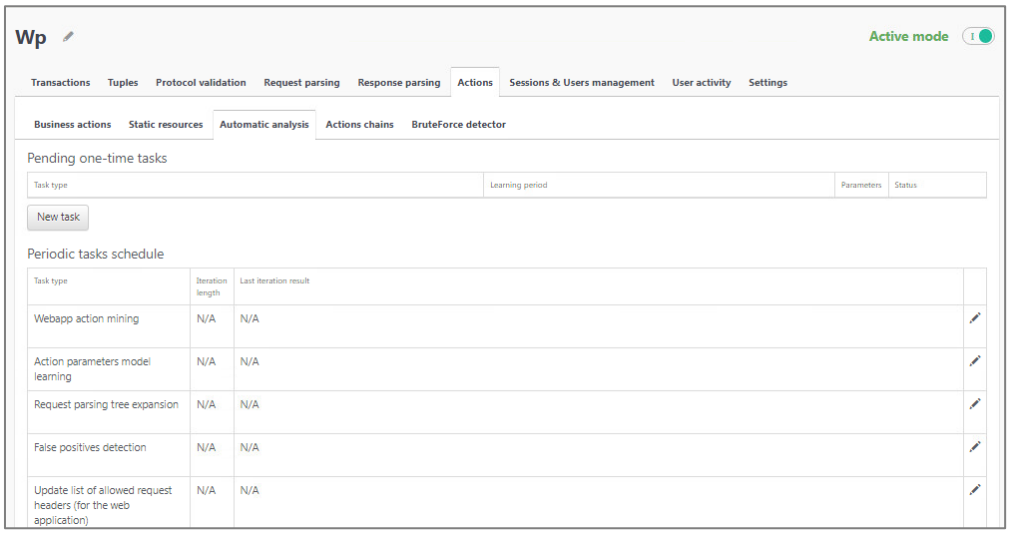
For detailed information on request filtering settings, see p. 43.

Automatic analysis

The **Automatic analysis** tab includes information about periodic tasks. The tab consists of two sections. The **Pending one-time tasks** section includes a schedule of periodic tasks. The **Periodic tasks schedule** section includes finished tasks. You

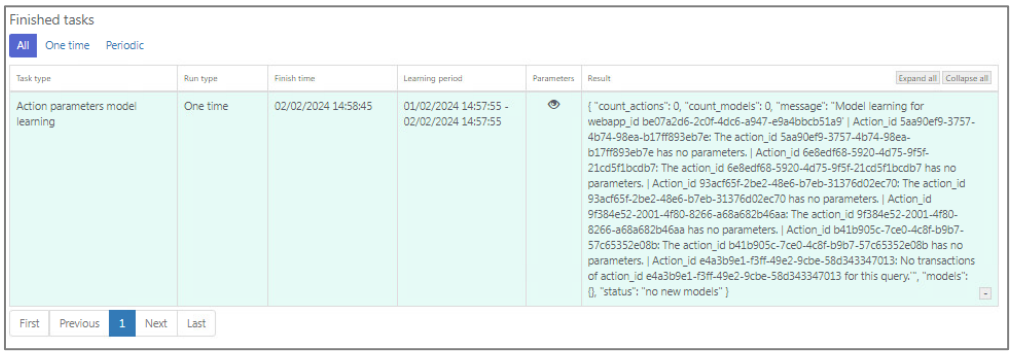
can create a new one-time task by clicking **New task**. The **Periodic tasks schedule** table includes the **Task type**, **Iteration length**, **Last iteration result** columns. The last column has the  icon. Click it to edit the task parameters.

An example of the **Pending one-time tasks** section interface is shown in the figure below.



In the **Finished tasks** section, you can view all finished, periodic and one-time tasks. All finished tasks are sorted from new to old ones. Tasks are highlighted based on the execution results. If a task is successfully completed, it is highlighted in green. Otherwise, it is highlighted in red. To edit a task, click  in the **Parameters** column. To view the result of a task execution, click **+** in the **Result** column.

An example of the **Finished tasks** section interface is shown in the figure below.

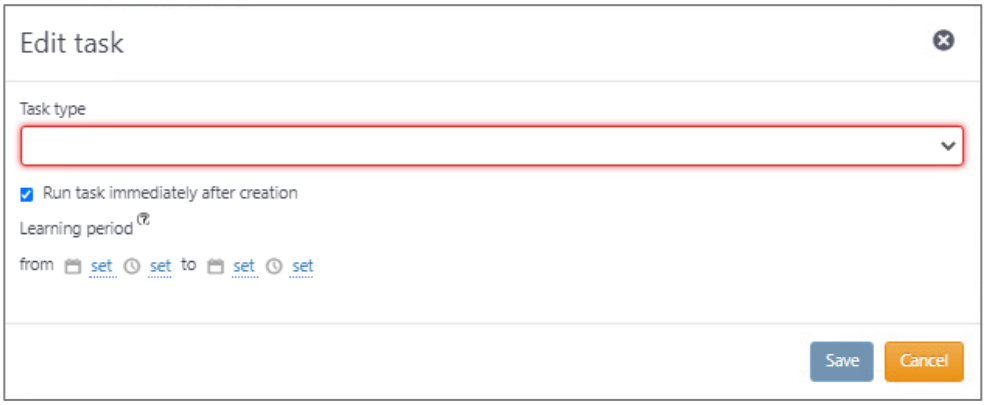


The periodic task types are listed in the table below.

Task	Purpose
Webapp action mining	Creates actions for the application
Action parameters model learning	Configures the application's action model
Request parsing tree expansion	Automatically expands the query tree model for a specified application
False positives detection	Automatically suppresses false alarms based on specified parameters
Update list of allowed request headers (for the web application)	Updates the list of allowed request headers in protocol validation for the selected application
Static sources patterns update	Updates the settings of static resource request filtering templates for the selected application
Static sources URLs update	Adds paths to static resources
Action chains mining	Creates action chains for the application

To add a new one-time task for learning:

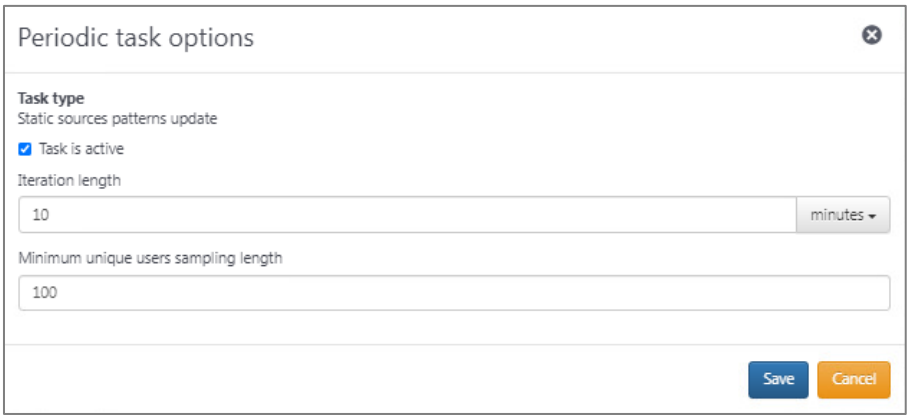
1. Click **New task**.



2. Specify the task type in the **Task type** drop-down list and other parameters.
4. Click **Save**.

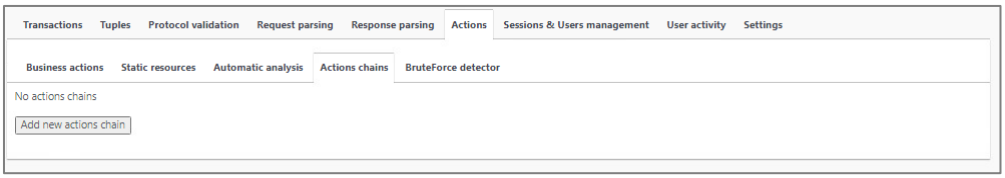
   The record with the task parameters and results appears below in the **Finished tasks** section.

Click  to edit the task parameters. The **Periodic task options** dialog box appears.

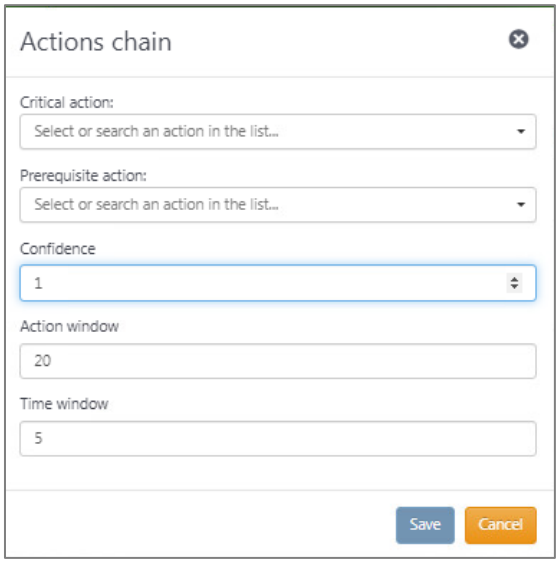


Actions chains

You can create simple action scripts to track in this tab. An example of the **Actions chain** tab interface is shown in the figure below.



Click **Add new actions chain** to create a new action chain. The **Actions chain** dialog box appears.

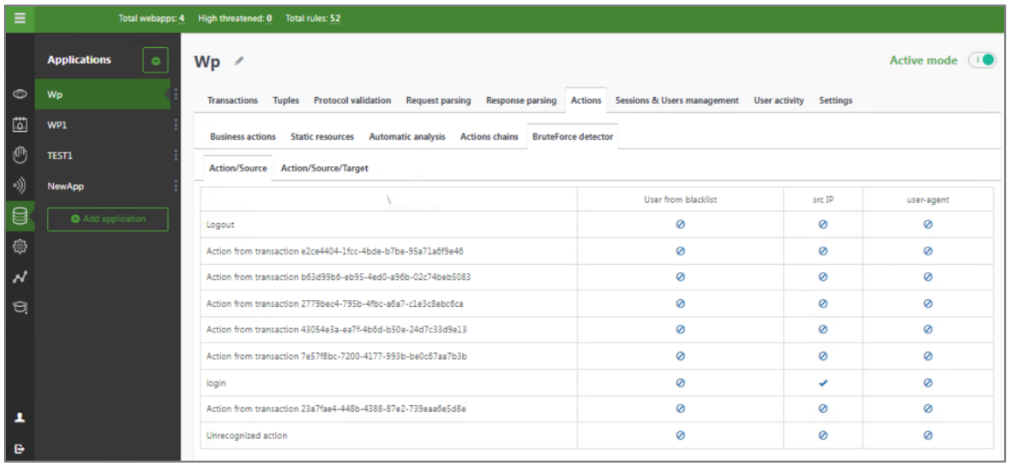


You can edit and delete all actions and view their information but you cannot edit the **Critical action** parameter. You can track chains in the **Automatic analysis** tab of the **Actions** tab by task of the **Action chains mining** type.

Brute-force detector

A brute-force attack detector can detect brute-force attacks. For this purpose, the brute-force attack detector implements a token bucket algorithm. It provides a limit on the number of requests to the web application per unit time (rate limiting). Sources are the subjects for which restrictions on the number of requests per unit time are enforced.

An example of the **BruteForce detector** tab interface is shown in the figure below.



For detailed information on how to configure a brute force attack detector, see p. 68.

Sessions & Users management

A session is a mechanism implemented to distinguish one user's request from another. There is no such thing as a session tracking mechanism in the HTTP protocol itself, so cookies, request bodies, and headers are used to convey session information. The standard option is that when a client goes to an application, he is given a cookie that uniquely identifies him. This cookie is then sent with each request from that client. This cookie can have a set lifetime.

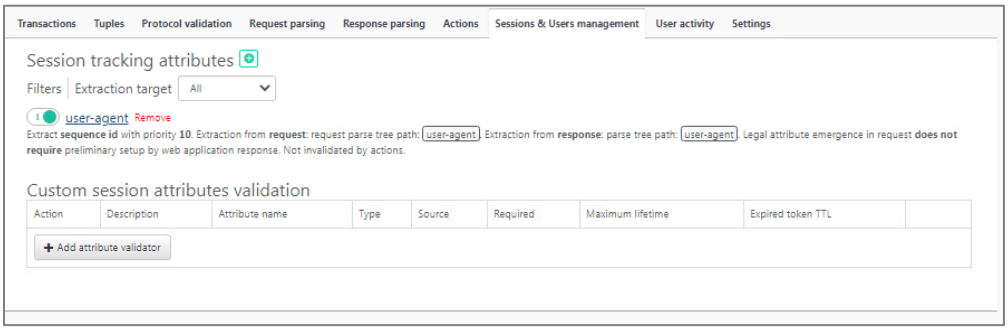
The session model is a mechanism that allows Continent WAF to bind to the mentioned attributes in order to track user sessions.

There are the following possible ways to use a session model:

- to restrict access to the profile for unauthorized users;
- to limit the number of sessions of one user by login account;

- to protect against theft of session IDs.

An example of the **Sessions & Users management** tab interface is shown in the figure below.



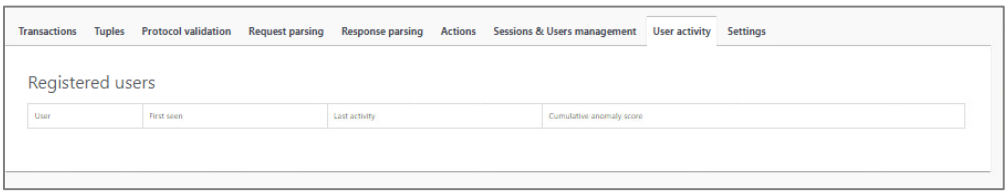
You can add session attributes in the tab. For detailed information on how to add a session attribute, see p. 71.

User activity tab

In the **Registered users** table of the **User activity** tab, you can view the user activity in the following columns:

- **User** — the value of the session identifier with the attribute type **Extract user name**;
- **First seen** — the date and time of the first appearance of the user session;
- **Last activity** — the date and time of the last appearance of the user session;
- **Cumulative anomaly score** — the number of cumulative anomalies for the last or active user session.

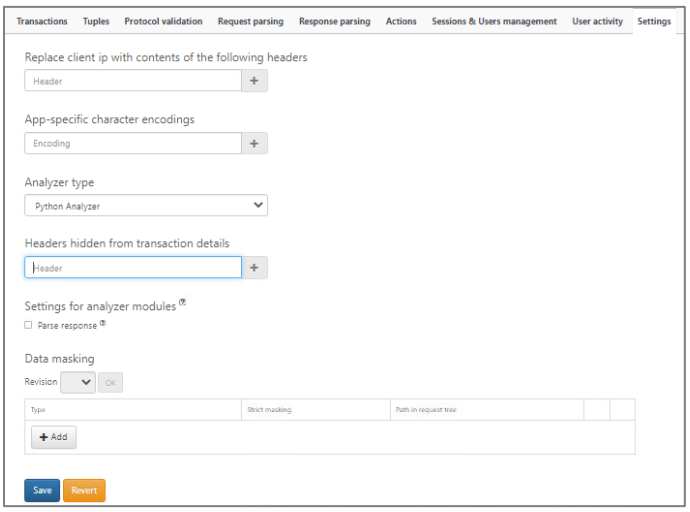
An example of the **User activity** tab interface is shown in the figure below.



Settings tab

In the **Settings** tab, you can configure application settings.

An example of the **Settings** tab interface is shown in the figure below.



This tab displays the following application settings:

- **Replace client ip with contents of the following headers** — the form for specifying the request header that contains the real IP address of the client. This setting is necessary if the WAF is preceded by a proxy, such as a load balancer or TLS server. The proxy substitutes the client IP address of the protected application for its own address. Proxies operating at layer 7 of the OSI model typically add a special header containing the client's IP address to client requests. For example, the x-real-ip header.
- **App-specific character encoding** — the form for specifying the specific character encoding used on the protected web application. If the application has specific character encodings and they are not specified in this form, query parsing errors occur, resulting in blocking by default.
- **Analyzer type** — selects the analyzer type. The analyzer type is specified in the Continent WAF setting and depends on the type of analyzer used.
- **Headers hidden from transaction details** — the form of specifying the headers that are not displayed when you view transactions.
- **Settings for analyzer modules** — settings are applied only if the corresponding modules are configured to use them.

  In this case, the **Parse response** setting is displayed. It is responsible for building the response parse tree based on the application. For detailed information on module settings, see [**1**].
- **Data masking** — a replacement of characters in selected fields. For detailed information on masking settings, see p. 56.

An administrator can delete applications in the **Settings** tab.

To add a header:

1. Specify information in the required field.
2. Click + to the right and click **Save**.

To delete a header from the list:

1. Click − to the right of the required record.
2. Click **Save**.

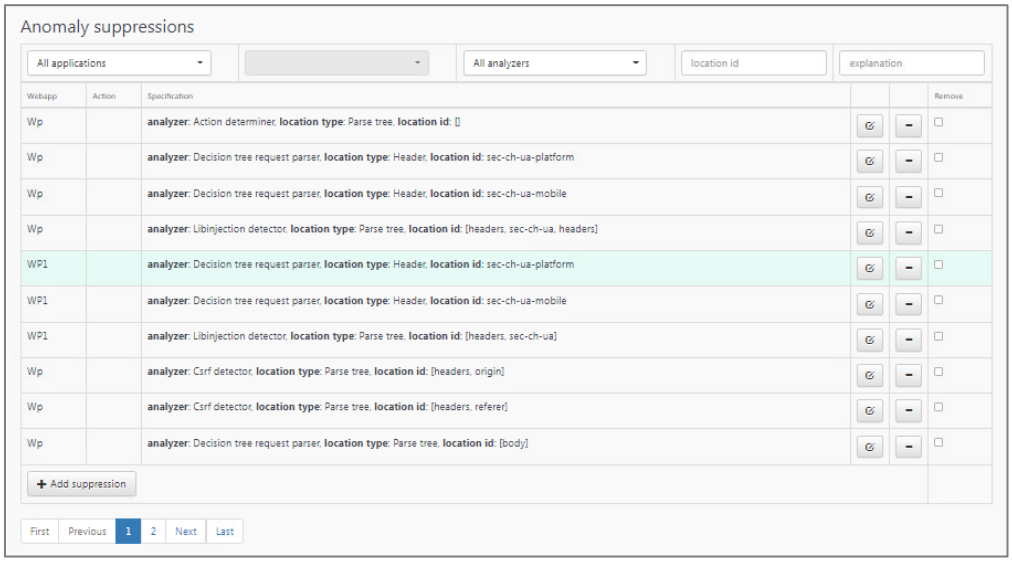
Settings section

To go to the **Settings** section, click  on the Navigation panel.

Analyst and operator interface

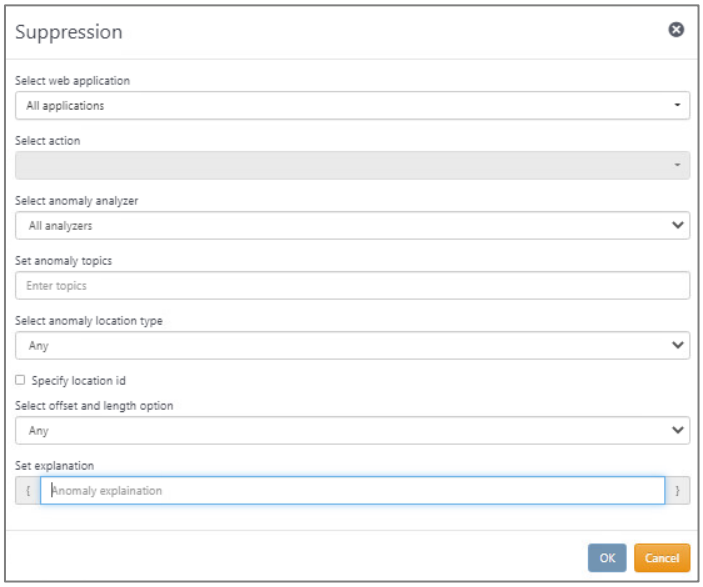
In the **Settings** section, the **Anomaly suppression** section functions are available for an analyst and operator.

An example of the **Anomaly suppression** section interface is shown in the figure below.



You can sort and find the required suppression by specifying parameters in the drop-down menus.

An analyst can add suppression by clicking **Add suppression**. The **Suppression** dialog box for adding a suppression is given in the figure below.



Administrator interface

This is an administrative interface. For the detailed description, see [**1**].

Reports section

In this section, you can generate security event reports for installation or tenant applications. Each report contains the following information:

- **Overview** — this section of the report contains information in graphical form and contains the following subsections:
  - **Statuses** — the graph in the **Statuses** subsection shows the statistics of web server responses by HTTP status code classes. An HTTP status code is an integer of three decimal digits that indicates the result of a request and determines what actions to take next. The set of status codes is a standard and is described in the corresponding RFC documents. The first digit indicates the state class;
  - **Blocks** — this graph displays statistics on the decisions made with respect to the transactions received by the protected application. The possible solutions are: allow the transaction, block the request, block (rewrite) the response;

    - **Delay** — this graph shows the delay in sending a response by the protected application after receiving a request. Such statistics is important from an information security point of view as well. For example, peaks in the application latency graph may indicate attempts to find or exploit vulnerabilities that load the application in an unusual way;
    - **Hostility** — this graph displays an integral indicator of hostility, the value of which characterizes the degree of anomaly severity of the application traffic at a given time. The indicator is measured in percent and is calculated using the following formula: 1/2 * [(number of attacks for the last minute) / (number of transactions for the last minute) + (number of attacks for the last minute) / (maximum number of attacks per minute in the past)].
- **Total traffic statistics for the application** — includes general statistics about the application. It includes information about the distribution of transactions by geographic location of sources, by anomaly type, and by application actions;
- **Security events** — includes information about security events with the number of transactions included in them.

In this section, you can generate one-time or periodic PDF reports containing information about system operation for a reporting period.

You can specify the following parameters for a report generation:

- reporting period (from, to);
- one or several protected applications;
- the report type:
    - **Brief** — only the information described above is included;
    - **Full** — in addition to the information in the summary report, more detailed statistics on security events detected by the system based on response rules. Each security event includes one or more response rule triggers of the same type. The period for which the same types of triggers are grouped into a single event is determined automatically by the system. The event can also be manually closed by the administrator. In this case, it is moved to the archive, and similar triggers will form a new event.

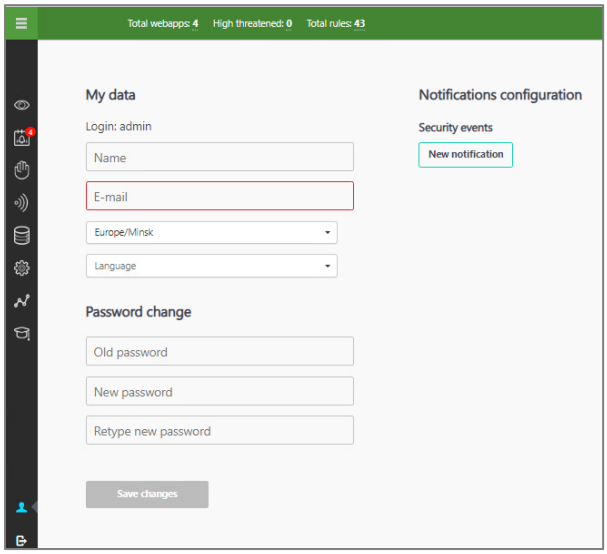
      The following information is provided for each event:

        - Event name. For example, it can be the name of a rule that triggers this event;
        - Level. Event threat level: **Info**, **Low**, **Medium**, **High** or **Critical**;
        - Period (event lifetime), during which similar actions are related to the given event;
        - Number of transactions. The number of transactions that are registered in this event (how many times the rule was triggered);
        - Statistics by event source (IP addresses, autonomous systems, countries, users);
        - Anomaly statistics (by types of anomalies and triggered analyzers);
        - Statistics on destinations (actions) in the application.

User settings and Log out buttons

Click  at the bottom of the Navigation panel to open a window with your account details. You can change account information, set a new password and customize notifications in this section. The view of the section is shown in the figure below.

An example of the **Station logs** panel interface is shown in the figure below.

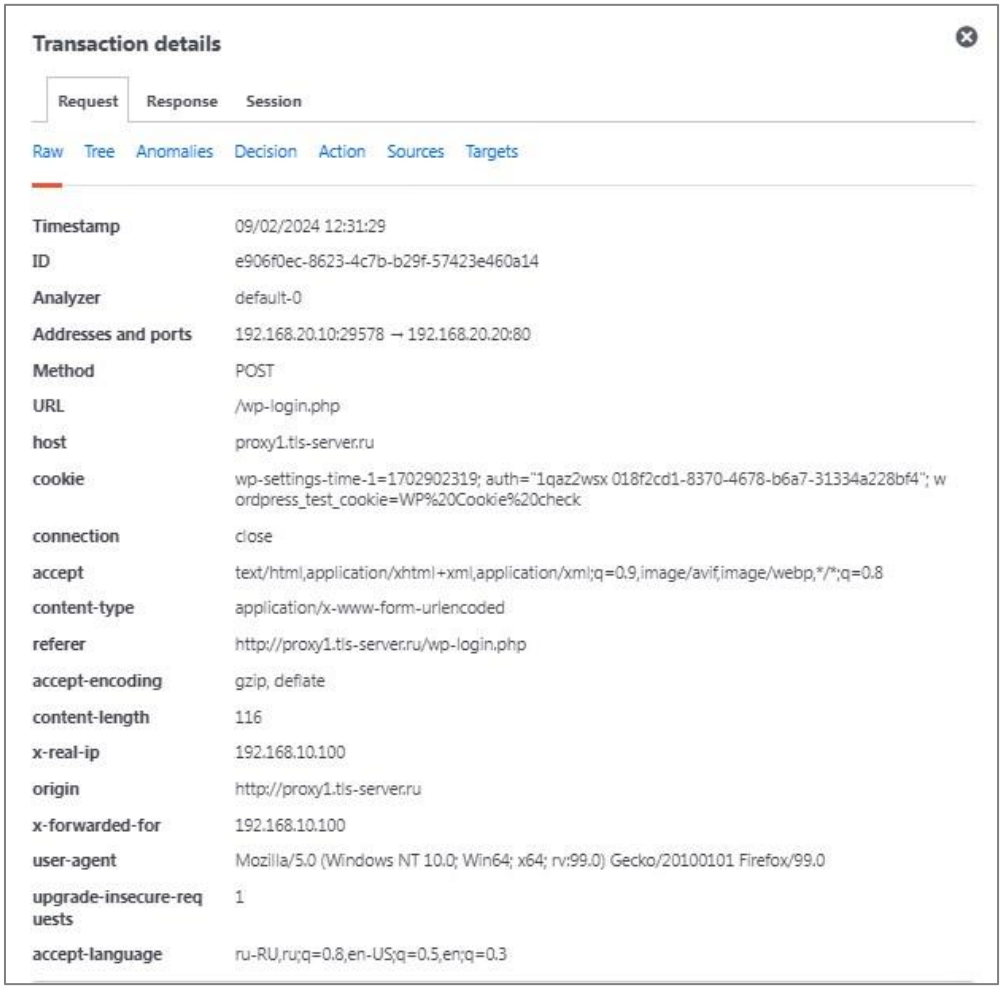


Click  at the bottom of the Navigation panel to exit the console.

Chapter 3
Functions and settings of Continent WAF elements

View transactions

If you double-click a transaction in the list, the **Transaction details** dialog box appears as in the figure below.



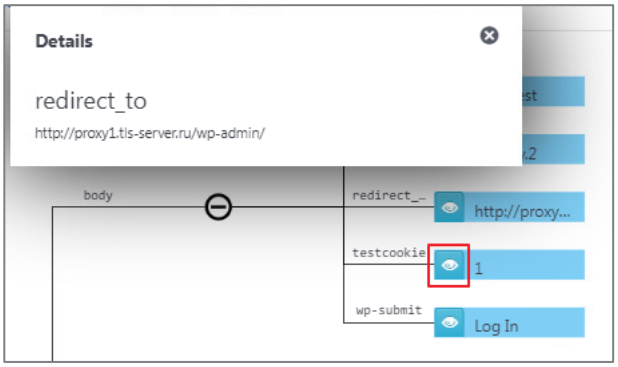
The **Transaction details** dialog box contains the following three tabs:

- **Request**;
- **Response**;
- **Session**.

Request tab

This tab contains detailed information about the request. The **Request** tab contains the following seven subtabs:

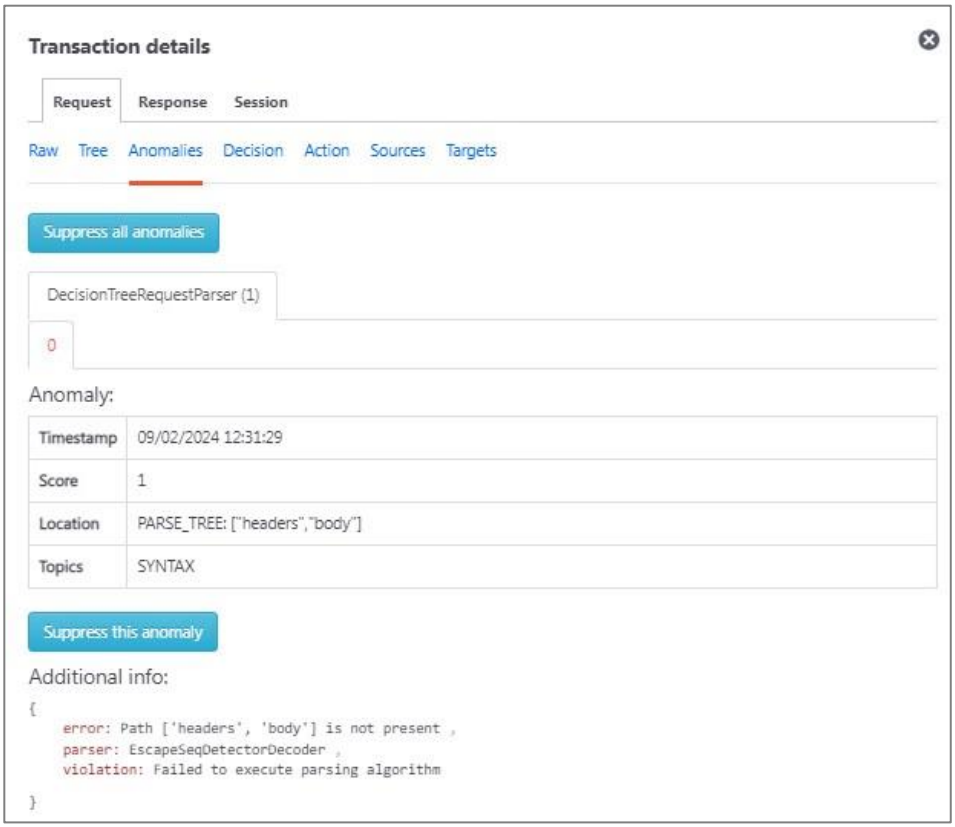
- **Raw** — request in text form.
- **Tree** — parsed request tree. This subtab content is created by the request parsing tree model. You can view the value for each tree element by clicking the respective button.



You can expand and collapse the tree elements using the ⊕ and ⊖ buttons respectively. You can also additionally parse each tree element. For settings of the parsing tree model, see p. 51.

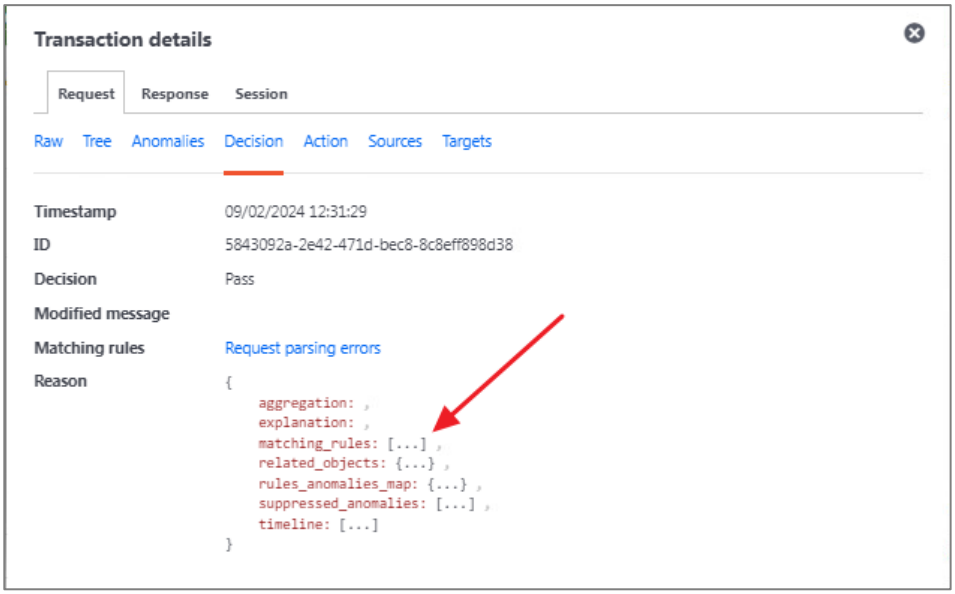
- **Anomalies** — contains a list of anomalies that were detected in this transaction. This subtab also displays additional information about the detected anomaly.

  In case of a false positive response you can suppress the detected anomaly by clicking **Suppress this anomaly**, if there are several anomalies — by clicking **Suppress all anomalies**. The **Suppression** dialog box appears. It allows you to edit the properties of the suppressed anomaly. If you suppress a list of anomalies, the full list of suppressed anomalies appears. You can exclude each anomaly from the list by clicking [−] and edit an anomaly by clicking [✎].

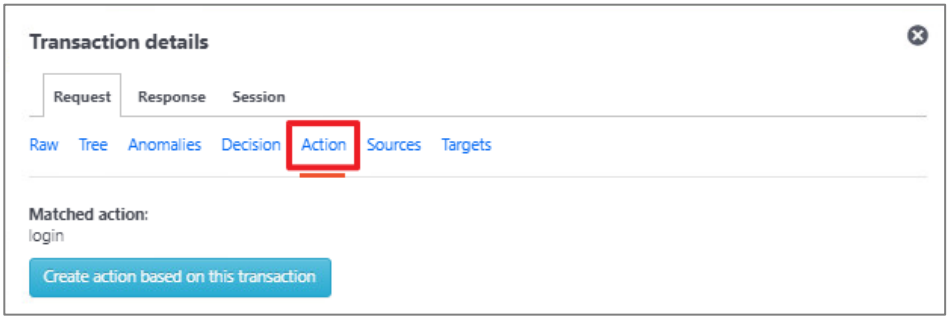


- **Decision** — contains data about the Continent WAF decision on the transaction, specifically the transaction timestamp, its ID, decision on whether to block or pass the transaction, modified message, matching rules and reason.

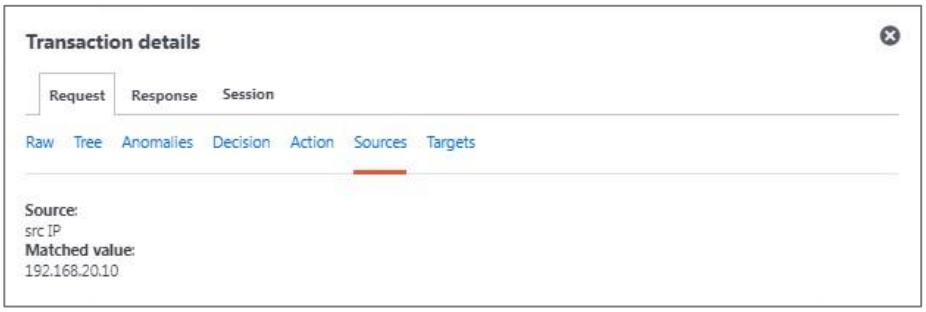
  In the **Reason** section, you can view details by clicking [...] in the required field.



- **Action** — contains the action and its parameters that matched this transaction. On this subtab, you can view the values of the action parameters that were specified during the action configuration. You can also semi-automatically create an action based on this transaction (for action settings, see p. 63).



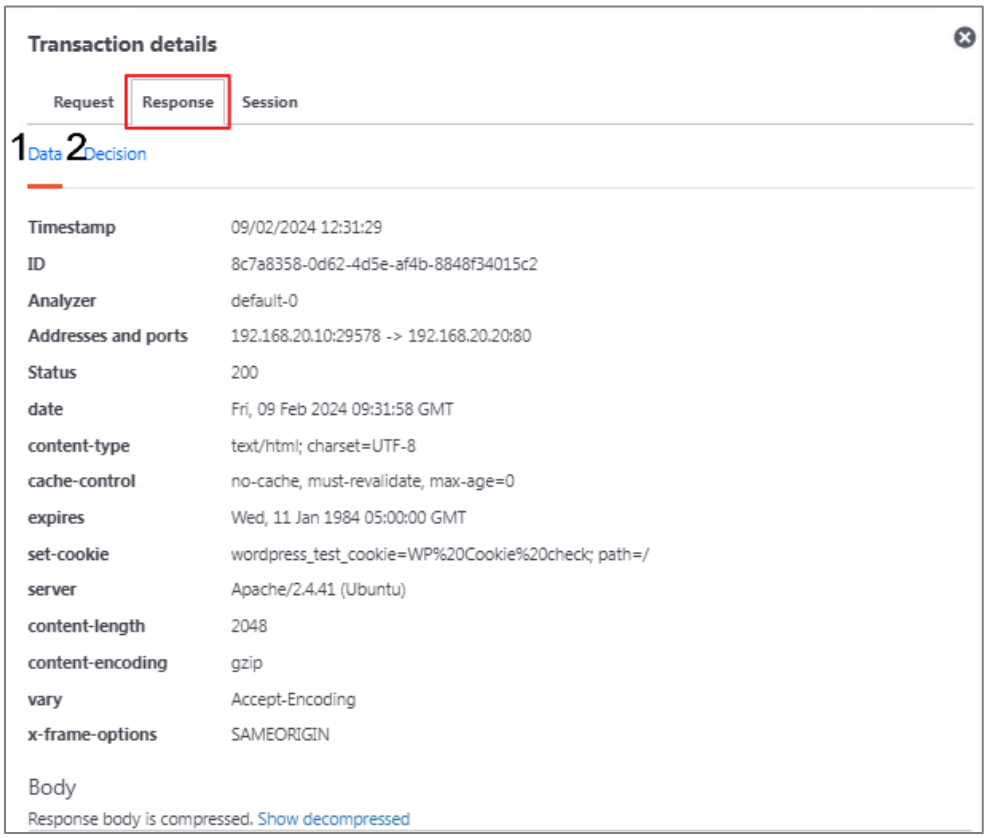
- **Sources** — contains found sources and its values for this transaction. For detailed information about the sources, see p. 63.



- **Targets** — contains found targets and its values for this transaction.

Response tab

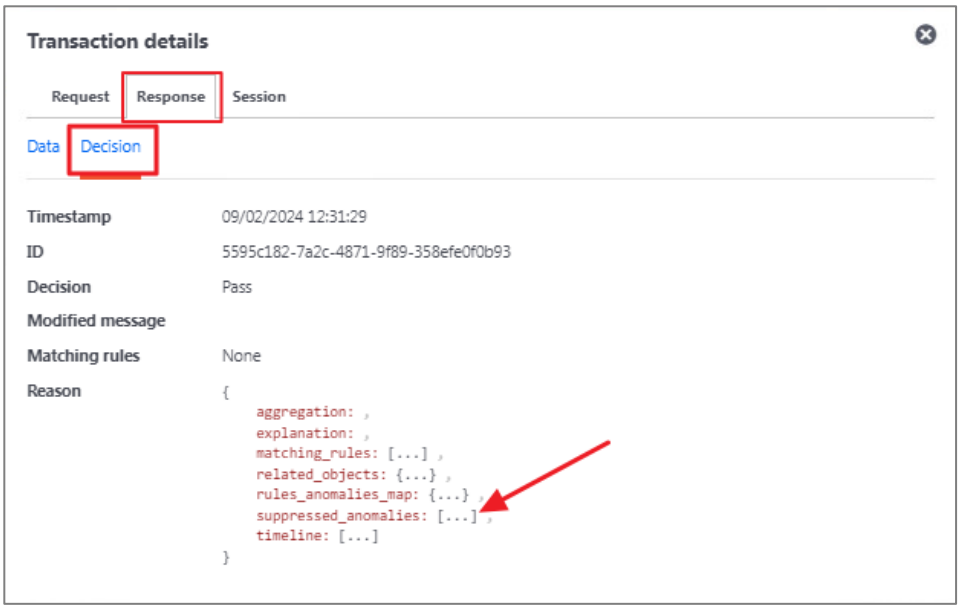
This tab contains information about the response.



The **Response** tab contains the following two subtabs:

- **Data** — contains response fields and its values.
- **Decision** — contains data about the Continent WAF transaction response decision, namely the transaction response timestamp, its ID, decision on whether to block or pass the transaction response (the decision of both the request and the response can block the transaction, for example, for an Open Redirect attack), modified message, matching rules and reason.

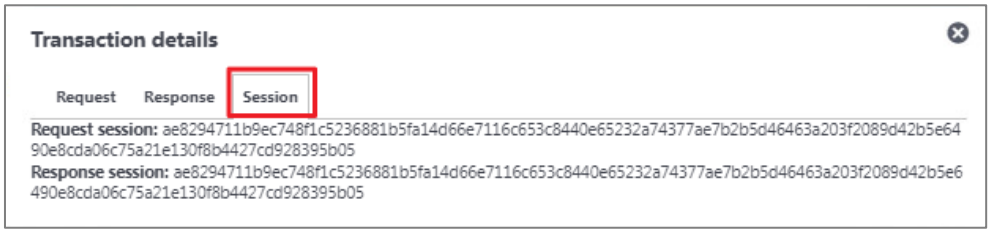
  In the **Reason** section, you can view details by clicking [...] in the required field.



Note. This tab may also contain the section with anomalies. For example, an anomaly from OpenRedirectDetector returns as a response.

Session tab

This tab contains information about the values of the request session and response session as well as the user. This tab appears only if the session model is configured (for session model configuration, see p. 71). The **User** field appears only if a session attribute with the **Extract user name** attribute type is configured.



Configure filtering requests to static resources

When a protected web application page loads, a set of requests to download files like icons, scripts and style sheets is performed. Continent WAF also displays the kind of requests that gets in the way of the correct traffic analysis.

You need to configure filtering requests to static resources to decrease the load of Continent WAF analyzers and remove uninformative traffic, because by default each transaction is placed into a common pool and is analyzed (a parsing tree is created, signatures are checked, etc.). To avoid analyzing transactions to static resources, the **Static resources** module on the **Actions** tab of the **Applications** section is used.

To pass transaction data on the nginx level, the **Static resources** module has the **Accelerated analysis of requests to static resources** toggle.

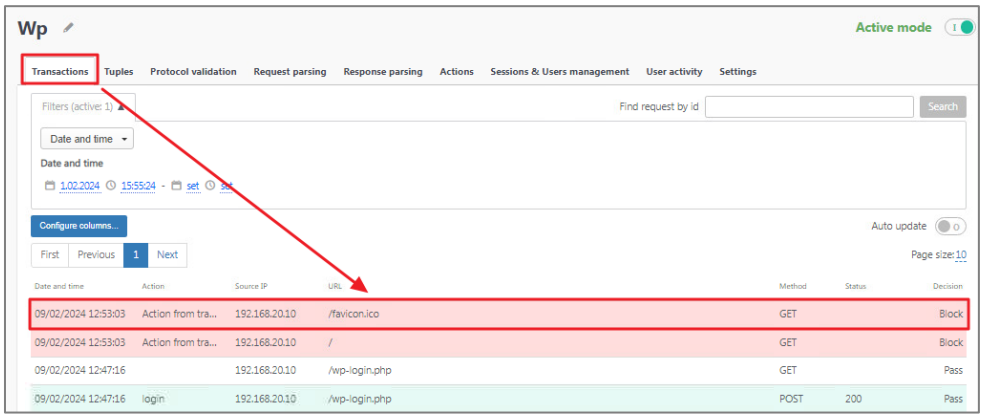
In Continent WAF, you can create static resources in the following three ways:

- semi-automatic creation of patterns for filtering requests to static resources;
- automatic creation of patterns for filtering requests to static resources;
- manual creation of patterns for filtering requests to static resources.

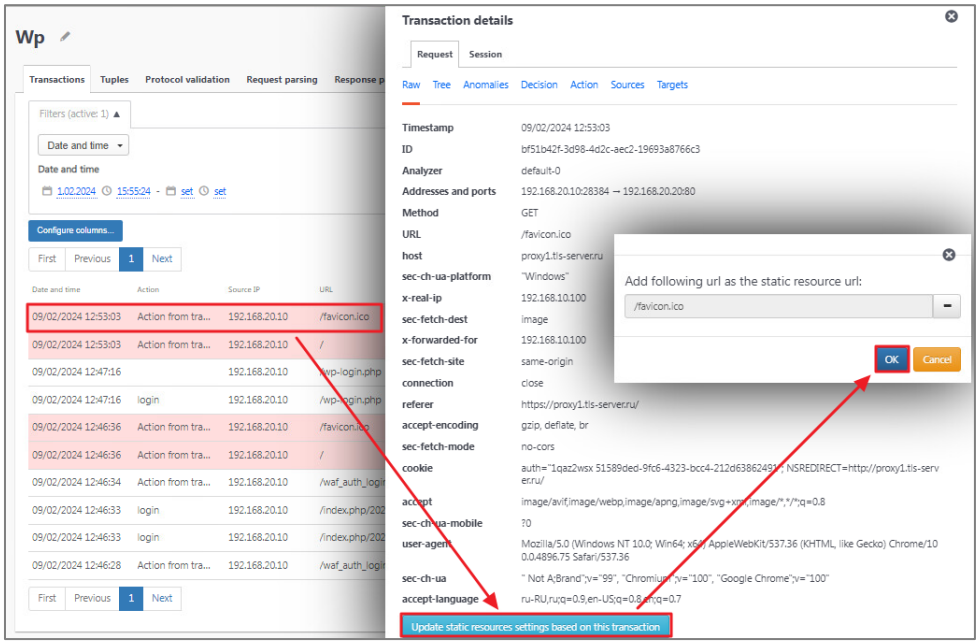
Semi-automatic creation of patterns for filtering requests to static resources

In Continent WAF, you can add files to static resources from the transaction menu one by one.

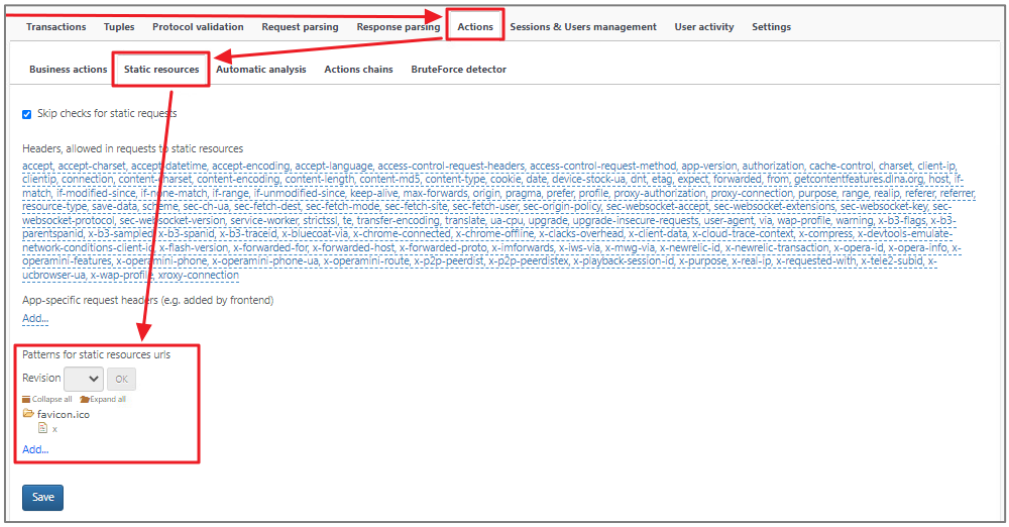
In the **Applications** section, on the **Transactions** tab, you need to find the transaction with the request to the static resource.



Double-click the transaction. Its description appears. Click **Update static resources settings based on this transaction**. In the appeared dialog box, click **OK**.

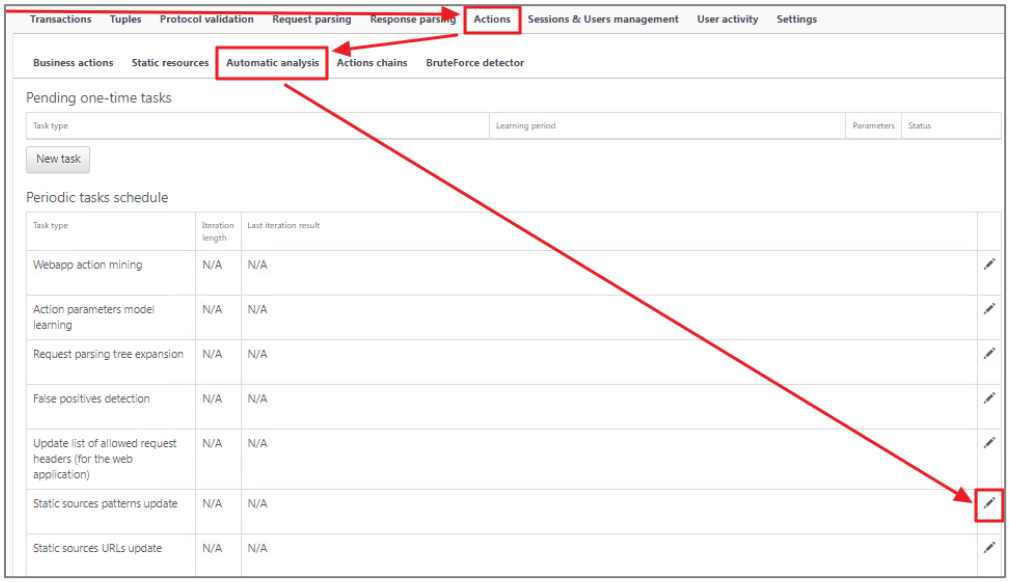


To check that the path to this file was added to the configuration of filtering requests to static resources, go to **Applications | Actions | Static resources**.



Automatic creation of patterns for filtering requests to static resources

Continent WAF can automatically add requests to static resources using machine learning. This module is enabled by default and can be additionally configured via the Continent WAF web interface. To do so, go to **Applications | Actions | Automatic analysis** and edit the **Static sources patterns update** task.



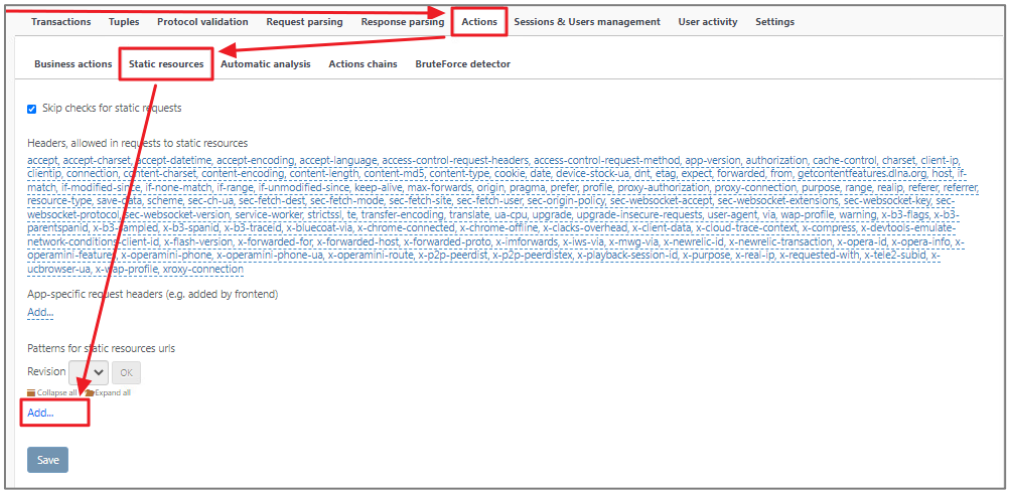
The appeared dialog box contains the following two fields:

- **Iteration length** — an interval over which transactions to static resources are considered;
- **Minimum unique users sampling length** — a threshold for the number of requests from unique (different) users upon reaching which the path to a static resource is recorded in the settings of filtering requests to static resources.

Default values are 10 and 100 respectively. The settings of filtering requests to static resources will change only if 100 different users have the same request to a static resource within 10 minutes.

Manual creation of patterns for filtering requests to static resources

In Continent WAF, you can add patterns for filtering requests to static resources manually. To add a path to a static resource manually, go to **Applications | Actions | Static resources** and click **Add**.

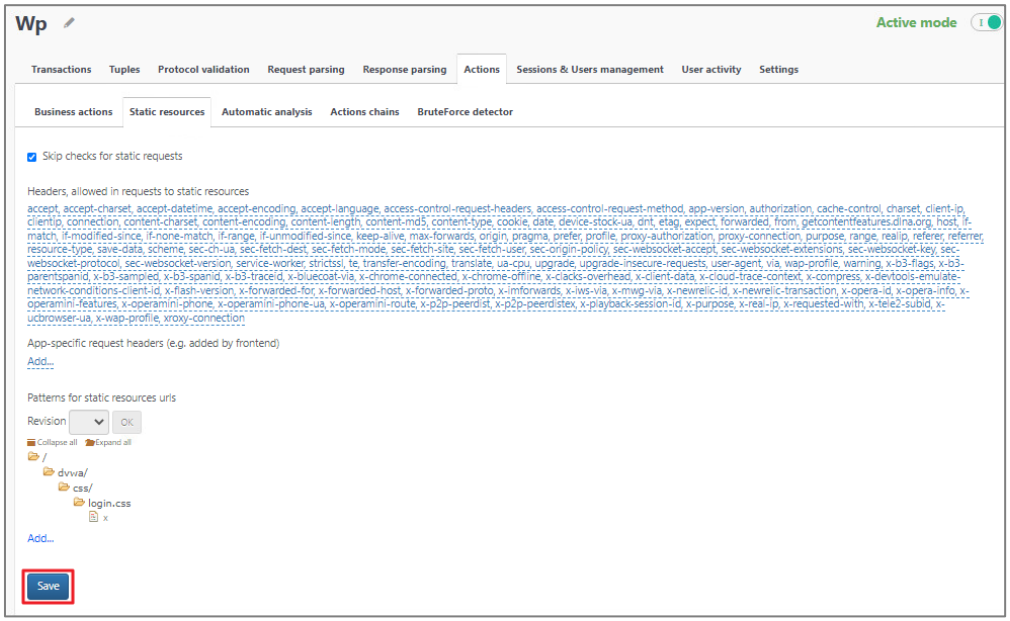


A dialog box for adding a pattern appears. You can add a full path to the file as well as specify its last part using a regular expression.

To add a full path to the file, enter the path to the file in the **New element** field, click **Add**, then click **Save**.

To add a path to the file using a regular expression, enter the path to the file in the **New element** field, click **Add**, then add a regular expression into the same field, select the **Regular expression** check box and click **Add**. Then, click **Save**. You can enter a regular expression right away without adding a common part. You can also add several regular expressions.

After adding a pattern, you need to save the changes on the **Static resources** subtab.

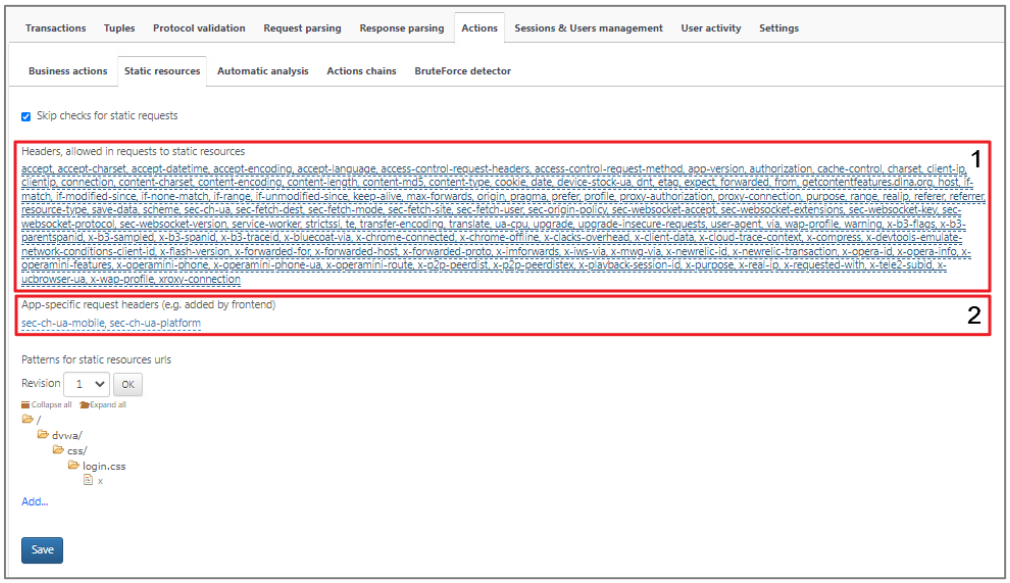


Note. Unlike automatic and semi-automatic adding, during manual adding of patterns for filtering requests to static resources, headers are not added to the **Headers, allowed in requests to static resources** field. Therefore, if requests to static resources contain headers that are not in this field, you need to add them manually. Requests to static resources with headers that are not in this list will be displayed even after the filtering pattern is created (even if these headers are specified in the protocol).

Add and remove headers allowed during filtering requests to static resources

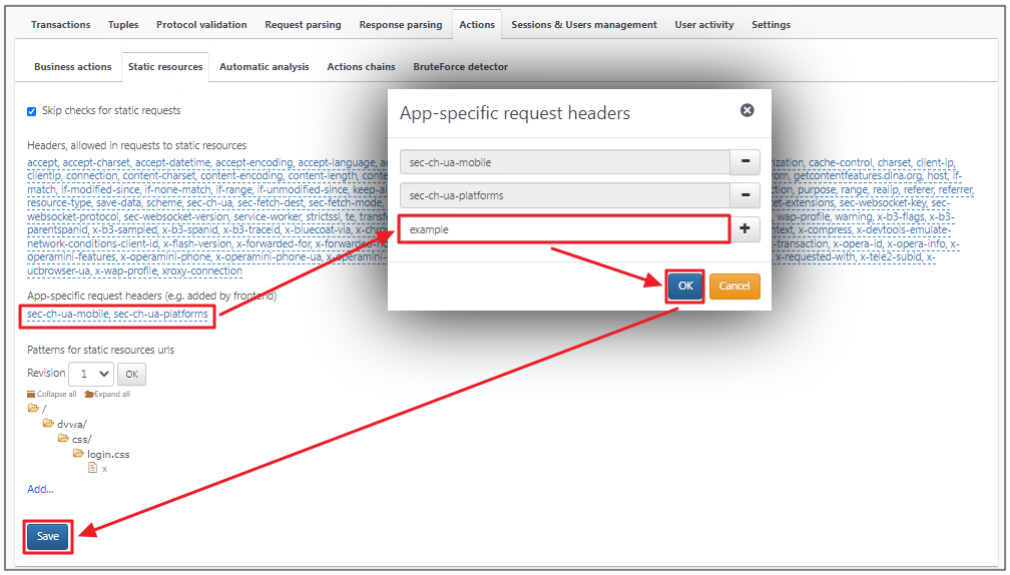
On the **Static resources** subtab, you can add headers allowed during filtering requests to static resources into the following two fields:

- **Headers, allowed in requests to static resources** — in this field you can specify headers that are allowed for use in requests to static resources and are used on all tenant or installation applications;
- **App-specific request headers (e.g. added by frontend)** — in this field you can specify headers that are allowed for use in requests to static resources and are used only on the selected app. By default, headers are added into this field during automatic and semi-automatic adding of configuration patterns for filtering requests to static resources.



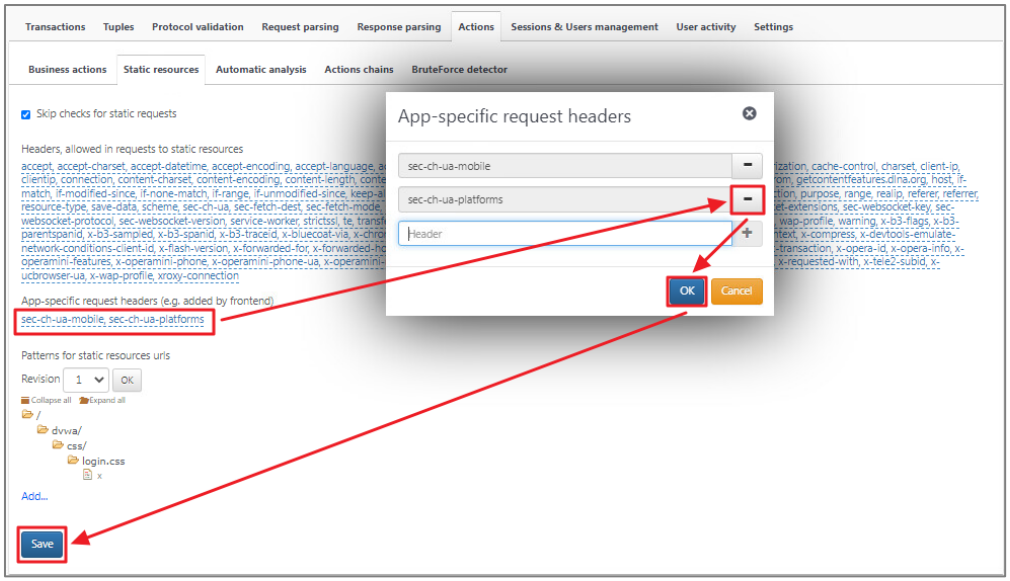
To add headers allowed in requests to static resources, select one of the two fields mentioned above and left-click its values. In the appeared dialog box, enter the

header name and click **OK**. After adding the header, you need to save the changes on the **Static resources** subtab.



To add an additional field for entering a header, click +.

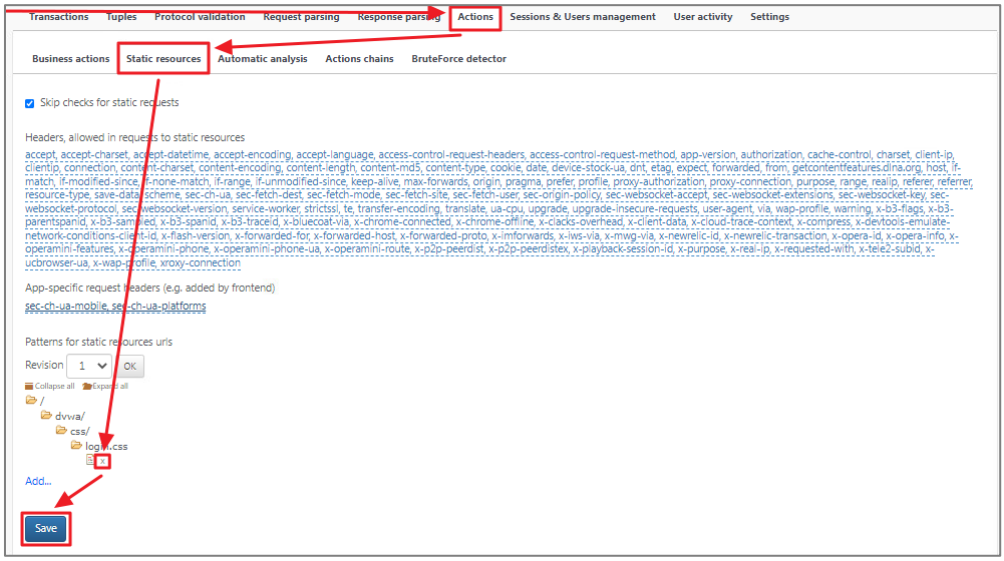
To remove headers allowed in requests to static resources, select one of the two fields mentioned above and left-click its values. In the appeared dialog box, click − in the field of the header you need to remove, and then click **OK**. After removing the header, you need to save the changes on the **Static resources** subtab.



Remove patterns for static resource addresses

In Continent WAF, you can remove patterns for filtering requests to static resources only manually.

To remove a path to a static resource, go to **Applications | Actions | Static resources** and click X next to the pattern you need to remove. After removing patterns for filtering requests to static resources, you need to save the changes on the **Static resources** subtab.



Revisions of patterns for static resource addresses

In Continent WAF, you can switch between revisions of this subtab configuration (configurations of all fields like **Headers, allowed in requests to static resources**, **App-specific request headers (e.g. added by frontend)**, **Patterns for static resources urls**). You can select the revision in the highlighted area in the figure below.

After selecting the revision different from the current one, you can click **OK** to view this revision configuration. To apply the selected revision, click **Revert**.

Configure protocol validation

To configure protocol validation, go to **Applications | Protocol validation**. The analyst can edit parameters by filling in the appearing dialog boxes, changing the state of check boxes and filling in the fields. For the changes, the following three operations are available: save the changes, show the changes and discard the changes. If you click **Show changes**, the **Difference** dialog box appears. This dialog box displays all changes as code.

Description of protocol validation parameter fields

You can see the description of protocol validation parameter fields in the table below.

Field	Description	Parameters	
Allowed header name regexp	Regular expression that each transaction header must match	A field for entering a regular expression	
Header validators	Check whether header values match specified parameters	The **Content-type validator** check box enables or disables checking whether the Content-type header matches the specified parameters	**Allowed media types** — you can specify a list of allowed MIME types (Multipurpose Internet Mail Extensions) that describe the nature and format of a document, file or set of bytes. MIME types are defined according to the RFC 6838 specification

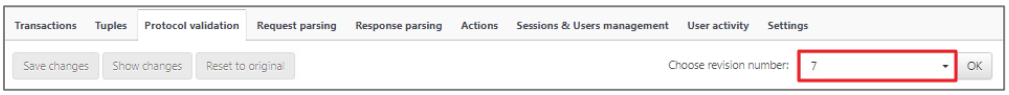
Field	Description	Parameters	
			Allowed parameters values — a field for entering a regular expression that the Content-type header parameter values must match
			Allowed parameters names — a field for entering names allowed for the Content-type header parameters
			Allowed encodings — a field for entering allowed encodings of the Content-type header
		The **Content-length validator** check box enables or disables checking whether the body header length matches the length specified in the Content-length header	
		The **Transfer-encoding validator** check box enables or disables checking whether the Transfer-encoding header matches the specified parameters	**Allow chunked requests** — when you click this text, a check box appears. This check box allows or prohibits chunked text transmission
			Allow request compression — when you click this text, a check box appears. This check box allows or prohibits request compression
		The **Host validator** check box enables or disables checking whether the host header matches the specified parameters	**Allowed hostname regexp** — a field for entering a regular expression
		The **Range validator** check box enables or disables checking whether the Range header matches the specified parameters	**Maximum amount of range intervals** — a field for entering a maximum amount of intervals in the Range header (for example, if you specify 2 as the maximum interval amount in the Range header, the transaction with the Range: bytes=200-1000, 2000-6576, 19000- header will be blocked)
Maximum header length	A field for entering maximum header length	A field for entering a numeric value of maximum header length (if at least one of the transaction headers is longer than this value, the transaction will be blocked)	
Request_line validator parameters	Check whether url, method header values match specified parameters	**Allow 'OPTIONS *' requests** — when you click this text, a check box appears. This check box allows or prohibits requests with the OPTIONS method	
		Allowed path extra chars — a field for entering characters allowed in a header in addition to the standard set	
		Maximum query length (0 - unlimited) — a field for entering a numeric value of maximum length of URL header parameters	
		Don't allow urls in absolute form — when you click this text, a check box appears. This check box allows or prohibits requests in absolute form (URL in absolute form: **scheme://server/path/resource**, where **scheme** is a website access scheme; **server** is a name of the computer on which the resource is located; **path** is a sequence of directories leading to the target object; **resource** is usually a file name)	
		Allow invalid percent-encoding — when you click this text, a check box appears. This check box allows or prohibits requests with incorrect URL encoding (for example, if you allow requests with incorrect URL encoding, transactions with characters not encoded in UTF-8 will be passed)	

Field	Description	Parameters
		Allowed methods — a field for entering the names of allowed methods
		Maximum path length (0 - unlimited) — a field for entering a part of the path request
		Allowed query extra chars — a field for entering characters allowed in URL parameters in addition to the standard set
Replace certain chars with dash (-) in header names	A check box that enables forced replacement of characters specified in the **Characters to be replaced with dash (-) in header names** field	
Characters to be replaced with dash (-) in header names	A field for entering characters you need to replace with dash. This field appears if the **Replace certain chars with dash (-) in header names** check box is selected	
Allowed headers	A field for entering headers allowed in requests	
Allowed Header Patterns	A field for entering regular expressions for specifying headers allowed in requests	
Allowed headers values regexp	A field for entering a regular expression that header values must match	
Allowed duplicate headers	A field for entering headers allowed to be duplicated in requests	
Maximum headers count	A field for entering a maximum number of headers in a request	

Note. You can save the changes to protocol validation both for application and for the whole tenant/installation. To do so, select the respective item in the dialog box that appears after you click **Save changes**.

Revisions of protocol validation configurations

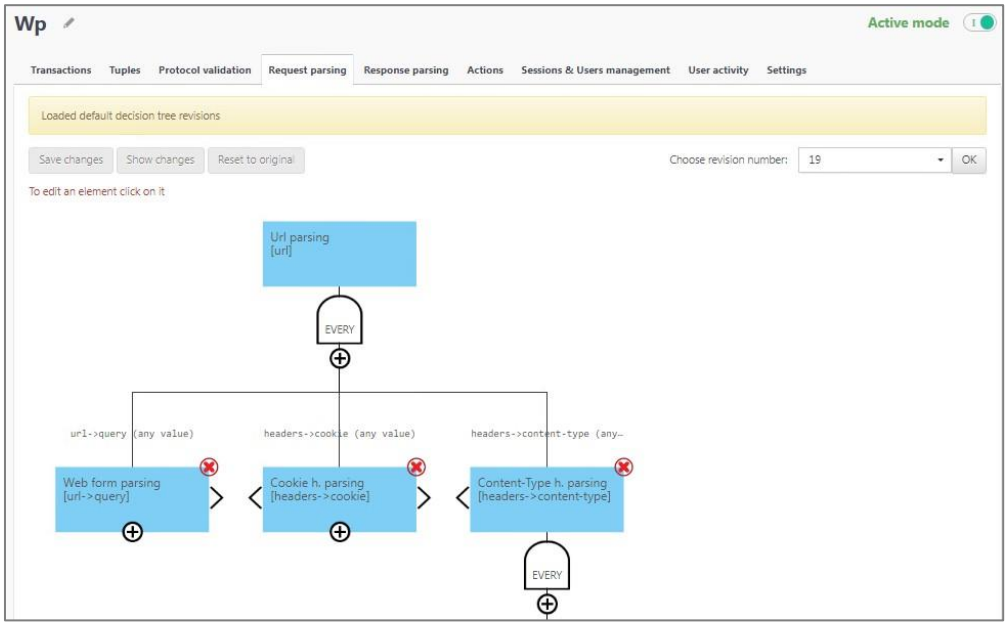
In Continent WAF, you can switch between revisions of this tab configuration. You can select the revision in the highlighted area in the figure below.



After selecting the revision different from the current one you can click **OK** to view this revision configuration. To apply the selected revision, click **Revert**.

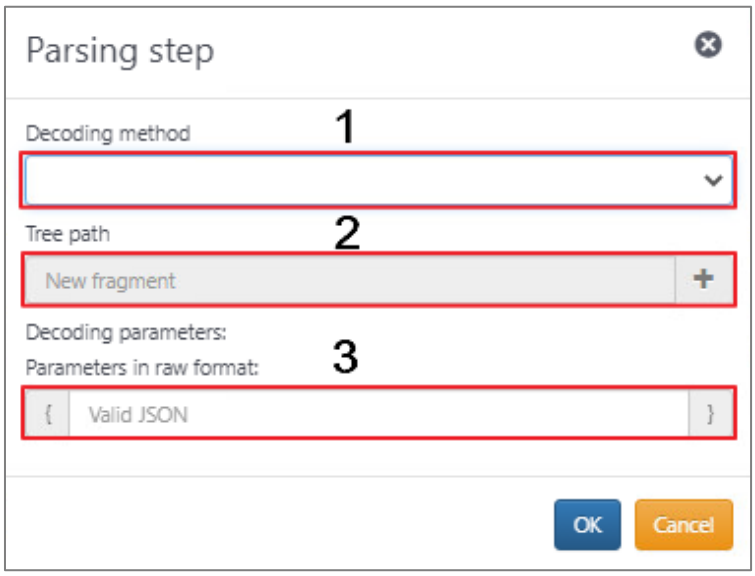
Configure parsing tree

In Continent WAF, you can expand the request parsing tree. To do so, go to **Applications | Request parsing**.



Add and remove decoding blocks

To expand the parsing tree, in the required part of the parsing tree, click ⊕. A dialog box for creating a parsing step appears.



You need to specify the following fields:

1. **Decoding method** — a method using which the value will be parsed. You can see the description of methods in the table below.

Field	Description
Deflate decompression	Decompression of data compressed using the deflate compression algorithm. You do not have to specify additional parameters in field 3
GZip decompression	Decompression of data compressed using the GZIP compression algorithm. You do not have to specify additional parameters in field 3
Random value marker	This parser disables the automatic task of updating/expanding the parsing tree for the node value specified in field 2. You do not have to specify additional parameters in field 3
Web form parsing (x-www-form-urlencode)	Decodes values in tuples with a key separated by '&', with '=' between key and value (content-type:application/x-www-form-urlencoded). You do not have to specify additional parameters in field 3
Content-Type header parsing	Content-type header parsing. Required for the correct operation of the protocol validation module, specifically checking allowed MIME types. You do not have to specify additional parameters in field 3

Field	Description
Cookie header parsing	Parses the cookie array contained in the Cookie header into separate elements. Used only for expanding the request parsing tree. You do not have to specify additional parameters in field 3
Set-Cookie header parsing	Parses the cookie array contained in the Set-Cookie header into separate elements. Used only for expanding the response parsing tree. You do not have to specify additional parameters in field 3
Static prefix parsing	Parses values located at the path specified in field 2 using the specified prefix and builds the parsing tree. This decoding method will not work if you do not specify the parameter in field 3. Example of a parameter in raw form: **"prefix":""**, you need to enter the prefix value in quotes after the **":"** sign
Regex pattern matching and group extraction	Splits the path value specified in field 2 using a regular expression (pattern) into a specified number of groups (group_names) and builds the parsing tree. The number of groups in pattern must match the number of names in group_names. Example of a parameter in raw form: **"group_names":["group1","group2"]**,**"pattern":"(\\d+)(\\d+)"**
Multipart parsing with automatic boundary selection	Parses the request body (needs to be used only for POST requests) with automatic delimiter detection and builds the parsing tree. For correct operation, do not specify the path value in field 2. You do not have to specify additional parameters in field 3
Multipart parsing with custom separator	Separates the path value specified in field 2 using the specified boundary and builds the parsing tree. This decoding method will not work if you do not specify the parameter in field 3. Example of a parameter in raw form: **"boundary":""**, you need to enter the value of the boundary using which the value will be separated in quotes after the **":"** sign
Url parsing	Serves as a URL decoder. Can be used without additional parameters in field 3 or with the following parameters: • **default_scheme** — scheme used in the URL; • **normalize_path** — whether the decoder needs to normalize the path (collapse redundant delimiters, links to a higher level, etc.); • **decode_path_parameters** — whether the decoder needs to decode path parameters. By default: **"default_scheme":"b''",**"normalize_path":"False"**,"decode_path_parameters":"False"**
PHP serialized object	Decodes a serialized PHP object. Builds the parsing tree. This decoding method will not work if you do not specify the parameter in field 3
Escape sequence decoding (\xaf \n etc)	Removes slash escaping based on the specified parameters: • **u_escape** — a Unicode escape starting with a backslash, a lowercase u and no more than 4 hexadecimal digits after; • **U_escape** — a Unicode escape starting with a backslash, an uppercase U and no more than 8 hexadecimal digits (the resulting code point must be less than MAX_UNICODE); • **octal_escape** — a backslash followed by 1 to 3 octal digits; • **hex_escape** — a backslash followed by x and no more than 2 hexadecimal digits; • **literal_escape** — a backslash followed by a character; • **CSS_escape** — a backslash followed by 1 to 6 hexadecimal digits. Each parameter can take only two values: true and false. Example of a parameter in raw form: **"CSS_escape":"true"**
Base16	Serves as a Base16 decoder. You do not have to specify additional parameters in field 3
Base32	Serves as a Base32 decoder. You do not have to specify additional parameters in field 3
Base64	Serves as a Base64 decoder. You do not have to specify additional parameters in field 3
Base64-url	Serves as a Base64-url decoder. You do not have to specify additional parameters in field 3
CSV	Separates a CSV format value. Can be used without additional parameters in field 3. You can also use the parameters separately: • **Delimiter** — separates the value using the specified character;

Field	Description
	• **Doublequote** — controls the processing of quotes in fields. If the value is **True**, two consecutive quotes are interpreted as one during reading, and each quote character embedded in the data is written as two quotes during writing; • **Escapechar** — refers to a one-character string used to escape the delimiter when you set a value for quoting; • **Quotechar** — refers to a one-character string that will be used to quote values if special characters appear in the field; • **Skipinitialspace** — determines the interpretation of a space after the delimiter. If the value is True, initial spaces will be removed. The default value is **False**. By default: **"delimiter":",", "doublequote":"True", "escapechar":"None", "quotechar":"""", "skipinitialspace":"False"**
DSV (values separated by arbitrary delimiter)	Separates the value using the two characters specified in field 3. By default: **"delimiters":",",":"**
GraphQL	GraphQL parser. Parses GraphQL into parameters and builds the parsing tree. You do not have to specify additional parameters in field 3
GWT RPC	GWT RPC parser. Parses GWT RPC into parameters and builds the parsing tree. You do not have to specify additional parameters in field 3
HTML	HTML parser. Parses HTML into parameters and builds the parsing tree. You do not have to specify additional parameters in field 3
HTML entities	HTML entity parser. Parses an HTML entity into parameters and builds the parsing tree. You do not have to specify additional parameters in field 3
JSON	JSON parser. Parses JSON into parameters and builds the parsing tree. You do not have to specify additional parameters in field 3
JSON RPC	JSON RPC parser. Parses JSON RPC into parameters and builds the parsing tree. You do not have to specify additional parameters in field 3
JSONP	JSONP parser. Parses JSONP into parameters and builds the parsing tree. You do not have to specify additional parameters in field 3
SOAP	SOAP parser. Parses SOAP into parameters and builds the parsing tree. You do not have to specify additional parameters in field 3
Url-decode	Serves as a URL decoder. You do not have to specify additional parameters in field 3
XML	XML parser. Parses XML into parameters and builds the parsing tree. You do not have to specify additional parameters in field 3
XMLRPC	XML RPC parser. Parses XML RPC into parameters and builds the parsing tree. You do not have to specify additional parameters in field 3
YAML	YAML parser. Parses YAML into parameters and builds the parsing tree. You do not have to specify additional parameters in field 3

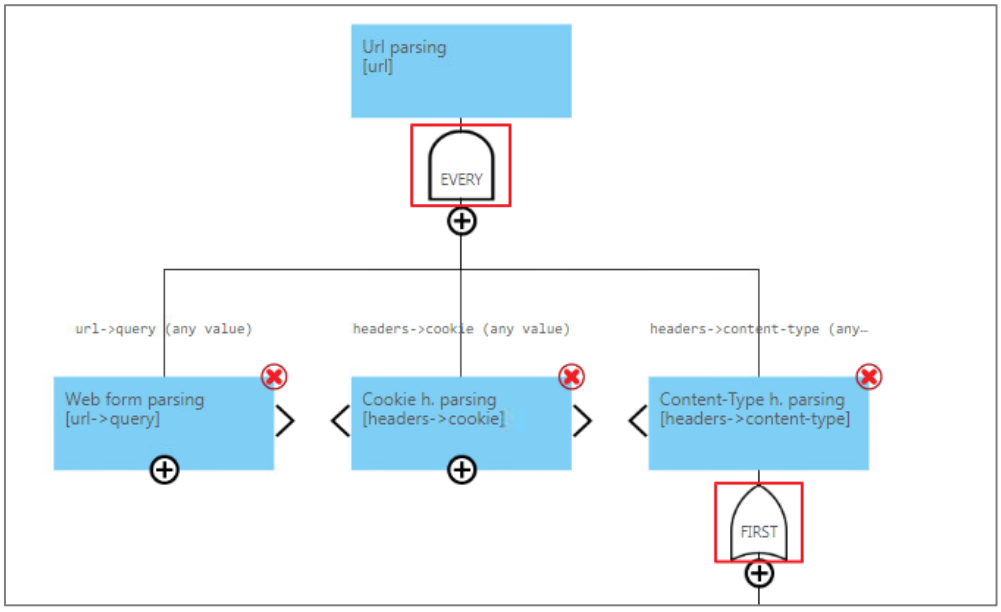
2. **Tree path** — a field for entering the path to the value that you need to be decode.
3. **Parameters in raw format** — a field for entering decoding parameters (not all decoding methods require these).

To remove a decoding block, click  in the top right corner of the block you need to remove.

Note. When you remove a block with another block (child) attached to it at the bottom, both the selected block and the child block will be removed.

Conditions for decoding blocks

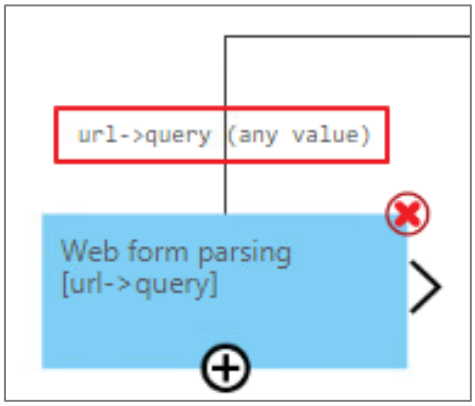
When you add a child block to a parent block, a condition appears between them. You can change a condition at any moment by left-clicking it.



The following two condition types are available:

1. **Every** — this condition means that all subsequent child decoding blocks must parse the value; if at least one child block does not parse the value, an anomaly is generated.
2. **First** — this condition means that any subsequent child decoding block must parse the value; if no child blocks parse the value, an anomaly is generated.

Each block has its predicate above it. You can specify it by clicking the highlighted area in the figure below.



The dialog box for specifying a predicate appears. To change the predicate, you need to click the highlighted area in the figure below.

The following four predicates are available:

1. **default** — when you select this predicate, this decoding block is considered to be always working.

2. **regexp** — when you select this predicate, you need to enter the path to the value and the regular expression that the value must match. This decoding block will work only if the specified condition is met.
3. **value** — when you select this predicate, you need to enter the path to the value and its exact value. This decoding block will work only if the specified condition is met.
4. **paths present** — when you select this predicate, you need to enter the path. This decoding block will work if this path is present in the request/response (depending on which parsing tree is expanded).

Note. Parent block predicates are prioritized; if they are not met, the child blocks do not work either.

Revisions of parsing tree configurations

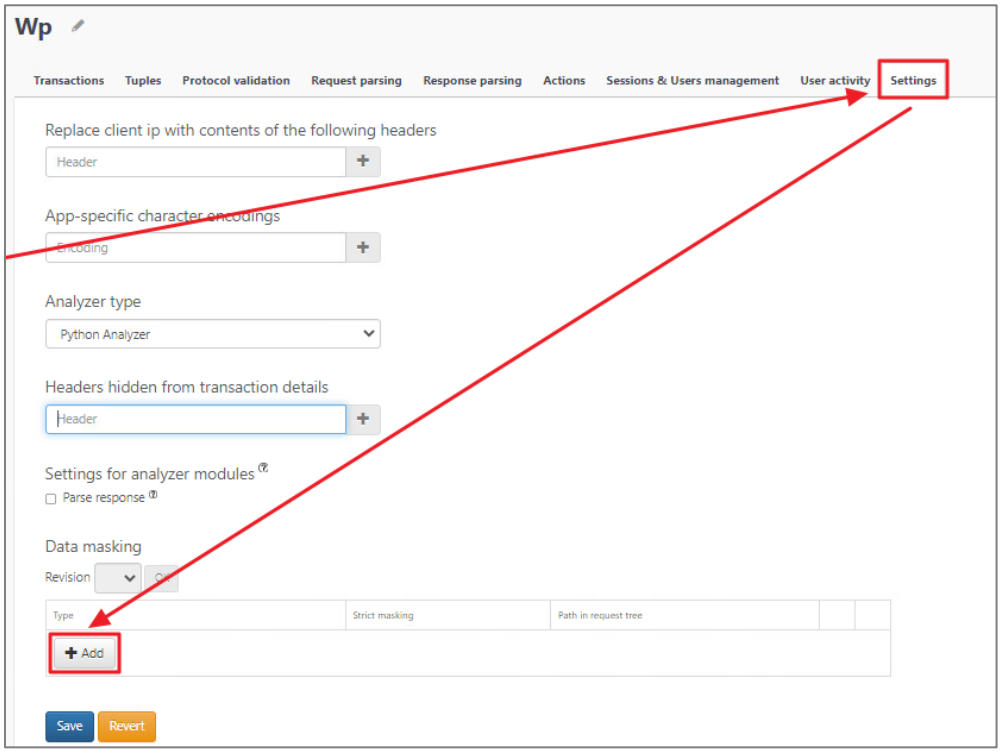
In Continent WAF, you can switch between revisions of this tab configuration. You can select the revision in the highlighted area in the figure below.

After selecting the revision different from the current one, you can click **OK** to view this revision configuration. To apply the selected revision, click **Revert**.

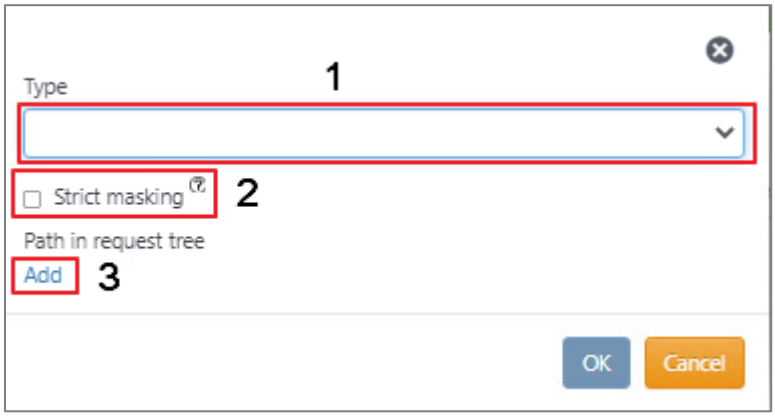
Masking

In Continent WAF, you can mask the fields (changes the field display in the Continent WAF web interface).

To configure masking, go to **Applications | Settings** and, in the **Data masking** section, click **Add**.



A dialog box for creating a mask appears.



This dialog box contains the following parameters (see the figure above):

1. **Type** — for selecting a masking type from the drop-down list. You can see the description of types in the table below.

Type	Description
Smart replace (preserve characters type (digit, lowercase/uppercase ASCII))	Keeps the character type (digits, lowercase/uppercase ASCII) and replaces it with another random character
Replace characters with random bytes	Replaces the characters of the masked element with random bytes
Password (only printable characters)	Replaces the characters of the masked element with random ones
Number	Replaces only numbers with random numbers
CVC	Replaces triples of digits with random symbols
Credit Card Number	Replaces a string that matches the form of a credit card number with random digits

2. **Strict masking** — used for replacement with only one character of each type (456 -> 000, word -> aaaa).
3. **Path in request tree** — the parsing tree path to the element that you need to mask.

If the data in the parsing tree is not masked after you configure the masking, you need to check that masking is enabled in the analyzer settings.

Note. It is incorrect to configure masking for a URL or its part. This leads to incorrect transaction display in the web interface.

Working with lists

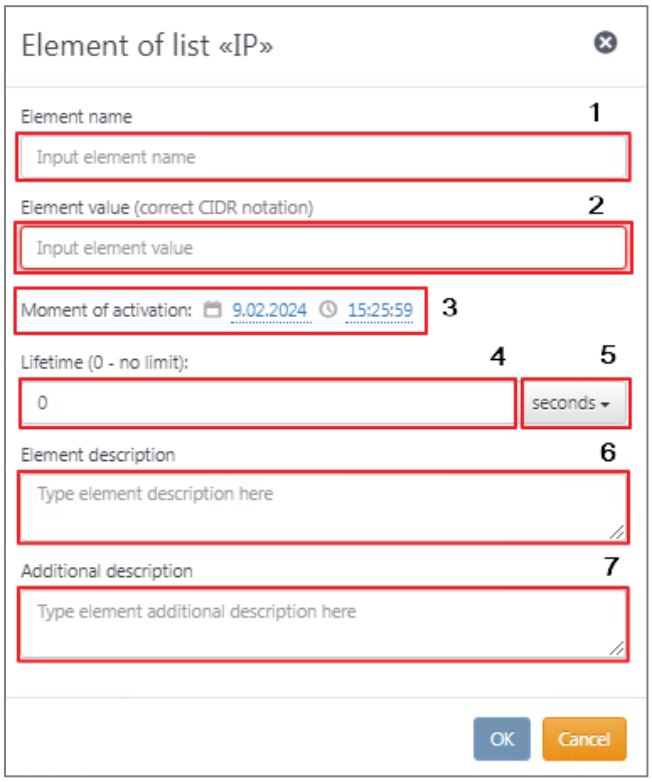
In Continent WAF, you can create lists that can be used in sources later.

Create a new list

To create a new list, go to **Sources and lists | Lists** and click **Add list**. In the appeared dialog box, specify the following parameters:

- **List name** — a field for entering a list name. We recommend giving lists explicit names as when you select the list in the source, you can see only the name;
- **Select web application** — for selecting an application from all installed on the tenant or installation;
- **Type** — for selecting list elements extension. The following three element types are available:
- **String**;
- **Integer**;
- **IP address** (both individual IP addresses and address ranges).

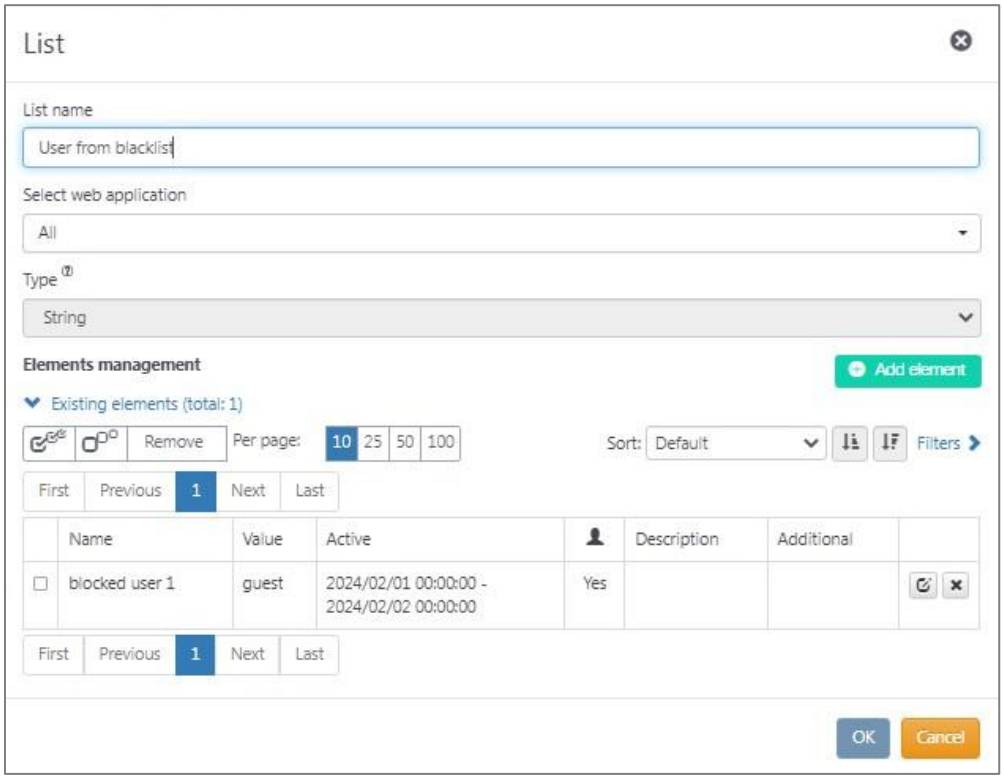
Next, add elements to the created list one by one by clicking **Add element**.



In the appeared dialog box, specify the following parameters:

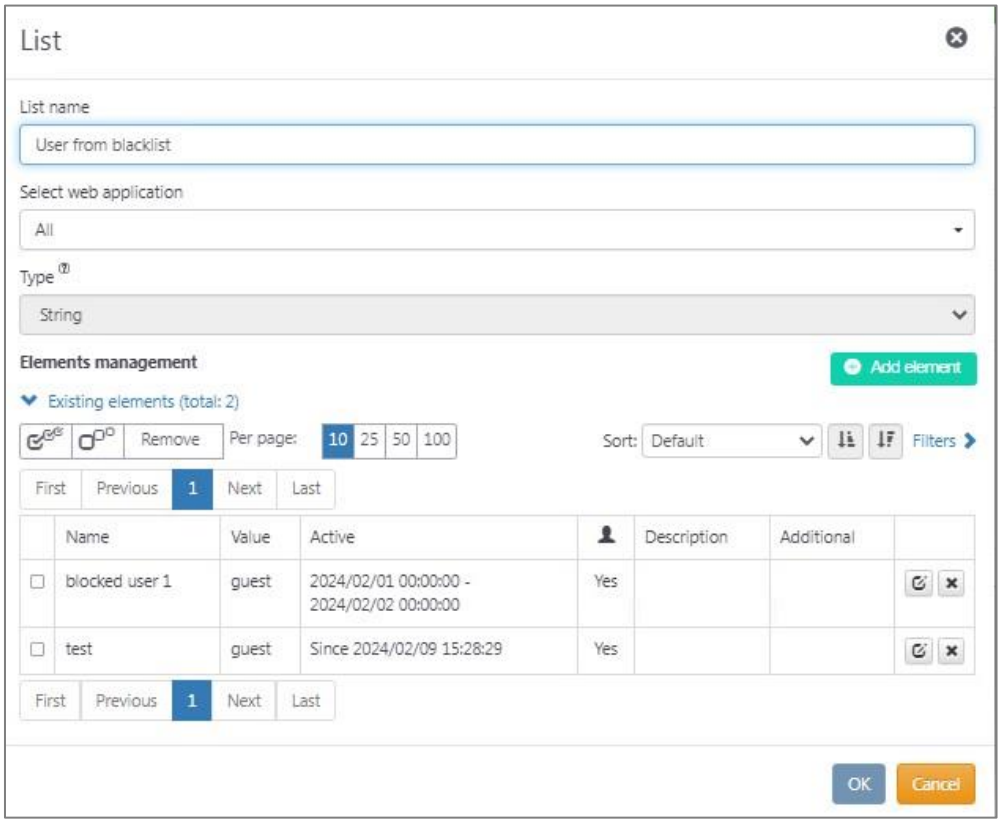
1. **Element name** — a field for entering an element name.
2. **Element value (correct CIDR notation)** — a field for entering an element value. This field checks whether the entered value matches the element type selected when creating the list.
3. **Moment of activation** — the date and time from which this list element starts working. After you select a date, the **Lifetime (0 - no limit)** field appears.
4. **Lifetime (0 - no limit)** — a field for entering a lifetime (you can enter fractional values). The 0 value means that the element lifetime is unlimited.
5. Select value — five options are available: **seconds**, **minutes**, **hours**, **days**, **weeks**.
6. **Element description** — a field for entering a description of this list element.
7. **Additional description** — an additional field for entering a description of this list element.

The dialog box of the created list changes its appearance based on the results of adding elements. Now it will display the new added elements that are in the list when you save it by clicking **OK**.

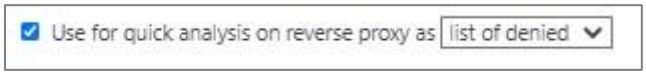


When you create and edit lists, you can configure the element display (see the figure below), for example, by selecting the sorting order, the number of elements displayed on the page, etc.

Besides, you can view the properties of list elements as well as edit individual elements and remove them.



If a web application is not selected in the list and the **Element type** field value is **IP address**, you can use this list for quick analysis on a reverse proxy. To do so, select the **Use for quick analysis on reverse proxy as** check box.

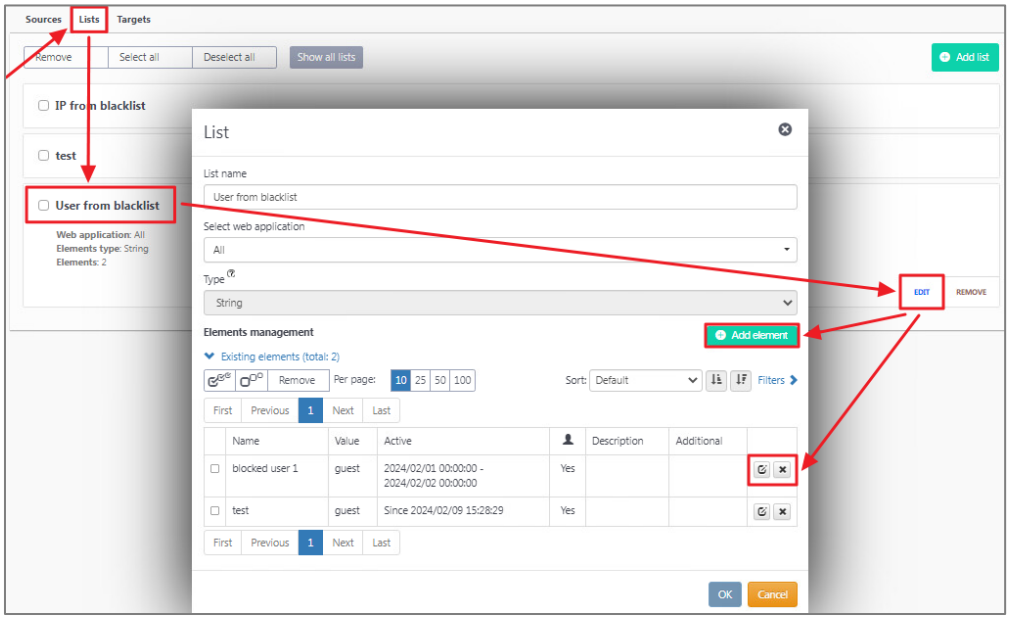


The following two options for using the list for quick analysis on a reverse proxy are available:

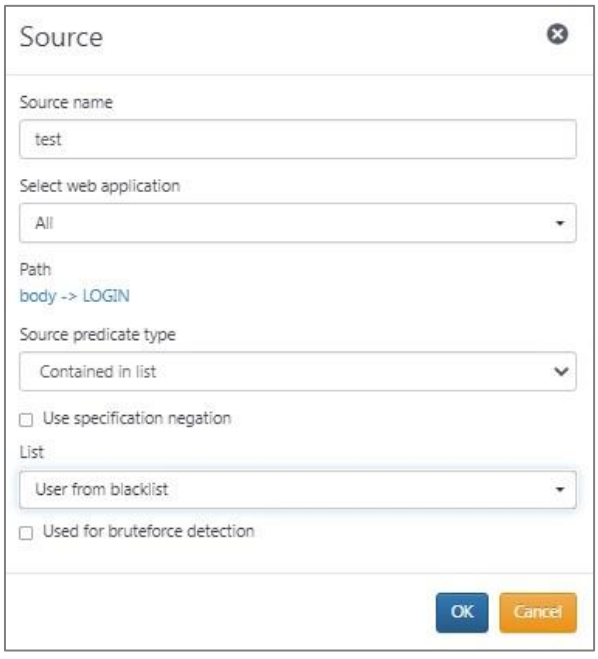
- **list of allowed** — transactions with IP addresses that match the list elements will be passed on the reverse proxy;
- **list of denied** — transactions with IP addresses that match the list elements will be blocked on the reverse proxy.

Edit list elements

To create a list, you need to go to **Sources and lists | Lists**, open the extended description of the required list and click **Edit**.



To use a list, create a new source (see the figure below). Enter the path in the parsing tree. Select **Contained in list** in **Source predicate** type. Then, select the required list from the ones created earlier.



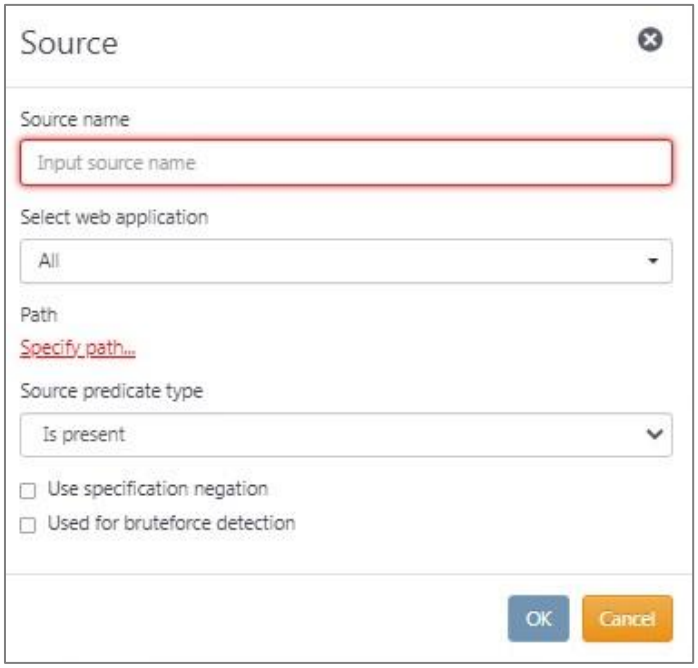
When you view information about such sources, you can quickly check which list membership will be checked for this source in its predicate.

Working with sources

In Continent WAF, you can use any element of the parsing tree as a source. The source can be used in the bruteforce detector as an element from which the number of requests is calculated or in a rule as an element presence (absence) of which is checked in incoming transactions. Lists can be used in sources as an element that performs filtering of source values.

Create a new source

To create a new source, go to **Sources and lists | Sources** and click **Add source**. In the appeared dialog box, specify the name and other parameters of a new source. In particular, you should specify a path in the request parsing tree. The value retrieved using the specified path is considered the value of this source.



To specify a path, click **Specify path**. The **Edit path** dialog box appears. In the **Set path** field, specify the path element. To add path elements in the parsing tree to the required node with the source value, click **+**.

Finally, select one of the predicate types. During a check for a source in a request, the token value from the selected path is retrieved first. Then, the selected predicate is applied to the retrieved value. If the predicate is **True**, then it is considered that the source with the specified name and value is successfully found in the request. Otherwise, it is considered that this source is not in the request.

The following predicates are available:

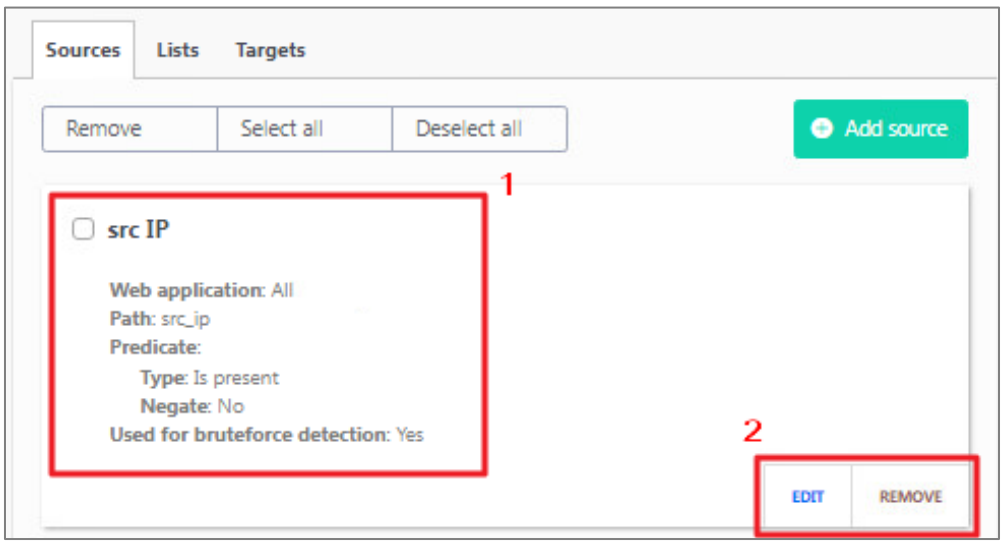
- **Is present** — there is a specified path in the parsing tree (but the value at this top of the parsing tree is not checked in any way).
- **Is absent** — the parsing tree does not contain the specified path.
- **Is empty** — there is no element value at the specified path.
- **Is not empty** — there is an element value at the specified path.

- **Is correct integer** — the element value at the specified path is a correct integer.
- **Is correct IP address** — the element value at the specified path is a correct IP address.
- **Equals** — the element value at the specified path matches the value specified in **Value** and **Value type** (**string**, **integer**, **IP address**).
- **Contains** — the element value at the specified path contains the value specified in **Pattern**.
- **Regular expression** — the element value at the specified path meets the regular expression specified in the additional parameter.
- **Contained in list** — the element value at the specified path matches the value from the list specified in **List**.
- **Contains substring from list** — the element value at the specified path contains a value from the list specified in **List**.
- **Check token expiration** — the element value at the specified path is not expired according to the specified values of additional parameters:
  - **Base path** — the base parsing tree path to the token. If specified, serves as a prefix for all other paths, otherwise the other paths are not suffixes, but absolute paths in the parsing tree;
  - **Expiration path** — the parsing tree path to the timestamp or expiration date of the token. If specified, the token expiration is checked by only this value, otherwise the time or date of token creation and lifetime will be checked;
  - **Creation path** — the parsing tree path to the timestamp or creation date of the token. Used along with the token lifetime if the path to the timestamp or expiration date is not specified;
  - **Lifetime path** — the parsing tree path to the lifetime of the token. Used along with the timestamp or token creation date if the path to the timestamp or expiration date is not specified;
  - **Lifetime in seconds** — a field for specifying a lifetime of the selected element.
- **Check GSSC expiration using specified lifetime** — the element value at the specified path is not expired according to the specified value in **Lifetime in seconds**.

To save a source, click **OK**.

View and edit sources

After you save a new source in the **Sources** tab, you can see the new added source (see the figure below). You can see the main properties of each source (name, parsing tree path, predicate type) by clicking it. To edit or remove existing sources, use the respective buttons in the bottom right corner of each source section. Analysts can edit and remove sources created by analysts. Administrators can edit and remove all created sources.



Working with actions

Continent WAF has such an entity as actions. Actions are comparing client operations on a web application to the application operation logic. When you create an action, you can add not only URL, but also other parameters of a request sent by the client.

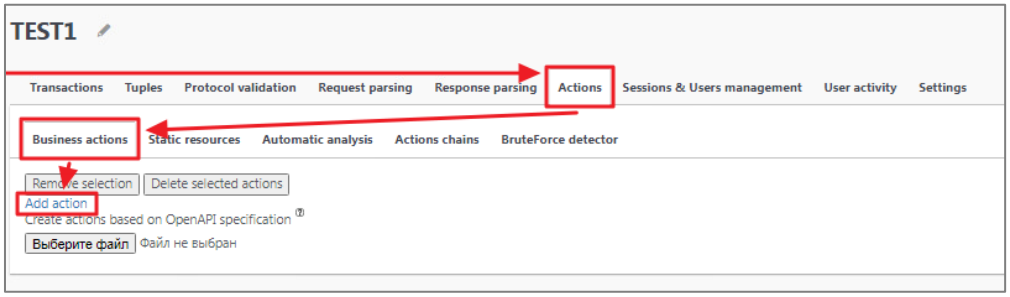
The created actions can be used for selective anomaly suppression, as a parameter in a rule, as a parameter when configuring a bruteforce detector, as parameters when configuring a session model as well as other related modules.

In Continent WAF, you can create actions in the following three ways:

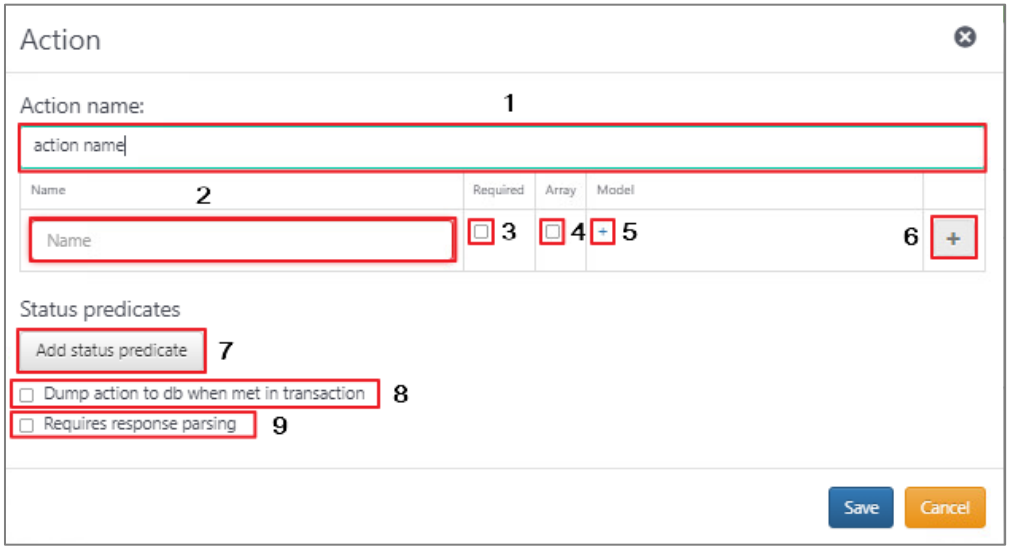
- manual creation of actions;
- semi-automatic creation of actions;
- automatic creation of actions.

Manual creation of actions

To create an action, go to **Applications | Actions | Business actions** and click **Add action**.

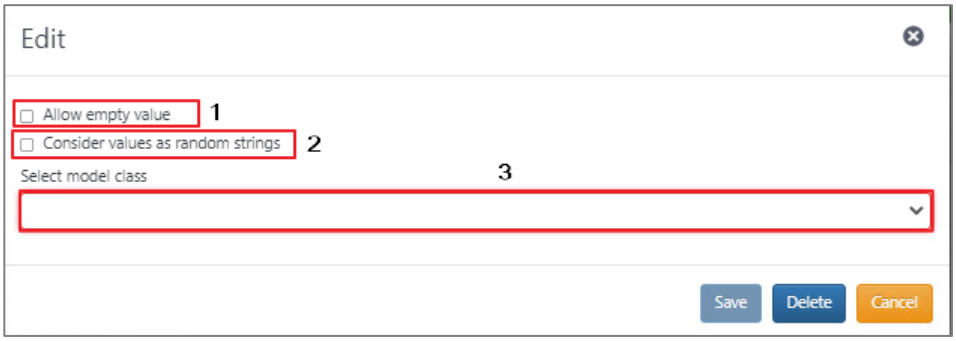


The **Action** dialog box appears.



This dialog box contains the following parameters (see the figure above):

1. **Action name** — a field for entering an action name.
2. **Name** — a field for entering an action parameter. As an action parameter, you can select one or several parsing tree elements that are critical within this action.
3. **Required** — the field enables the mandatory presence of a parameter for the action.
4. **Array** — the field enables characterizing the parameter as a data array.
5. **Model** — when you click **+**, the **Edit** dialog box appears.

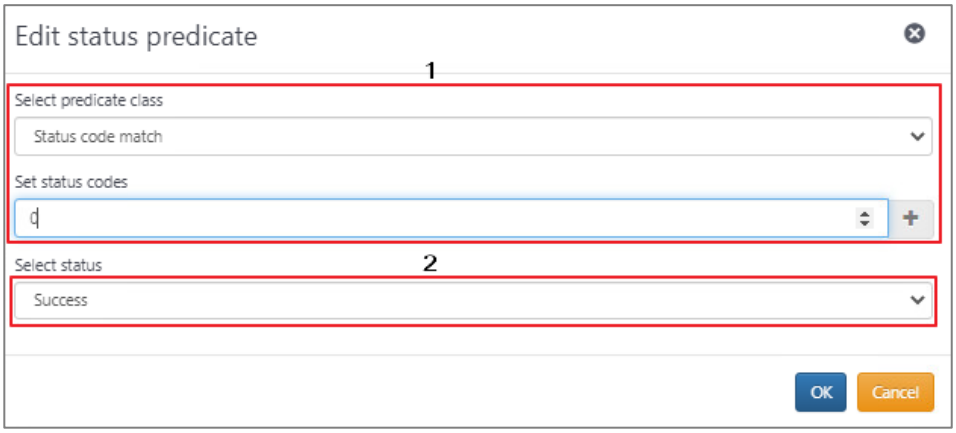


This dialog box contains the following fields:

- **Allow empty value** — a check box that allows or prohibits the empty value of the selected model parameter;
- **Consider values as random strings** — a check box that allows or prohibits the interpretation of the selected action model parameter as a random string;
- **Select model class** — for selecting a class of the action model parameter from a list of possible ones. You can see the list with descriptions in the table below.

Class	Description
Length range	Unchangeable sequence of integers
Pattern	Allows you to specify action model parameter values using a regular expression
Enumeration	Enumeration of allowed options for the value of an action model parameter
E-mail	Check whether the value of an action model parameter matches the standard form of an email address
Date and time	Check whether the value of an action model parameter matches the date and time format
Number	Check whether the value of an action model parameter contains only numbers
URL	Check whether the value of an action model parameter matches the resource ID of the uniform format
String	Check whether the value of an action model parameter matches the string
Filename	Check whether a file is transferred as the value of the action model parameter
Credit card number	Check whether the value of the action model parameter matches the standard form of a credit card number
Hex digits string	Check whether the value of an action model parameter matches the hexadecimal digits string
Structure (JSON, XML, URL query string)	Check whether a structure (JSON, XML, QUERY) is transferred as the value of the action model parameter
Token	Check whether a token with the specified length is transferred as the value of the action model parameter
GUID	Check whether a statistically unique 128-bit ID is transferred as the value of the action model parameter
Universal text model	Check whether a universal text model is transferred as the value of the action model parameter. A universal model cannot be created manually (only automatically)
Length expected value	Check whether the average value (weighted using the probabilities of possible values) of a random variable is transferred as the value of the action model parameter
Univariate gaussian feature	Check whether a value that matches the specified model is transferred as the value of the action model parameter
Multivariate gaussian feature	Check whether a value that matches the specified model is transferred as the value of the action model parameter

6. In the highlighted area, there is a button that adds another line for entering a model parameter.
7. **Add status predicate** — when you click this button, the **Edit status predicate** dialog box appears.



This dialog box contains the following fields:

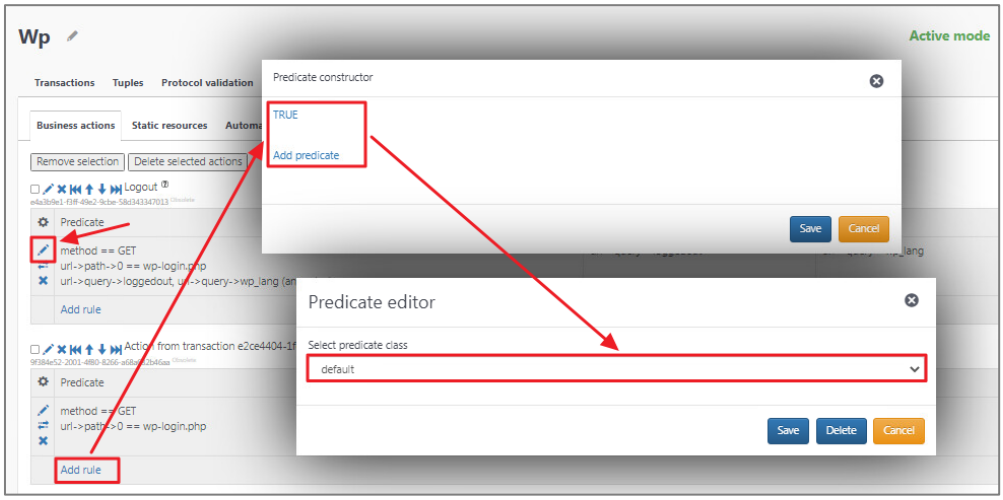
- **Select predicate class** — for selecting a predicate class from the specified list. Depending on the selected predicate class, the parameter using which this class is specified changes;

Class	Description	Parameters
Status code match	Check whether the value of the response code matches the one specified in the parameter. In case of a match, the transaction is assigned the success status selected in the **Select status** field	**Set status codes** — a field for entering the status code value. You can add additional input fields by clicking **+**
Body regexp match	Check whether the value of the response body matches the regular expression specified in the parameter. In case of a match, the transaction is assigned the success status selected in the **Select status** field	**Set regexp** — a field for entering a regular expression
Path response match	Check whether the values of the specified paths of the response parsing tree match the specified parameters. In case of a match, the transaction is assigned the success status selected in the **Select status** field	You can add parameters with the following predicate classes: • **regexp** — a check whether a value in the specified branch of the response parsing tree matches a regular expression; • **value** — a check whether a value in the specified branch of the response parsing tree matches the specified value; • **paths present** — a check whether the specified branch of the response parsing tree is present

- **Select status** — a field for selecting from the following three options: **Success**, **Validation error**, **Logic error**.

8. **Dump action to db when met in transaction** — this check box enables saving the action to the database if it is found in the transaction.
9. **Requires response parsing** — select this check box if the action uses a status predicate.

After you save the changes on the **Business actions** subtab, you can edit the action predicate by clicking **Add rule** (if the action already has predicates, click  in the **Predicate** section). The **Predicate constructor** dialog box appears. In this dialog box, you can add predicates by clicking the respective button.

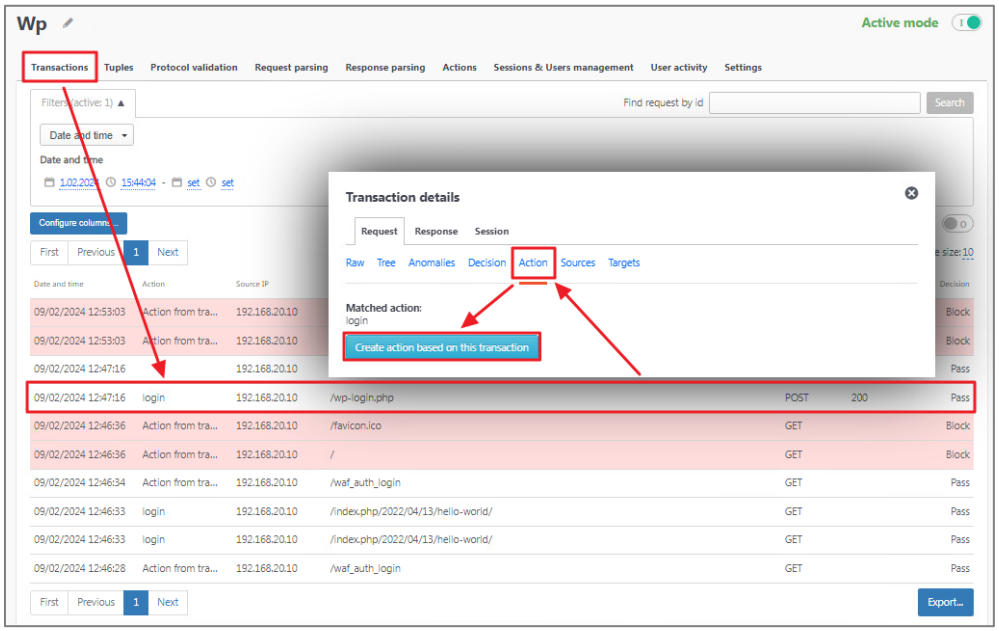


The following three values of the **Select predicate class field** are available:

- **regexp** — a check whether a value in the specified parsing tree branch matches a regular expression;
- **value** — a check whether a value in the specified parsing tree branch matches the specified value;
- **paths** present — a check whether the specified branch of the response parsing tree is present.

Semi-automatic creation of actions

In Continent WAF, you can create actions based on transactions. To create action this way, go to **Applications | Transactions** and open the **Transaction details** dialog box for the transaction for which you need to create an action. Then, go to the **Action** subtab and click **Create action based on this transaction**.

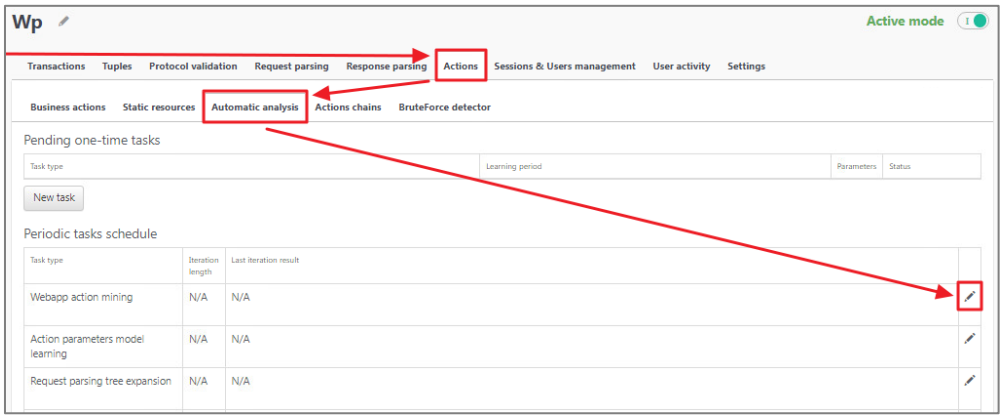


The information window appears. It contains the created action ID that is used to denote this action in databases.

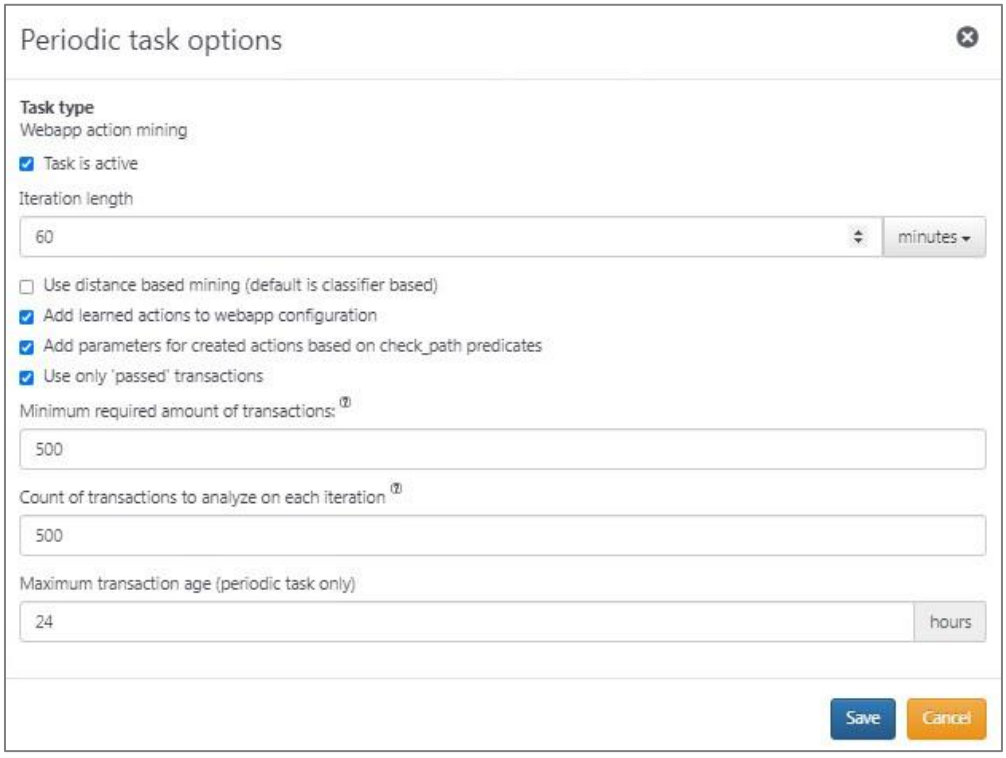
You can find the created action in **Applications | Actions | Business actions** at the bottom of the list.

Automatic creation of actions

Continent WAF can automatically add requests to static resources using machine learning. This module is enabled by default and can be additionally configured via the Continent WAF web interface. To do so, go to **Applications | Actions | Automatic analysis** and edit the **Webapp action mining** task.



The **Periodic task options** dialog box appears.



You can see the description of this dialog box fields in the table below.

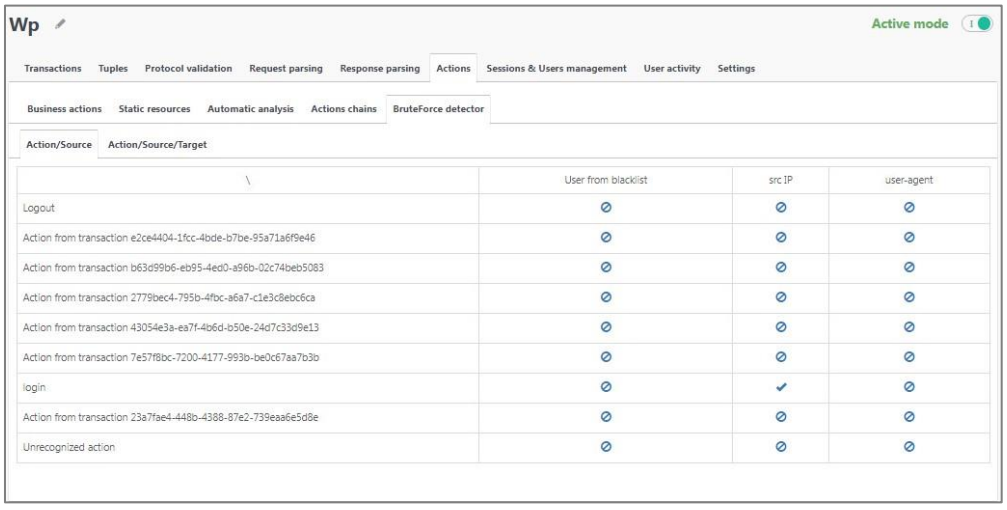
Field	Description
Iteration length	Task start interval
Use distance based mining (default is classifier based)	The default value is classifier based mining
Add learned actions to webapp configuration	Adds an action to the database to avoid duplicate actions
Add parameters for created actions based on check_path predicates	Creates an action model
Use only 'passed' transactions	Ignores all transactions that were not passed
Minimum required amount of transactions	Specifies a lower threshold of the number of transactions from different sources with the same structure. When this threshold is reached, an action is created. No more than 10000
Count of transactions to analyze on each iteration	Specifies a threshold of analyzed transactions at each analysis iteration. No more than 10000
Maximum transaction age (periodic task only)	Specifies the depth of transaction analysis. Only for a periodic task

Bruteforce detector

Bruteforce detector can detect bruteforce attacks — too frequent requests to separate actions of a web application. To do so, the bruteforce detector performs the leaky bucket algorithm and provides rate limiting. The subjects for which rate limiting restrictions are created are sources.

Action/Source bruteforce detector

This bruteforce detector works the following way: the bruteforce detector processes all requests for different source — web application action pairs separately from each other. In other words, for each individual request source and each individual web application action there is its own bucket with tokens the state of which monitors the frequency of requests from a specific source to a specific web application action. Therefore, you can configure, for example, the maximum number of requests per time unit from each individual IP address or create restrictions on the number of attempts to log on to a web application for each individual user. The **BruteForce detector** subtab contains a table. In this table, rows are configured sources and columns are configured actions in **Actions | Business actions**.

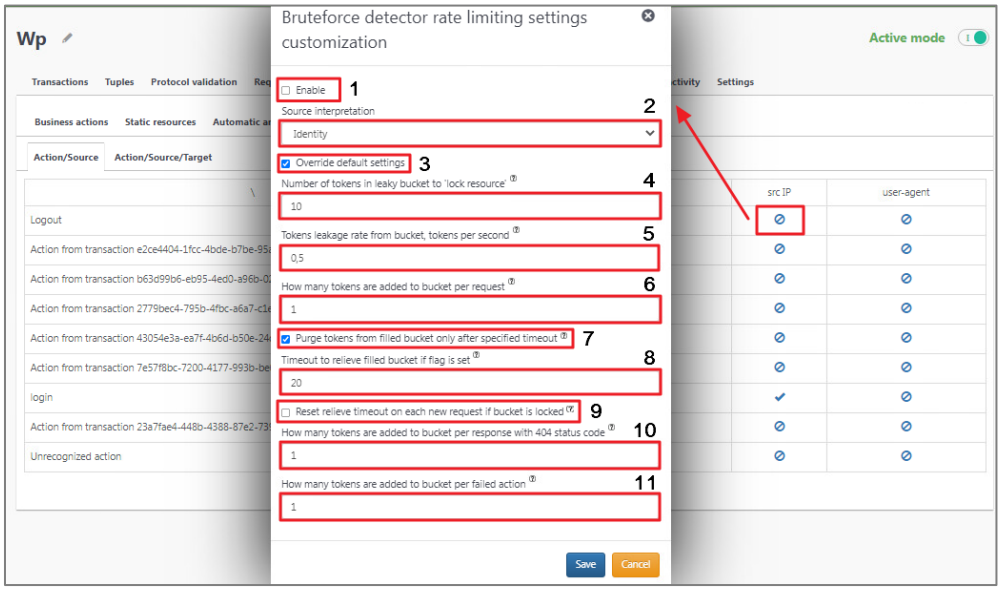


Each cell can take one of the following three values:

- ✔ — the bruteforce detector for the action — source pair is enabled and, by default, the detector parameter settings are set for the module and are available for viewing in the **BruteforceDetector** section of the analyzer module configuration;
- ✔✎ — the bruteforce detector for the action — source pair is enabled and configured, an individual token bucket with set parameters is created;

- — the bruteforce detector is disabled.

To change the detector state, click the required cell. The **Bruteforce detector rate limiting settings customization** dialog box appears.



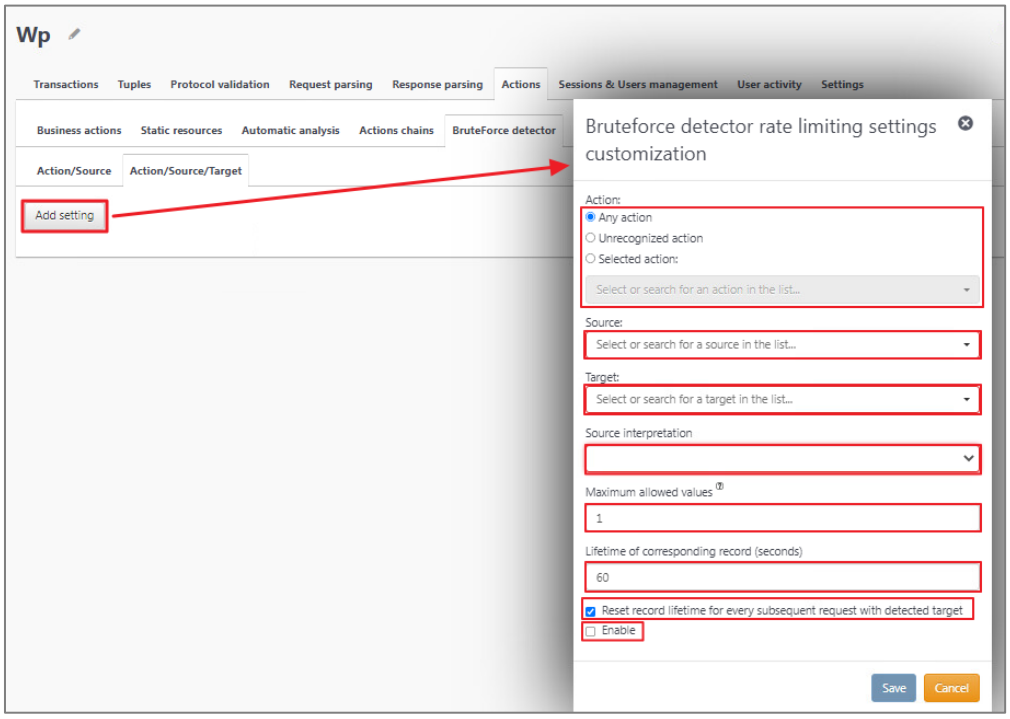
This dialog box contains the following parameters (see the figure above):

1. **Enable** — a check box for enabling the bruteforce detector.
2. **Source interpretation** — can take one of the following two values:
   - **Identity** — for each new source value, a new bucket is created;
   - **Value set** — creates one bucket for all source values.
3. **Override default settings** — a check box for enabling individual settings.
4. **Number of tokens in leaky bucket to 'lock resource'** — a number of tokens that the bucket can contain. If the bucket contains more tokens than the specified number, an anomaly is generated on next request.
5. **Tokens leakage rate from bucket, tokens per second** — a number of tokens that flow out of the bucket every second. This value determines how much time it takes for a full bucket to become empty.
6. **How many tokens are added to bucket per request** — each request adds a specified number of tokens to the bucket. After this, the check whether the bucket is overflowed is performed. Depending on the result, either an anomaly is generated or the request passes successfully.
7. **Purge tokens from filled bucket if flag is set** — if you select this check box, then when the bucket overflows, all subsequent requests from this source will generate anomalies for the specified time, regardless of the bucket state. After this timeout expires, the bucket is forcibly purged.
8. **Timeout to relieve filled bucket if flag is set** — sets the timeout for the previous option.
9. **Reset relieve timeout on each new request if bucket is locked** — if you select this check box, the subsequent requests during the timeout reset the timeout. Otherwise, the timeout does not reset.
10. **How many tokens are added to bucket per response with 404 status code** — the request is processed based on the current bucket state. When a web application response is received, the status code is analyzed. If it equals 404, the specified number of tokens will be added to the bucket. This option is useful if you want to detect attempts of forceful web app browsing — guessing direct URLs instead of following links.
11. **How many tokens are added to bucket per failed action** — the request is processed based on the current bucket state. When a web application response is received, the success of the action is analyzed. If the action is not successful, the specified number of tokens will be added to the bucket. You can see the use case of this option further.

Action/Source/Target bruteforce detector

The previous bruteforce detector (Action/Source) counts the number of requests from source to action. The Action/Source/Target bruteforce detector counts the number of requests from source to target within a specific action. This type of bruteforce detector fits well for searching through the values of individual fields, whether it is a login, password, promo codes or something else.

To add a configuration of Action/Source/Target bruteforce attacks, go to **Applications | Actions | BruteForce detector | Action/Source/Target** and click **Add setting**. The **Bruteforce detector rate limiting settings customization** dialog box appears.



This dialog box contains the following parameters (see the figure above):

1. **Action** — for selecting an action for which the bruteforce detector is configured.

   The following three options are available:

   - **Any action** — all actions on this application including unrecognized ones;
   - **Unrecognized action** — this action includes the transactions that do not match any of the configured actions;
   - **Selected action** — for selecting an action from the list of actions configured in this app.
2. **Source** — for selecting a source from the ones configured in the app.
3. **Target** — for selecting a target from the ones configured in the app.
4. **Source interpretation** — can take one of the following two values:
   - **Identity** — for each new source value, a new bucket is created;
   - **Value set** — creates one bucket for all source values.
5. **Maximum allowed values** — specifies the maximum number of different values that one web application object can take before generating anomalies.
6. **Lifetime of corresponding record (seconds)** — the lifetime of a record generated by the bruteforce detector.
7. **Reset record lifetime for every subsequent request with detected target** — resets the lifetime specified in field 6 when receiving a record identical to the already created one.
8. **Enable** — a check box for enabling the bruteforce detector.

Configure session model

A session is a mechanism for distinguishing between the requests of one user and the requests of another user. The HTTP protocol does not have session tracking mechanism. However, a solution was found — using cookies, request body, headers to transfer session information. The classic case is when the client goes to the application, he is assigned a cookie that uniquely identifies the client. Then, this cookie is sent with every request from this client. This type of cookie may have a set lifetime.

Session model is a mechanism that allows you to bind Continent WAF to these sessions.

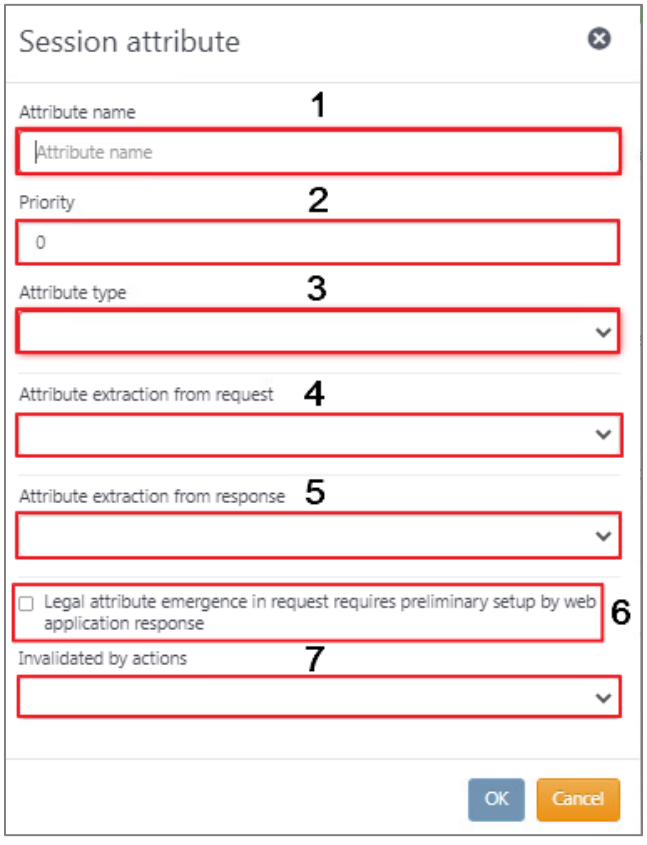
Create session attributes

Continent WAF generates session ID based on the session attributes. The session ID format is not specified, it is implemented as a hash function from a set of session attribute values and differs for different sessions. Used only within Continent WAF.

You can create session attributes either in **Applications | Sessions & Users management**, then the session attribute will be used for the selected application, or in **Settings | Session tracking**, then you can select whether this session attribute will be used for all installation applications or only for a specific application.

For the session model to work, you need to make sure that the lightweight session tracker (LWSessionTracker) module is enabled in the analyzer settings. If you need to track the preliminary session cookie assignment, you must enable the request parsing (DecisionTreeResponseParser) module. For details about the analyzer module settings, see [**1**].

You can see the dialog box for creating a session attribute in the figure below.



This dialog box contains the following parameters (see the figure above):

1. **Attribute name** — a field for entering a session attribute name.
2. **Priority** — a field for entering a session attribute priority. The lower the number, the more important this attribute is.
3. **Attribute type** — a field for selecting one of the following three attribute types:
   - **Extract session id** — a session attribute that uniquely identifies a session;

- **Extract sequence id** — required to make the transition from an unauthorized session to an authorized one in cases when the session attribute changes during authorization (for example, PHPSESSIONID);
- **Extract user name** — extracting the user name allows you to track user activity within a session.

4. **Attribute extraction from request** — a field for selecting the location of the request from which the session attribute will be extracted. In some cases, you need to specify the parsing tree path to the element that will be used as a session attribute.
5. **Attribute extraction from response** — a field for selecting the location of the response from which the session attribute will be extracted. In some cases, you need to specify the parsing tree path to the element that will be used as a session attribute.
6. **Legal attribute emergence in request requires preliminary setup by web application response** — if you select this check box, when a transaction uses a session attribute that was not previously set in the response, this transaction will be blocked.
7. **Invalidated by actions** — a field for selecting an action from the ones installed on the application as the one invalidating the session attribute.

Anomalies generated by session model

The LWSessionTracker module generates RequestAnomaly anomalies of the following types:

- for request attributes that needed to be set in the response but were not set:
  - **reason** — Session with these attributes is unknown;
  - **topics** — MODEL, DEFAULT_SESSION, CORRELATION;
  - **additionally** — a list with uuid, types, names and values of violating attributes and a hint that the situation may occur due to the fact that response parsing is disabled;
- when the limit on the number of users per sequence_id is exceeded:
  - **reason** — Multiple user ids in the same action sequence;
  - **topics** — AUTOMATION, SESSION, CORRELATION;
  - **additionally** — the current number of different users observed for the sequence;
- for custom attribute validation errors:
  - **reason** — Custom attribute validation error;
  - **topics** — MODEL, CORRELATION, INVALID_CUSTOM_ATTRIBUTE;
  - **additionally** — a list of errors from each of the custom validators (their specific format and contents depend on the fired validators).

Create action chains

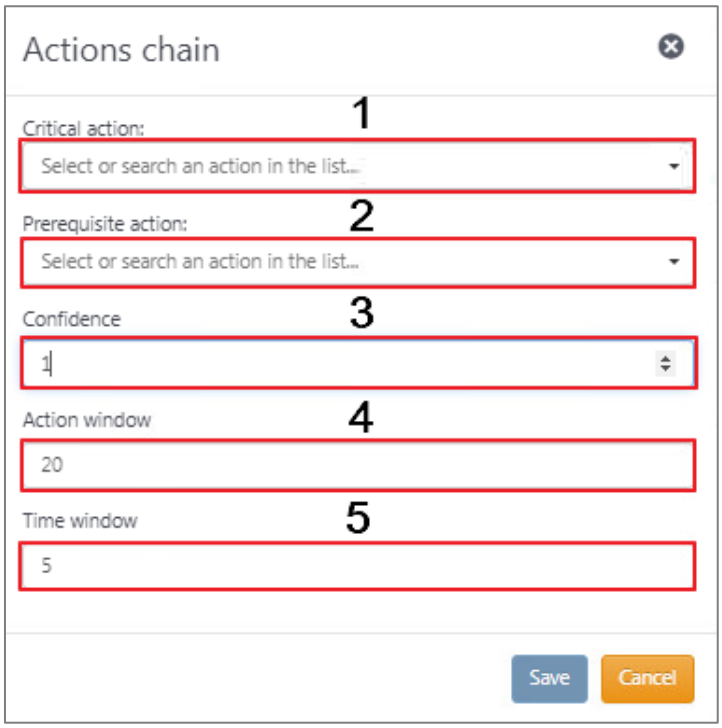
When working with a web application, it is assumed that users have stable action patterns primarily determined by the application web interface.

A classic example: a legitimate user cannot submit a form without downloading it first. Therefore, in Continent WAF, you can first see a GET request to download the authorization form, then there should be a POST request to submit the data. Bot systems very often ignore the action of downloading the form and immediately send POST requests with authorization data and disrupt the normal sequence of work with the web application.

The action chain operation is based on these differences. It is important to understand that action chains work on top of the session model.

To configure an action chain, go to **Applications | Actions | Actions chains** and click **Add new actions chain**.

The **Actions chain** dialog box appears. In this dialog box, you can create a new action chain.



This dialog box contains the following parameters (see the figure above):

1. **Critical action** — for selecting an action from the ones installed on the application as the action that should end the action chain.
2. **Prerequisite action** — for selecting an action from the ones installed on the application as the action that should start the action chain. You can select several prerequisite actions.
3. **Confidence** — fractional value from 0 to 1. Chain confidence level. The more often the actions in the chain occur simultaneously and in the same order compared to each other, the higher the confidence value for the chain is when it is generated by machine learning tools. The session anomaly detection module uses this parameter when chains are broken. As you know for certain what predicate action should occur before the critical action when creating chains manually, this value is set in field 1.
4. **Action window** — an integer greater than 0. When a transaction with a critical action appears, a trace that consists of the last **n** transactions is created, where **n** is the action window for the chain with this critical action. That means the action window is **n** transactions before the critical action and a predicate action must be present among these transactions.
5. **Time window** — an integer greater than 0. Value in minutes. Within the time window, the action window is considered, so the trace from **n** transactions (where **n** is the value of the action window) that were made in the last **m** minutes (where **m** is the value of the time window) before the critical action is created. Transactions outside the time window are not considered.

For correct module operation, you need to configure it in the analyzer. For details about the analyzer module settings, see [**1**].

Chapter 4
Main application scenarios

Operator work scenarios

The operator monitors event graphs and if emergencies occur, decides whether it is necessary to involve the analyst or the administrator of Continent WAF. Possible emergency indicators are as follows:

1. Transactions with 5xx response codes.
2. Changes in event graphs.
3. Increased number of events.
4. Application user tickets.

Transactions with 5xx response codes

The operator must monitor event graphs in the **Overview** section and if events marked red (transactions with 5xx response codes) appear in the **Status** event graph, the administrator must be notified immediately.

Changes in event graphs

Changes in event graphs:

- **Status**;
- **Decision**;
- **Delay**;
- **Session**;
- **Hostility** —

must be examined by time and compared to the events which happened at that time. The analyst and the administrator must be informed about the situation. For example, a steep drop to zero in all graphs indicates a malfunction of Continent WAF.

Increased number of events

If the number of events suddenly increases, you need to go to the **Events** section. For each event with increased number of firings, you need to open general information about the event and click **Details**. In the dialog box that appears, you can see general information for transactions assigned to the event and open transaction details. The **Decision** tab of the **Transaction details** dialog box allows you to see which rules affect the transaction and look through the rules themselves. For example, the increased number of events by marking rules may indicate an increase in legitimate number of requests to the application related to any events. The operator must analyze these facts and decide whether it is necessary to inform the analyst or the administrator.

Application user tickets

Application user can submit a ticket to the technical support. When submitting a ticket, a user needs to tell the technical support the Support id that displays when the user access to the application is blocked. For example:

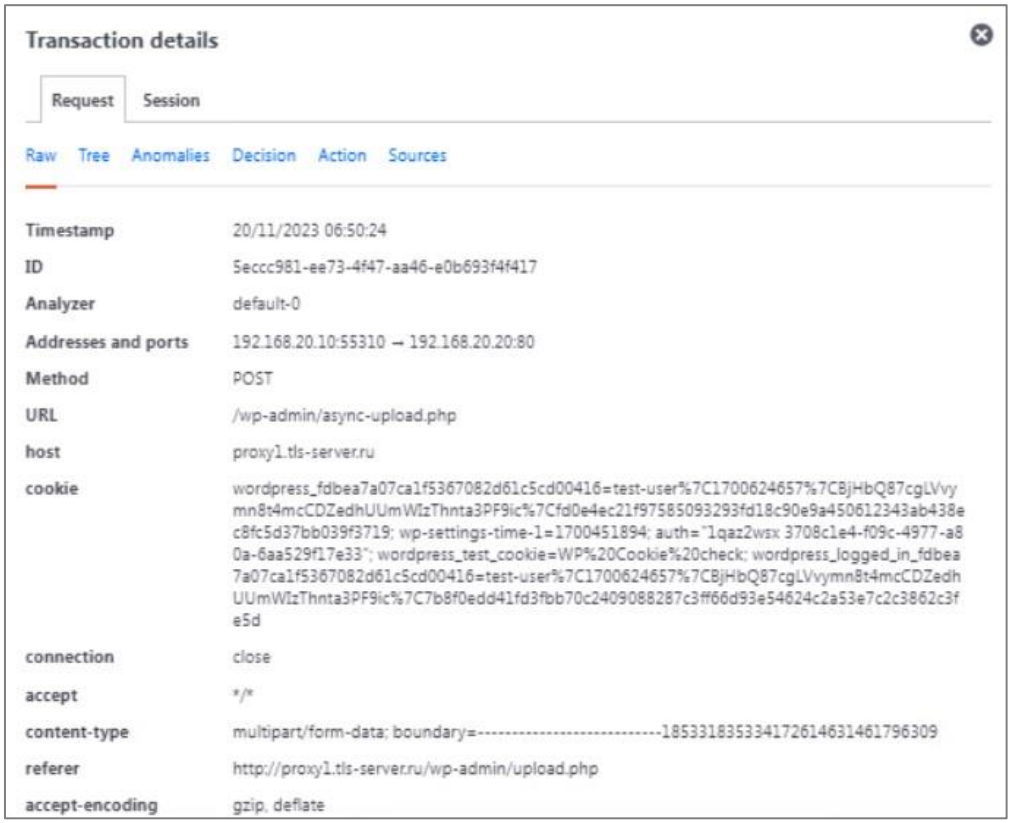
```
Access to resource was blocked.

Support id: c34090f1-2477-4022-b8f7-fc1b3d283136
```

In **Applications | Transactions**, the Continent WAF operator searches for this Support id by entering it in the **Find request by id** search bar in the top right corner.

The **Transaction details** dialog box appears as a search result.

The operator must determine which transaction part was blocked — the request or the response. If the request was blocked — the **Response** tab is missing from the **Transaction details** dialog box.



Continent WAF blocks server responses with the 5xx status or responses containing sensitive data leak (for example, SQL or JAVA error data) by default to prevent disclosing data about the application device, web component versions and diagnostic information about errors to the attacker. You can view the body of the blocked server response in **Transaction details | Response | Raw**.

If the response was blocked correctly — you should recommend the application user to repeat the request if the error or failure were temporary or inform the Continent WAF administrator about the occurred error or failure.

If the response was blocked incorrectly and this server response must be shown to users — you need to contact the analyst or the administrator of Continent WAF.

Analyst work scenarios

The analyst monitors events, adjusts rules, creates actions and suppresses anomalies.

Blocking analysis and false positive response suppression are performed in the following order:

1. Blocked transaction search.
2. Analysis of transaction anomalies that caused the blocking.
3. False positive response suppression.

Blocked transaction search

You can see an example of the message on the blocking page below:

```
Access to resource was blocked.

Support id: c34090f1-2477-4022-b8f7-fc1b3d283136
```

To search a blocked transaction in the Continent WAF web interface:

- Go to **Applications | Transactions**, enter the Support id value from the message in the **Find request by id** search bar and click **Search**.

Transaction details dialog box appears.

Analysis of transaction anomalies which caused the blocking

To analyze transaction anomalies which caused the blocking:

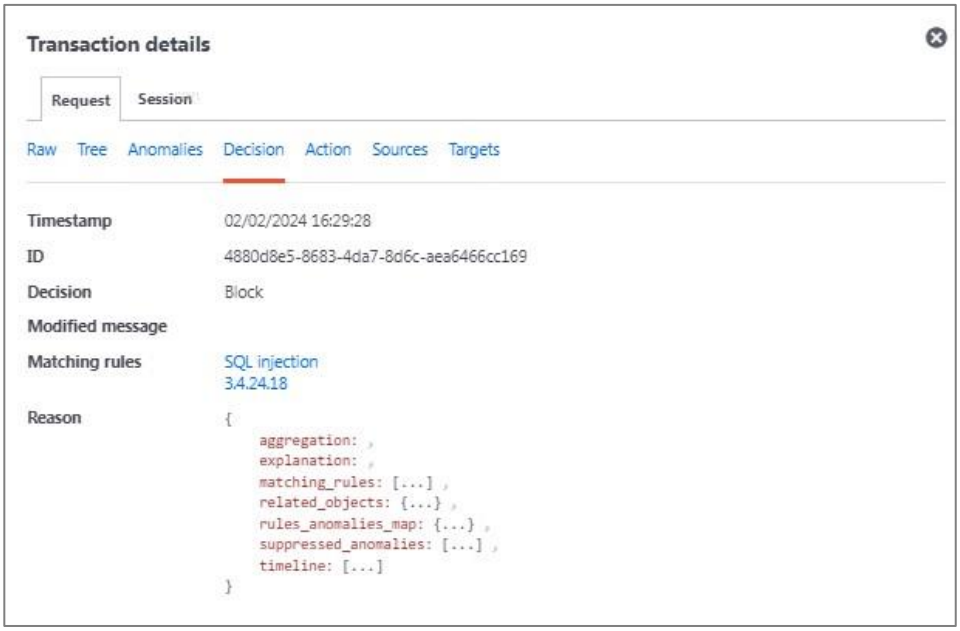
1. Search for the blocked transaction (see above).
2. Determine which transaction part was blocked — the request or the response. If the **Response** tab is missing from the **Transaction details** dialog box, the request was blocked.

   Continent WAF blocks server responses with the 500 status or responses containing sensitive data leak (for example, SQL or JAVA error data) by default to prevent disclosing data about the application device, web component versions and diagnostic information about errors to the attacker.

   You can view the body of the blocked server response in **Transaction details | Response | Raw**.

   Note. If the server response was blocked correctly, you need to repeat the actions. If the error occurs constantly, you need to consult with the subject matter experts to analyze the web server or web application error.

3. Go to the **Decision** tab of the blocked request where fired rules are displayed.

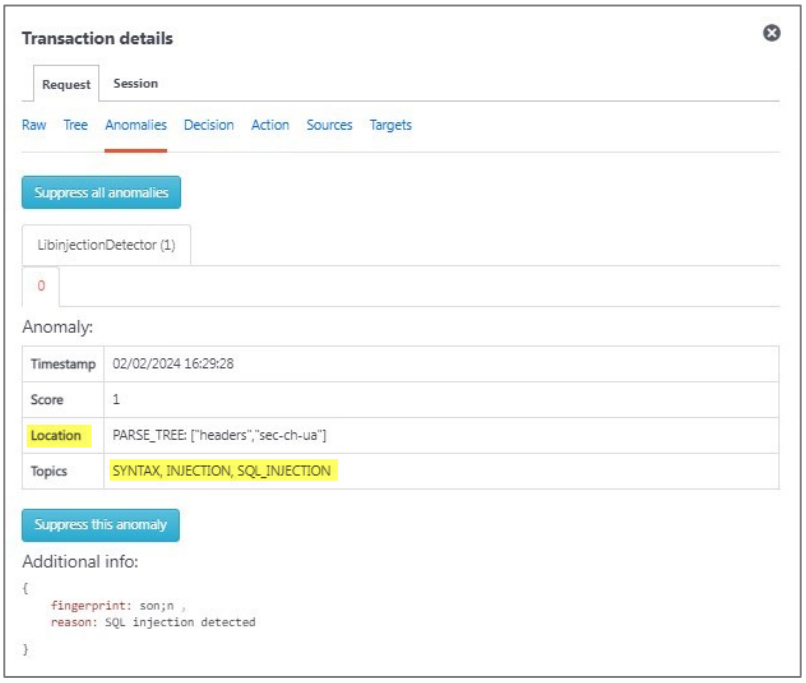


4. Click the rule name.

   The **Rule** dialog box appears. In this dialog box, you can analyze whether the rule is blocking or marking and the anomaly type that fires this rule.

   Click **Cancel**.

5. Go to the **Anomalies** tab of the **Transaction details** dialog box and analyze which anomaly types are related to blocking rules and are not related to suppressed anomalies.

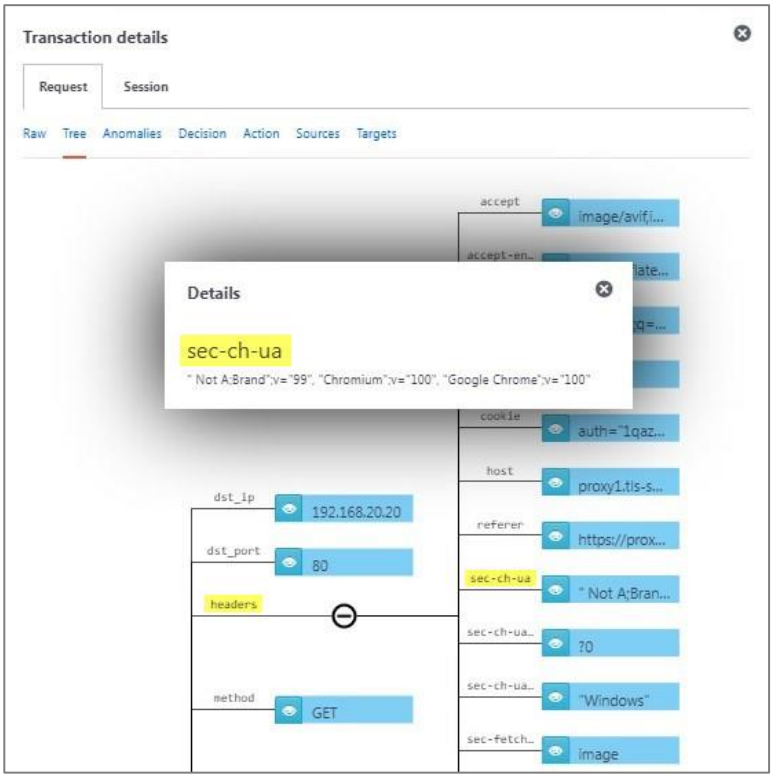


6. For the LibinjectionDetector analyzer, analyze the request element which caused the anomaly.

   In this example, you need to analyze 1 anomaly numbered **0** with the **SQL_INJECTION** type of the LibinjectionDetector analyzer.

   For each anomaly, the request element is determined (**Location**). In the example in the figure above, the anomaly #0 was detected in the **headers**, **sec-ch-ua** elements in the **Parse Tree** request part.

   On the **Anomalies**, **Tree** and **Raw** tabs, analyze the request element that caused the anomaly and decide whether this anomaly is a false positive response.



String values generated in the app randomly, for example, session IDs and tokens, encoded and compressed elements, contents of files uploaded to the app and text fields, user passwords and other can be a sign of a false positive response.

Signs of real attacks on the web application, for example, SQL (as in the figure above), JavaScript, PHP language constructions, shell commands, etc., can be a reason for a legitimate blocking. In these cases, you do not need to suppress the anomaly except the following scenarios:

- SQL requests can be a legitimate format of transferring data or control commands in HTTP requests, for example, for the Grafana application or application using GraphQL language. If SQL language constructions are found in requests of the majority of application clients, it may be a sign of such situation.
- Users can publish code examples on a protected resource. To decide whether to suppress such anomalies, you need to analyze the context of such publications.

During a false positive response, an anomaly suppression is performed as described in the **False positive response suppression** section for the LibinjectionDetector analyzer.

False positive response suppression

If you determine that an anomaly is a false positive response, you need to suppress the false positive response or adjust the model depending on the model type of the analyzer that detected this anomaly.

Suppress anomalies in server responses

To suppress anomalies in server responses:

1. In **Transaction details**, go to **Response | Anomalies** and click **Suppress all anomalies**.

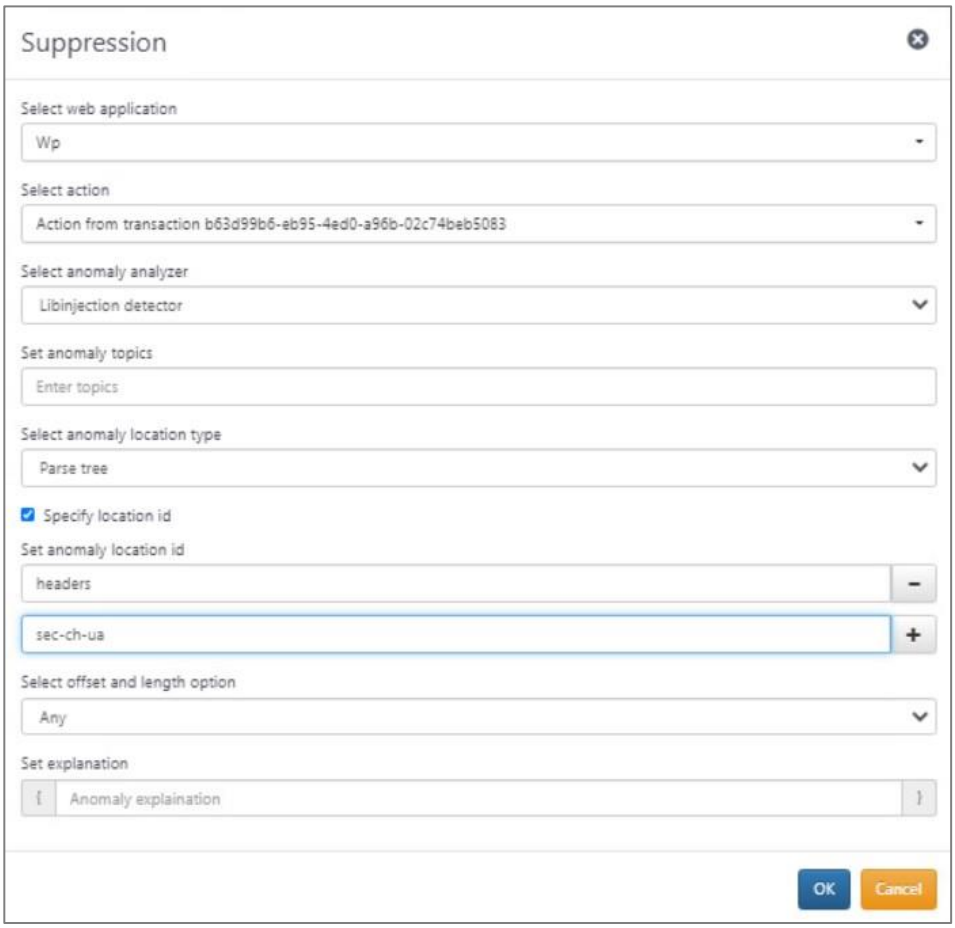
   A dialog box for suppressing all anomalies detected in the response appears.
2. Click **Suppress**.

Suppress anomalies in requests

To suppress anomalies in requests for the LibinjectionDetector analyzer:

1. In **Transaction details**, go to **Response | Anomalies** and click **Suppress all anomalies**.

   A dialog box with parameters of anomaly suppression appears.



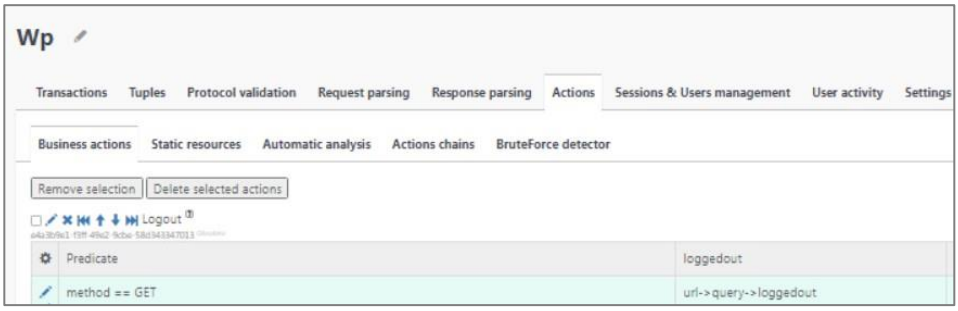
2. Perform the anomaly suppression.

   For anomalies in request elements, which can be common for all requests, for example, **Cookie** and **user-agent** headers, select **All actions** in the **Select action** drop-down list.

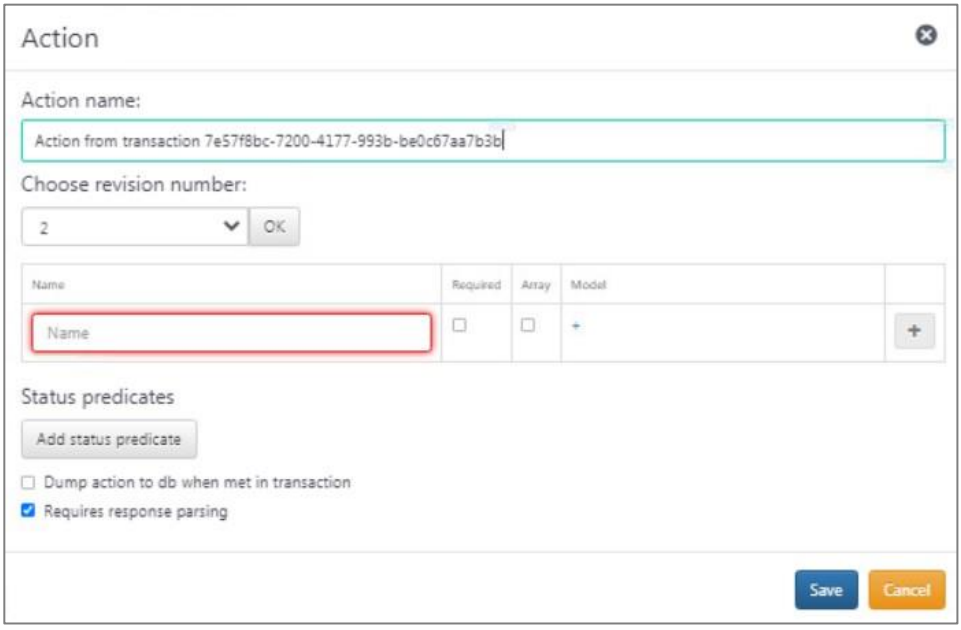
To suppress anomalies in requests for the ActionParamValidator analyzer:

1. To adjust the parameter model, on the **Action** tab, determine which action this request is related to.

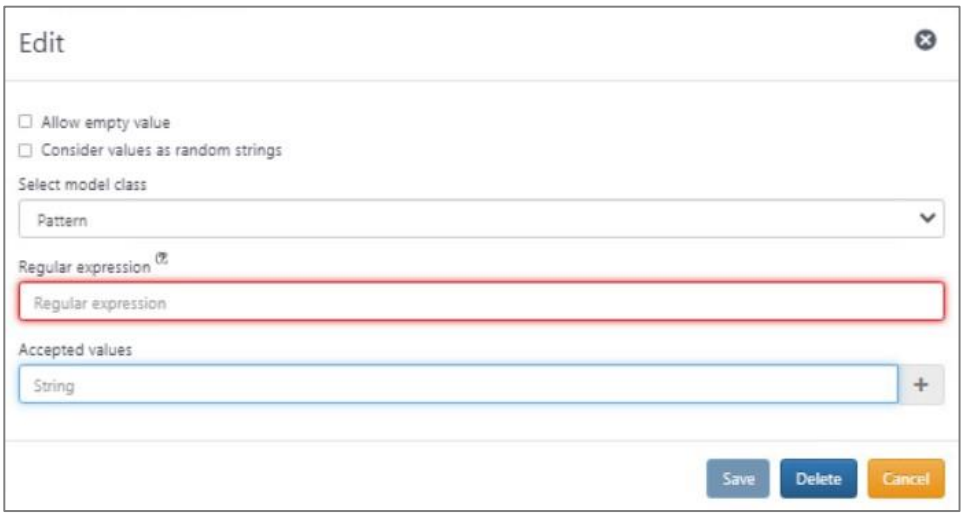
2. Go to **Applications | Actions**.



3. Find the action which parameter caused the anomaly.
4. Click the  icon located near the name of the required action.
   A dialog box appears like in the figure below.



5. Find the required parameter and edit its syntax model.



Examples of configuring nontrivial bruteforce detector settings

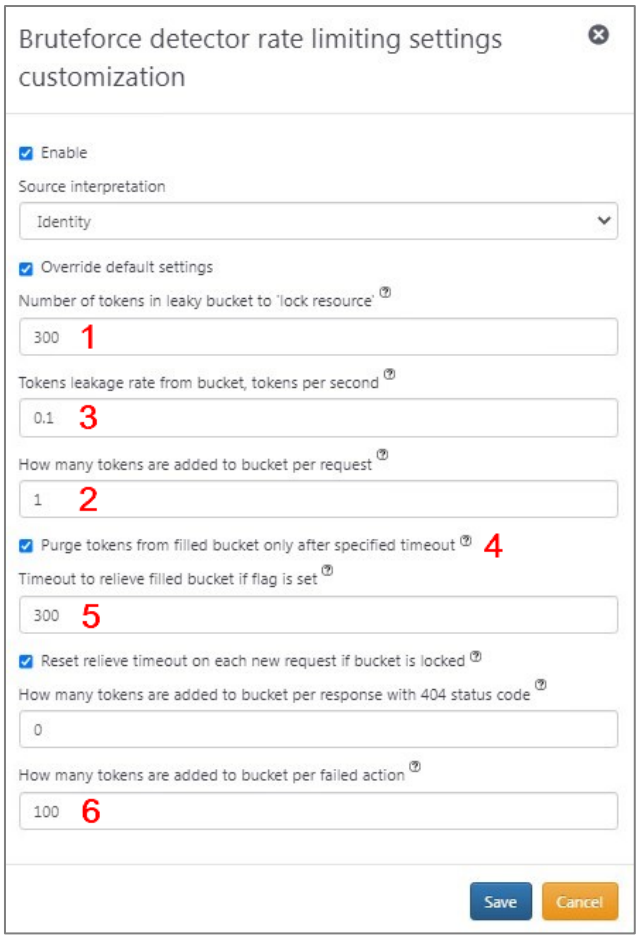
The rate limiting setting protects logon from bruteforce attacks via limiting the frequency of requests to log on to the web application from a single IP address to 4 requests per minute (average). However, this setting can be inapplicable in some situations; for example, if a large number of users log on via NAT from a single IP address or log on in sync at the same time.

You can see an alternative option for protecting the login action from bruteforce attacks below.

Configure the following settings for the **login** — **useragent** pair:

1. A bucket with relatively large size (see **1** in the figure below).
2. A regular request to log on to the web app adds an insignificant amount of tokens (see **2** in the figure below).
3. Tokens leakage rate is low, but the bucket is guaranteed to clear in an hour (see **3** in the figure below).
4. If the bucket is overflowed (see **4** in the figure below), purge tokens from it (see **5** in the figure below) after 5 minute timeout.
5. Add a significant amount of tokens (see **6** in the figure below) if the action failed.

These settings limit the number of failed web app logon attempts from a single user. In case of bruteforce attack, the fourth request is likely to be blocked because of the bucket overflow. Within the next 5 minutes, all further logon attempts from this user will be blocked because of the timeout. After the timeout, the bucket is relieved which grants a regular user who forgot the password three more logon attempts.



The second possible scenario is using bruteforce detector to block users from the blacklist.

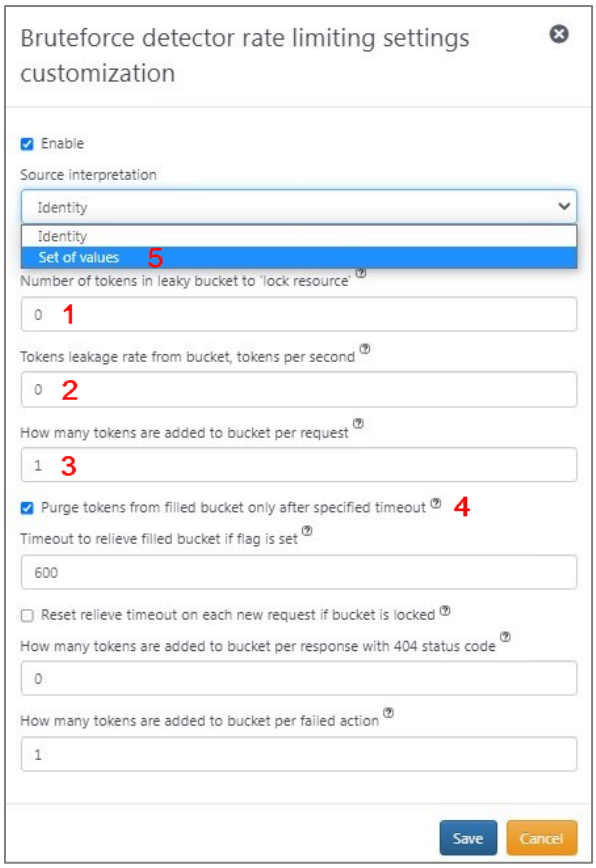
Configure the following settings for the **login** — **User from blacklist** pair:

1. A bucket with zero size (see **1** in the figure below).
2. Tokens do not leak from bucket over time (see **2** in the figure below).
3. Each request adds a token (see **3** in the figure below).
4. The first request with this source overflows the bucket and tokens are purged after timeout (see **4** in the figure below).

The first request will overflow the bucket and will be blocked.

In this case, there is no use in distinguishing different blacklisted users — only the fact of belonging to the blacklist matters. Therefore, you do not have to create, store and monitor separate leaky buckets for each such user. In the **Source**

interpretation drop-down list, select **Set of values** instead of **Identity** (see **5** in the figure below). Now one common bucket is used for the whole group of any values from the **Users from blacklist** source.



Similarly, you can create more complex rate limiting restrictions for groups of different subjects belonging to one of the sources.

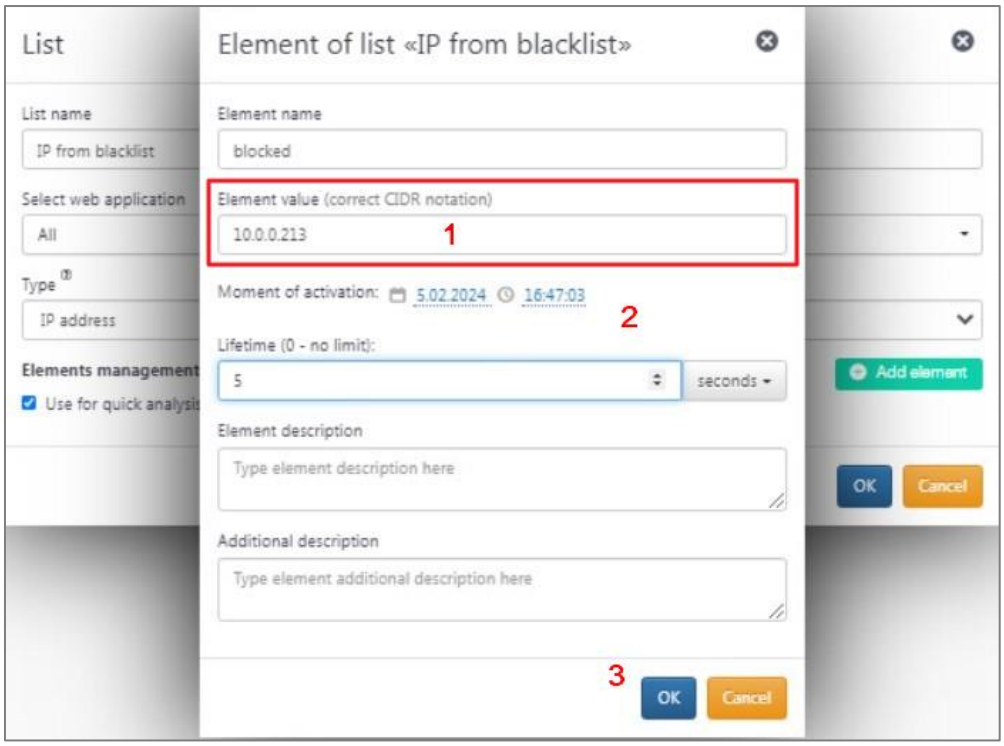
Using IP address lists on reverse proxy server

Lists elements of which are records in a CIDR notation can be uploaded directly to the scwaf-nginx reverse proxy server and used on it as black- or whitelists of IP addresses.

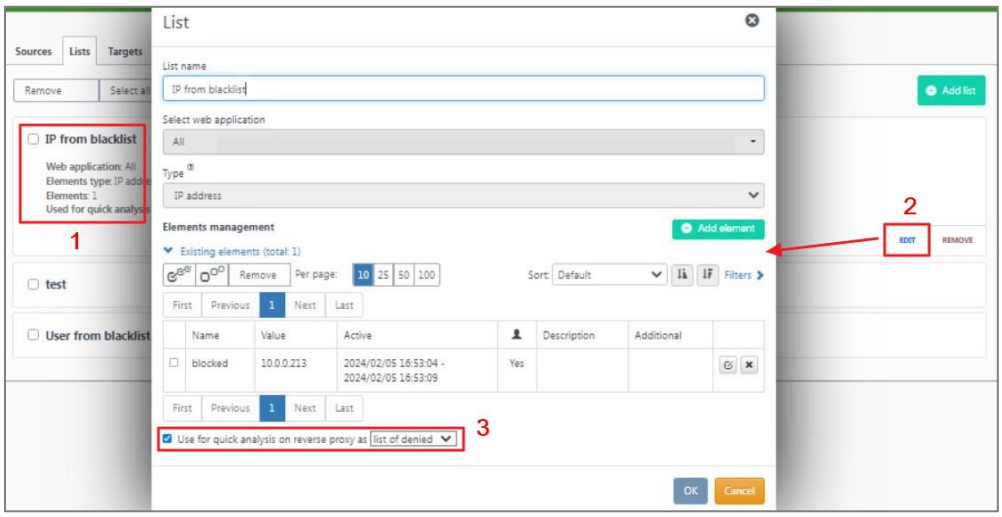
You can see an example of such list usage during blacklist IP addresses creation. First, you need to create a new list.

In the **Type** drop-down list, select **IP address** (see **1** in the figure below). Select the **Use for quick analysis on reverse proxy as** check box (see **2** in the figure below). Select **list of denied** in the drop-down list (see **3** in the figure below).

Add elements to the list — specify the IP address or the range of IP addresses (see **1** in the figure below). If necessary, specify other element parameters (see **2** in the figure below). Click **OK** to save (see **3** in the figure below) the element.



After you click **OK** in the dialog box for creating a list, the new list appears in the tab (see **1** in the figure below). To view and edit it, click **Edit** (see **2** in the figure below). Look through IP addresses and other parameters of the list, make sure the use of the list as the list of prohibited IP addresses on the reverse proxy is excluded (see **3** in the figure below).

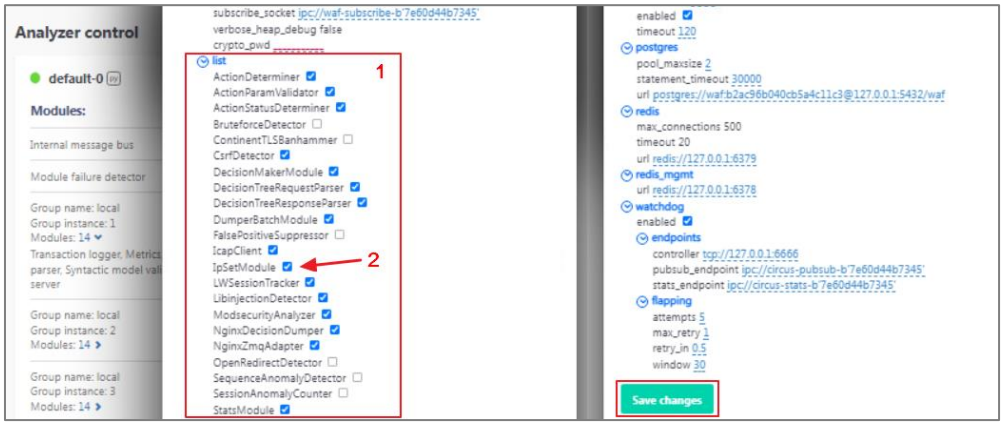


Go to **Settings | Analyzer control** and make sure IpSetModule is enabled (see **1** in the figure below).

Otherwise select the check box next to its name (see **2** in the figure below). Save the new settings revision by clicking **Save changes**.

The new list of prohibited IP addresses will be transferred to the reverse proxy after a few minutes or after the analyzer restarts.

If IpSetModule was disabled, the settings will change and it will be enabled after the analyzer restarts.



Before the new settings apply, the transaction list displays missed Continent WAF requests from two different IP addresses.

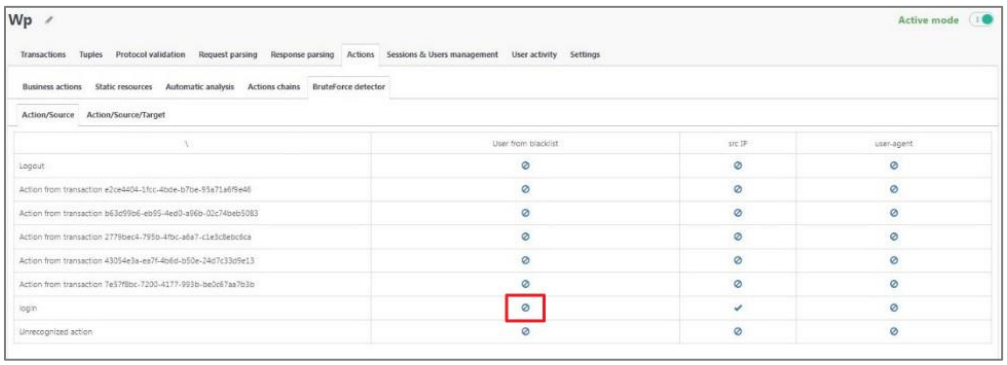
After the new settings are applied, requests to the web application from the blacklisted IP address are blocked.

When you view one of such blocked transactions, the anomaly from the NginxDecisionDumper module is displayed. The anomaly description states that this IP address is blacklisted and indicates the UUID of the respective list.

Using sources in response rules

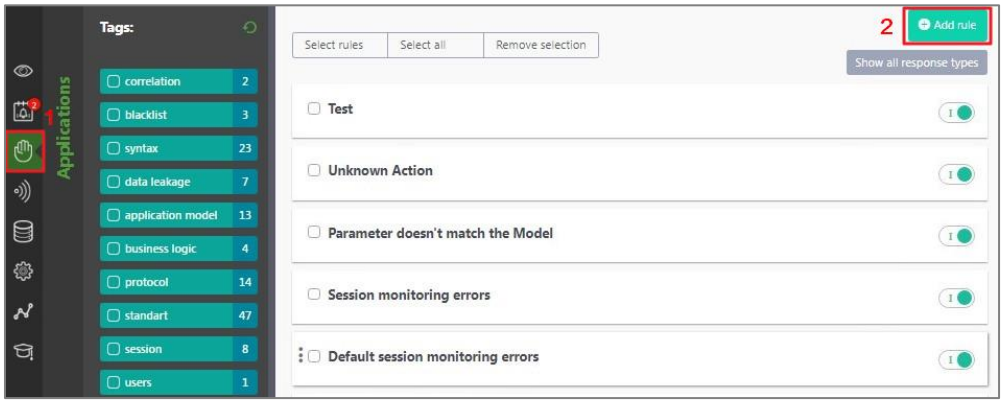
You can use sources in response rules as a possible element that is checked by the rule specification.

For example, the source is **User from blacklist**. In this example, the rate limiting settings of bruteforce detector are disabled for this source.



When you try to log on to the web application using the blacklisted user name, for example, **guest**, the transaction list shows that Continent WAF passed this request. When you view the transaction details, the sources include **User from blacklist** with the **guest** value.

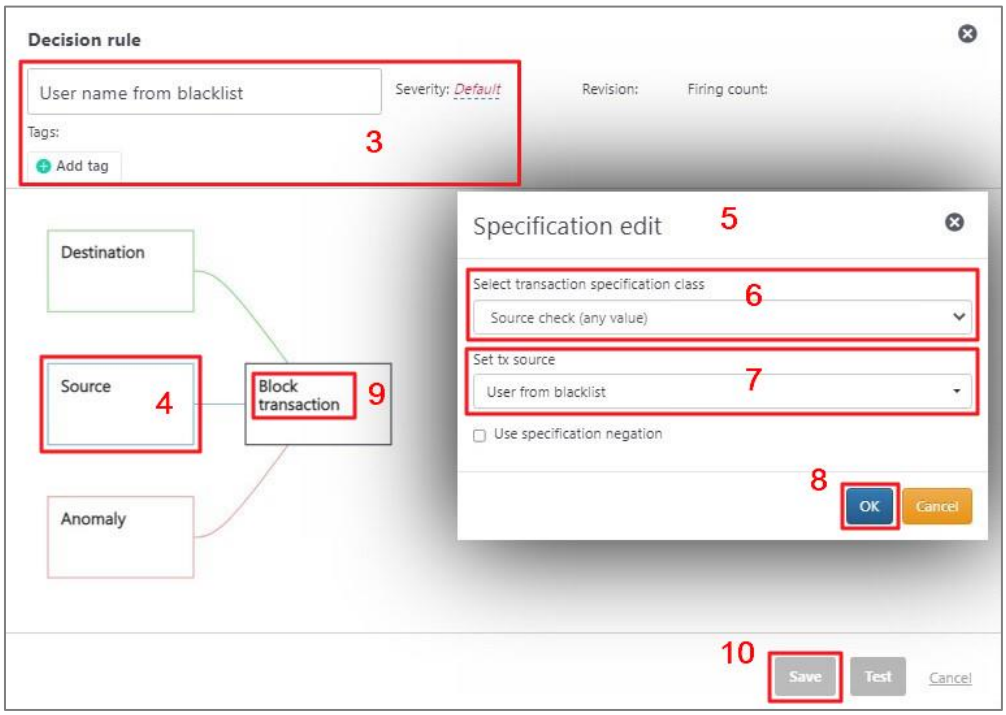
Create a response rule. To do so, go to **Rules** (see **1** in the figure below) and click **Add rule** (see **2** in the figure below).



In the appeared dialog box, specify the name and other parameters (for example, tag list) of the new rule (see **3** in the figure below). Click the **+** icon in the **Source** cell (see **4** in the figure below) to set the source specification for this rule. The **Specification edit** dialog box appears (see **5** in the figure below).

Select the transaction specification class in the drop-down list (see **6** in the figure below), in this example — **Source check (any value)**. Specify **User from blacklist** as the source (see **7** in the figure below) by selecting it in the drop-down list.

To save the transaction source specification, click **OK** (see **8** in the figure below). Set **Block transaction** as the action for the rule (see **9** in the figure below). To save the created response rule, click **Save** (see **10** in the figure below).



The created rule appears in the response rule list (see **1** in the figure below). Its specification (see **2** in the figure below) indicates that the transaction will be blocked if it contains the **User from blacklist** source.

Right after you create the rule, it is disabled by default. Enable this rule by turning on the toggle opposite the rule name (see **3** in the figure below).

It may take some time to upload the new rule from the database to the analyzer. When a blacklisted user (for example, **guest**) attempts another request, the response rule blocks this request.

Administrator work scenarios

The administrator can perform analyst tasks. Besides, the administrator performs the following tasks:

- Continent WAF installation;
- updating;
- backup;

- creating and configuring applications;
- creating Continent WAF users;
- analyzer restart;
- audit of Continent WAF user actions.

For detailed information about the functions, see [**1**].

Documentation

1. Continent WAF. Version 2. Administrator Guide