SECURITY code

Continent WAF Version 2

Administrator 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 administrators of Continent WAF, Version 2 (hereinafter — Continent WAF). It contains information about the installation and confiduration 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.

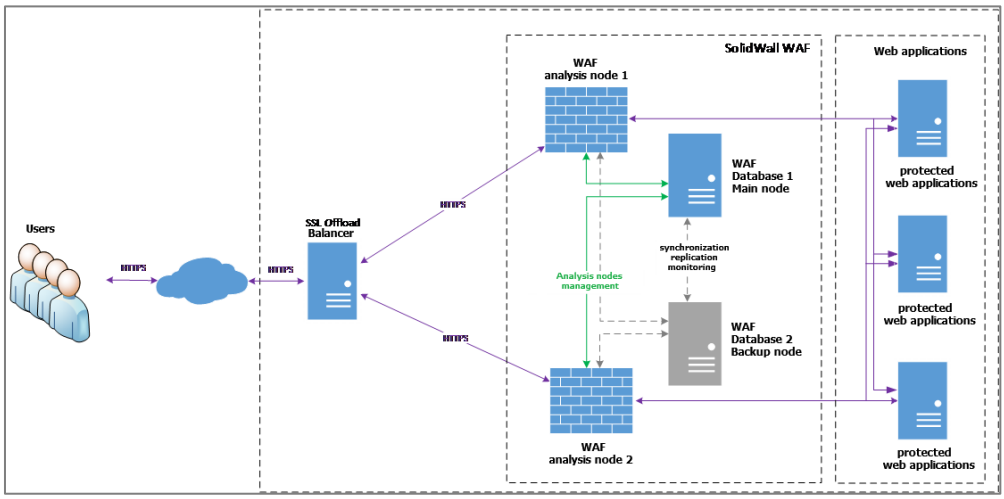
Continent WAF performs the following:

- 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 lcase of contingencies.

Qualifications:

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

A user account is assigned the read only role.

System requirements

The recommended system requirements are presented 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
Install Continent WAF

Continent WAF comes in several modifications: an appliance, a virtual machine or a distribution kit on a removable drive. In the case of an appliance or a virtual machine, Continent WAF is installed and set up by the developer's representatives.

Get ready for Continent WAF installation

If you use Astra Linux, you need to create a new en_US.UTF-8 locale before the installation. Uncomment **#** in front of **ofen_US.UTF-8 in /etc/locale.gen** and run the **locale-gen** command as an administrator.

The distribution kit comes in a **.tgz** archive taking up about 1 GB.

To prepare for Continent WAF installation:

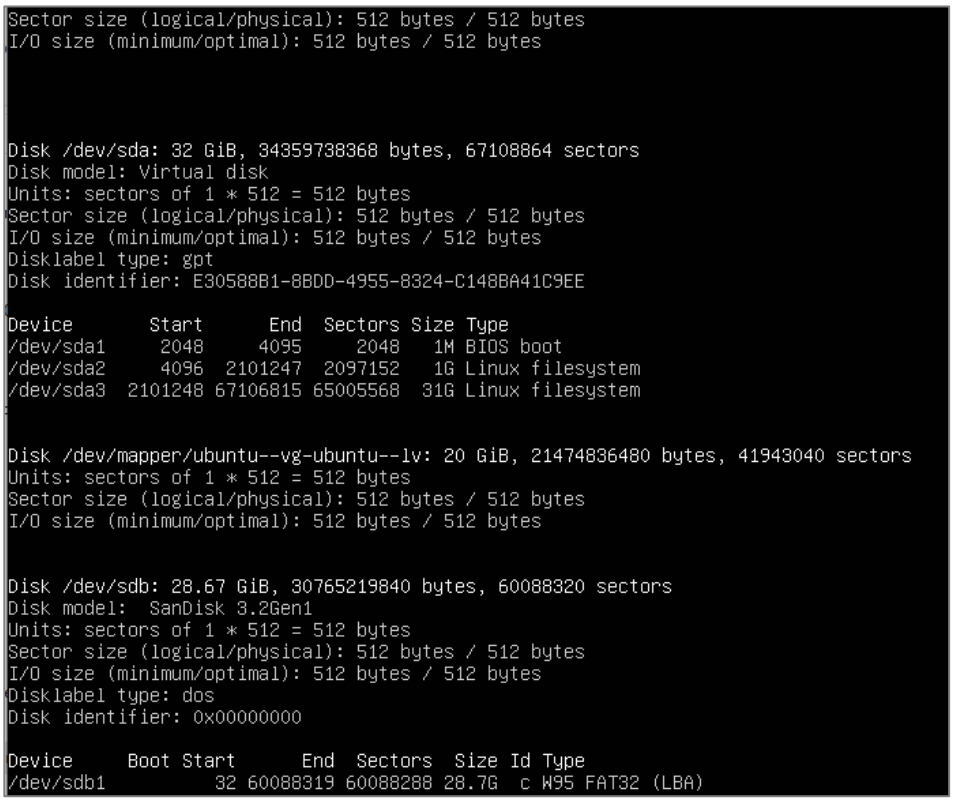
1. Create an OS user with administrator rights. Specify **waf** for both their username and password.
2. Plug in a USB flash drive with the Continent WAF distribution kit.
3. Create a folder to mount the USB flash drive by running the command:

```
sudo mkdir /mnt/usb
```

4. Find out the ID of the USB flash drive by running the command:

```
sudo fdisk -l
```

Now you can see a list of disks with their IDs.



```
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes



Disk /dev/sda: 32 GiB, 34359738368 bytes, 67108864 sectors
Disk model: Virtual disk
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: gpt
Disk identifier: E30588B1-8BDD-4955-8324-C148BA41C9EE

Device       Start      End  Sectors Size Type
/dev/sda1     2048     4095     2048   1M BIOS boot
/dev/sda2     4096  2101247  2097152   1G Linux filesystem
/dev/sda3  2101248 67106815 65005568  31G Linux filesystem


Disk /dev/mapper/ubuntu--vg-ubuntu--lv: 20 GiB, 21474836480 bytes, 41943040 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes


Disk /dev/sdb: 28.67 GiB, 30765219840 bytes, 60088320 sectors
Disk model:  SanDisk 3.2Gen1
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x00000000

Device     Boot Start      End  Sectors  Size Id Type
/dev/sdb1          32 60088319 60088288 28.7G  c W95 FAT32 (LBA)
```

5. Mount the USB flash drive by running the following command where **/dev/xxx** is the ID of the USB flash drive (for example, **/dev/sdb1**):

```
sudo mount /dev/xxx /mnt/usb
```

6. Copy the distribution kit archive to **/home/waf** by running the command:
   - Without a cluster:

```
cp /mnt/usb/scwaf-distrib-ubuntu-20.04-master-deployment--compliant.tgz /home/waf
```

- For a cluster:

```
cp /mnt/usb/scwaf-distrib-ubuntu-20.04-master-deployment--compliant-replication.tgz /home/waf
```

7. Unmount the USB flash drive by running the command:

```
sudo umount /dev/xxx
```

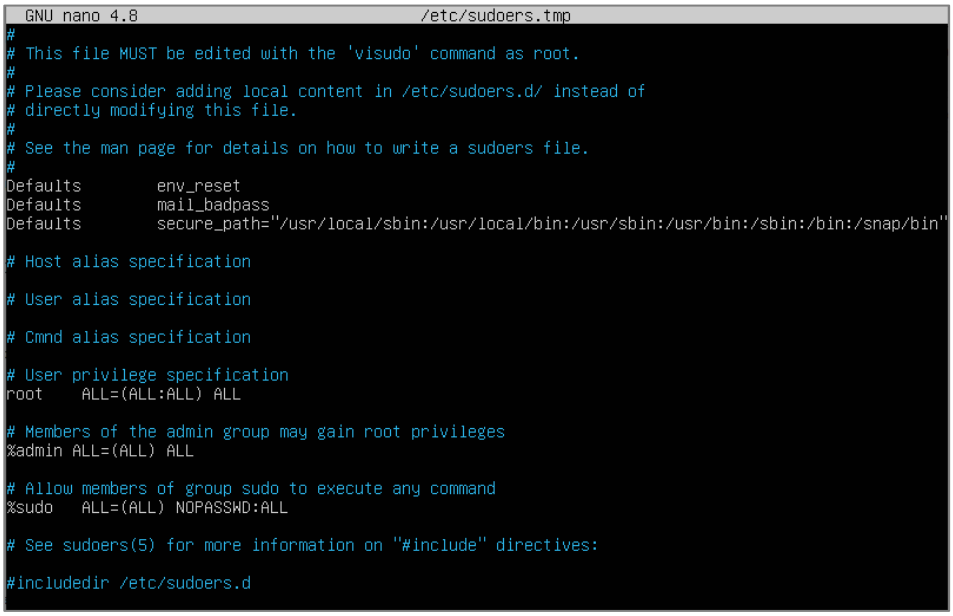
8. Set the **sudo** mode for the **waf** user by running the command:

```
sudo visudo
```

The **/etc/sudoers.tmp** file opens. After the line **#Allow members of group sudo to execute any command**, replace the line with the following one:

```
% sudo ALL=(ALL) NOPASSWD:ALL
```

For an example, see the figure below.



To exit, press <**Ctrl**>+<**X**>. Then confirm the changes by pressing <**Y**> and confirm the file name by pressing <**Enter**>.

9. Unpack the distribution kit archive by running the command:

- Without a cluster:

```
tar xf scwaf-distrib-ubuntu-20.04-master-deployment--compliant.tgz /home/waf
```

- For a cluster:

```
tar xf scwaf-distrib-ubuntu-20.04-master-deployment--compliant-replication.tgz /home/waf
```

Attention! Only the user with the name **waf** can install Continent WAF. The **root** user must not install the software.

Set up the traffic capture interface (only for passive mode)

Note. All commands in this section require root privileges.

Let **ethX** be the name of the interface on which traffic should be captured, where **X** is the ordinal number of an interface (for example, **eth0**). Make sure the interface is configured in the **/etc/netplan/00-installer-config.yaml** file.

Note. The name of the interfaces may vary, an example is given for a virtual implementation.

```
# This is the network config written by 'subiquity'
network:
  ethernets:
    ens160:
      addresses:
        - 172.16.10.1/24
    ens192:
      addresses:
        - 172.16.20.1/24
    ens224:
      addresses:
        - 172.16.30.1/24
  version: 2
  renderer: networkd
```

By running the **ifconfig ethX** command, make sure that the required interface is up and running.

The words **UP** and **RUNNING** must be present in the command output.

```
waf@waf:/$ ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0c:29:d8:fe:61
          inet addr:172.16.8.132  Bcast:172.16.8.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fed8:fe61/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:986546 errors:0 dropped:92585 overruns:0 frame:0
          TX packets:222701 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:188585678 (188.5 MB)  TX bytes:269264085 (269.2 MB)
```

If the interface is not up and running, run the **ifup ethX** command.

Start installing Continent WAF

To start the installation:

1. Go to the directory with the installer by running the command:

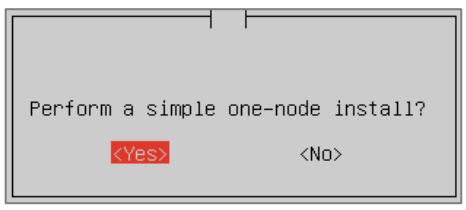
```
cd scwaf-distrib
```

2. Run the command (without **sudo**):

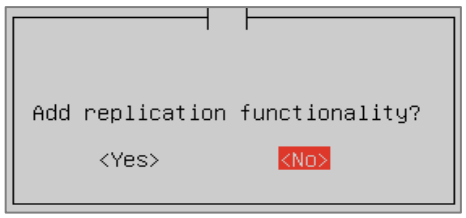
```
./install_interactive.sh
```

The installation starts. You are prompted to select the installation options:

- Without a cluster:



- For a cluster:



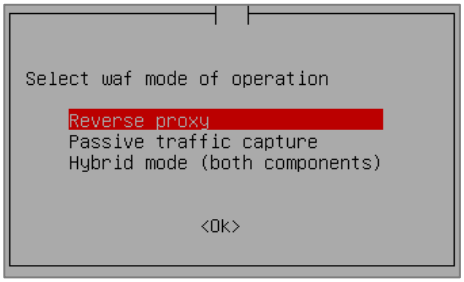
There are the following installation options available:

1. Simple installation in active mode (see below).
2. Simple installation in passive mode (see p. **12**).
3. Advanced installation (not recommended, see p. **13**).
4. Installation in a cluster (see p**. 16**).

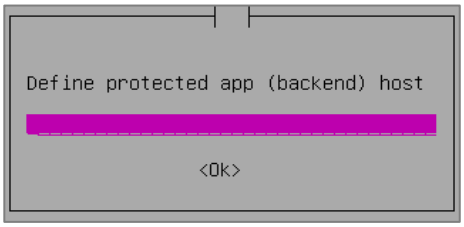
Simple installation in active mode

To install Continent WAF in active mode:

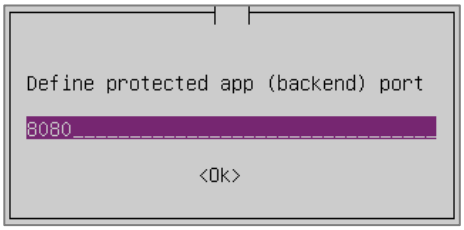
1. Select <**Yes**> for **Perform a simple one-node install?** and press<**Enter**>.
   The following window appears:



2. Select **Reverse proxy** and press <**Enter**>.
   The following window appears:



3. Specify the IP address of the web application you want to protect and press <**Enter**>.
   The following window appears:

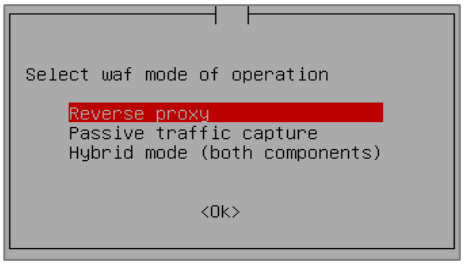


4. Specify the port and press <**Enter**>.
   Wait for the installation to be completed. It might take 10 –15 minutes.

Simple installation in passive mode

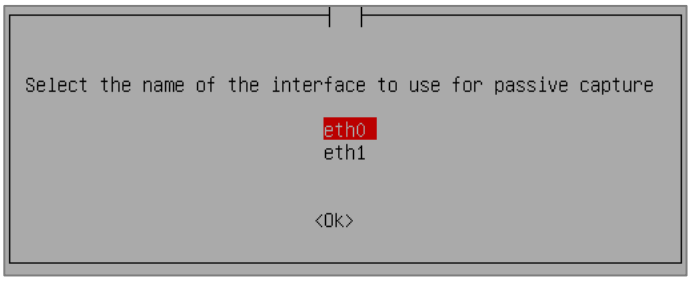
To install Continent WAF in passive mode:

1. Select <**Yes**> for **Perform a simple one-node install?** and press<**Enter**>.
   The following window appears:



2. Select **Passive traffic capture** and press <**Enter**>.

   Depending on your network configuration, the following window can appear:



3. Select the interface to capture traffic and press <**Enter**>.

   Wait for the installation to be completed. It might take 10–15 minutes.

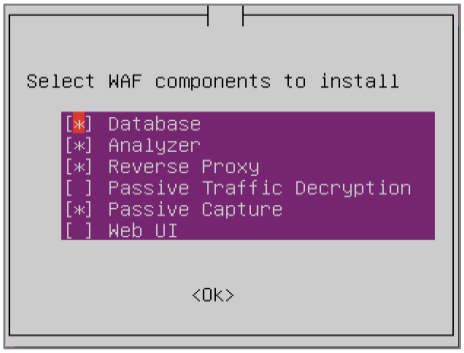
Advanced installation (not recommended)

Advanced installation provides more options for configuring Continent WAF software.

For advanced installation of Continent WAF:

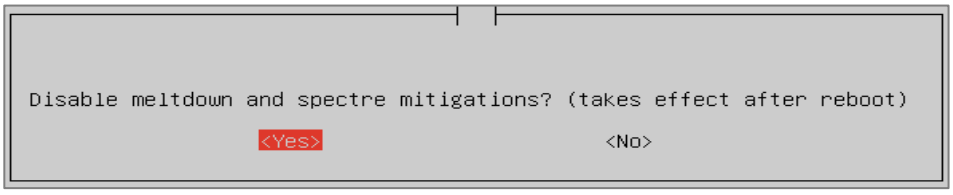
1. Select <**No**> for **Perform a simple one-node install?** and press<**Enter**>.

   The following window appears:



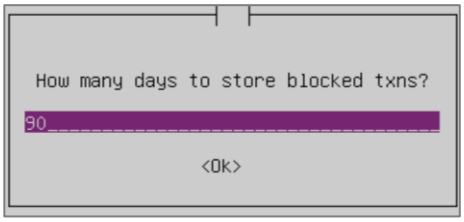
2. Select Continent WAF components to be installed. Use the space key to select and unselect the components. Press <**Enter**>.

   You are prompted to disable meltdown and spectre mitigations.



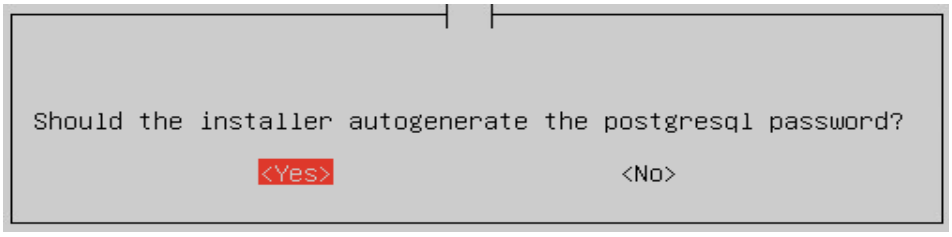
3. Select <**Yes**> or <**No**> depending on your information security requirements and press <**Enter**>.

   The following window appears:

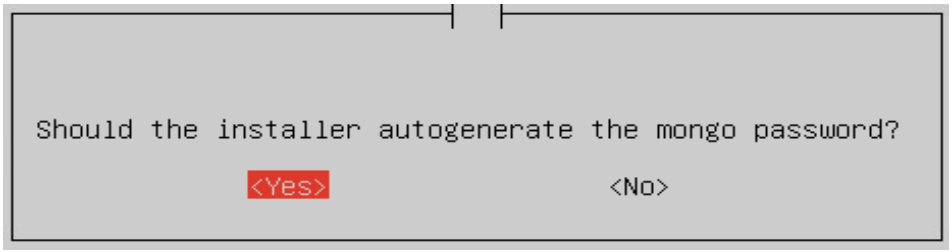


4. Specify how long you want to store blocked transactions and press <**Enter**>.
   The following window appears:

5. Specify how long you want to store correct transactions and press <**Enter**>.
   The following window appears:

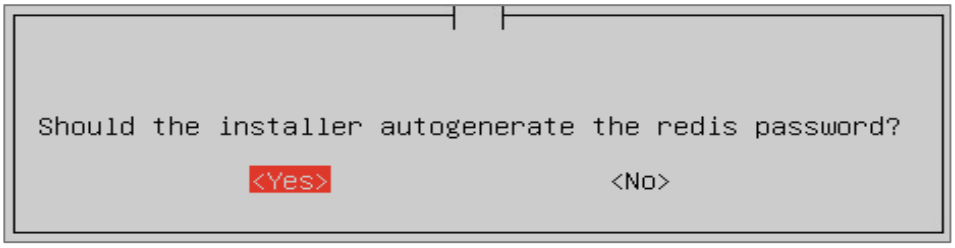


6. Choose whether you want to autogenerate a password for Postgresql. We recommend using autogenerated passwords. Press <**Enter**>.
   The following window appears:



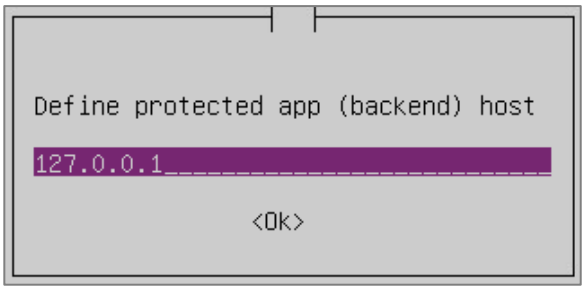
7. Choose whether you want to autogenerate a password for Mongo. We recommend using autogenerated passwords. Press <**Enter**>.
   The following window appears:

8. Specify the analyzer name and press <**Enter**>. The following window appears:



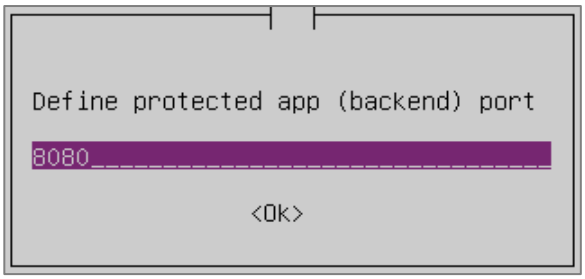
9. Choose whether you want to autogenerate a password for redis. We recommend using autogenerated passwords. Press <**Enter**>.

   The following window appears:



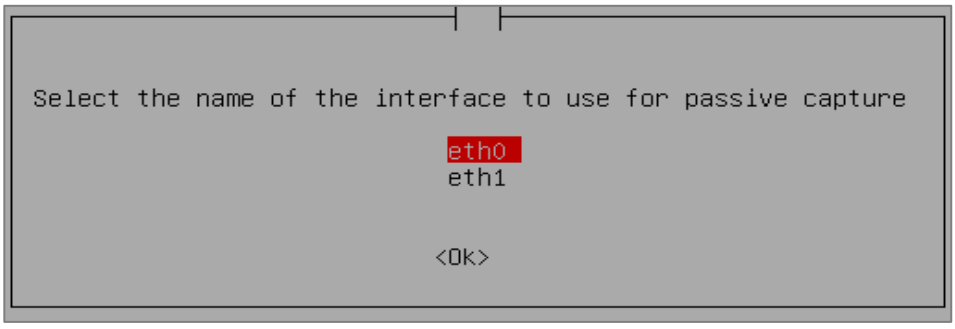
10. Specify the IP address of the web application you want to protect and press <**Enter**>.

    The following window appears:



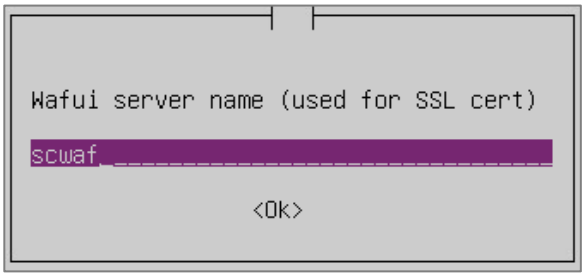
11. Specify the port and press <**Enter**>.

    Depending on your network configuration, the following window can appear:

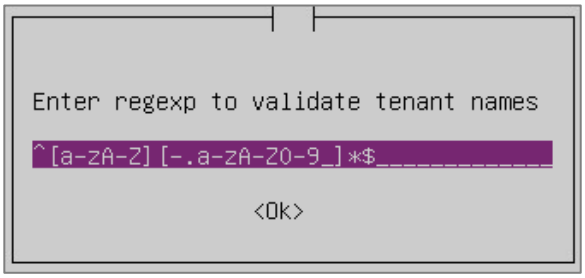


12. Select the interface to capture traffic and press <**Enter**>.

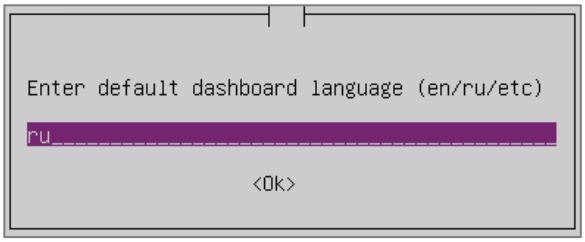
    The following window appears:



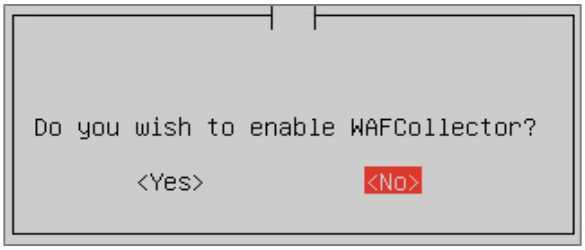
13. Specify the server name and press <**Enter**>.
    The following window appears:



14. Specify regular expressions and press <**Enter**>.
    The following window appears:



15. Select the default interface language (en/ru/etc) and press <**Enter**>.
    The following window appears:



16. Choose whether you want to enable WAFCollector and press <**Enter**>.
    Wait for the installation to be completed. It might take 10–15 minutes.

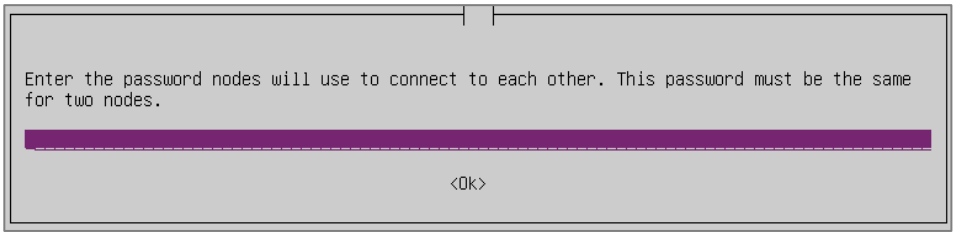
Installation in a cluster

Make sure there is network connectivity between nodes.

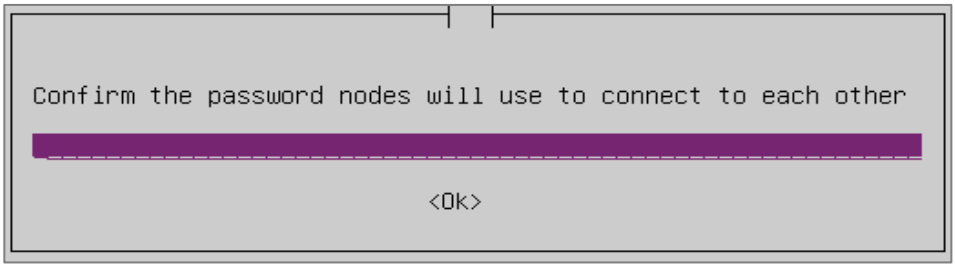
To install Continent WAF in a cluster:

1. On the first node, select <**Yes**> and press <**Enter**>.
   The following window appears:

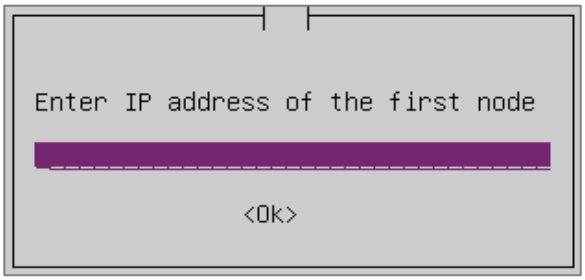
2. Select master and press <**Enter**>.
   The following window appears:



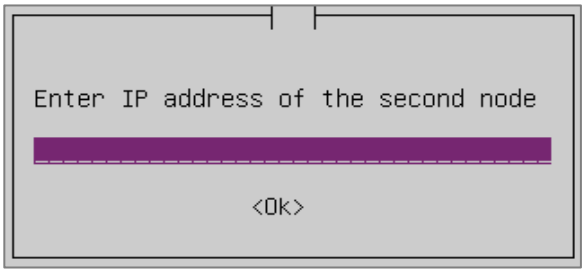
3. Enter the password to connect two nodes to each other and press <**Enter**>.
   The following window appears:



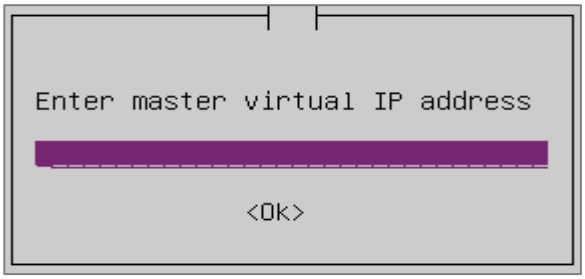
4. Confirm the password and press <**Enter**>.
   The following window appears:



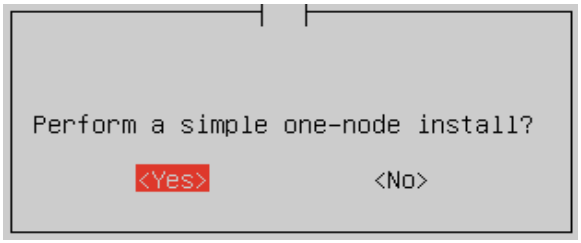
5. Specify the IP address of the first node and press <**Enter**>.
   The following window appears:



6. Specify the IP address of the second node and press <**Enter**>.
   The following window appears:



7. Specify the virtual IP address of the first node and press <**Enter**>.
   The following window appears:



8. Perform the simple installation in active mode (see p. **12**), in passive mode (see p. **12**) or the advanced installation (not recommended, see p. **13**).
9. Restart the first node.
10. Make sure that all the processes are up and running on the first node by running the command:

```
sudo crm status
```

11. Repeat steps **1–8** on the second node.
12. Restart the second node.
13. Make sure that all the processes are up and running on the second node by running the command:

```
grep -A 1 RECAP /var/tmp/install.log
```

Continent WAF installation last steps

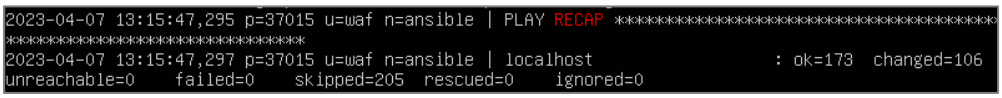
During the installation, you can see many diagnostic messages that are copied to the **/var/tmp/install.log** file.

In the case of an installation failure, send this file to technical support for further analysis.

To check if the installation was a success, run the command:

```
grep -A 1 RECAP /var/tmp/install.log
```

If the message PLAY RECAP with zero failed steps appears, that means that the installation was a success.



Chapter 3
Initial setup

Configure operation mode

Continent WAF can work with network traffic in two modes:

- traffic copy mode (receives traffic from the switch SPAN port and registers attacks and abnormal behavior);
- inline installation mode with the block option (reverse proxy server).

You can select Continent WAF operation mode during the installation, which is described in detail in [**1**].

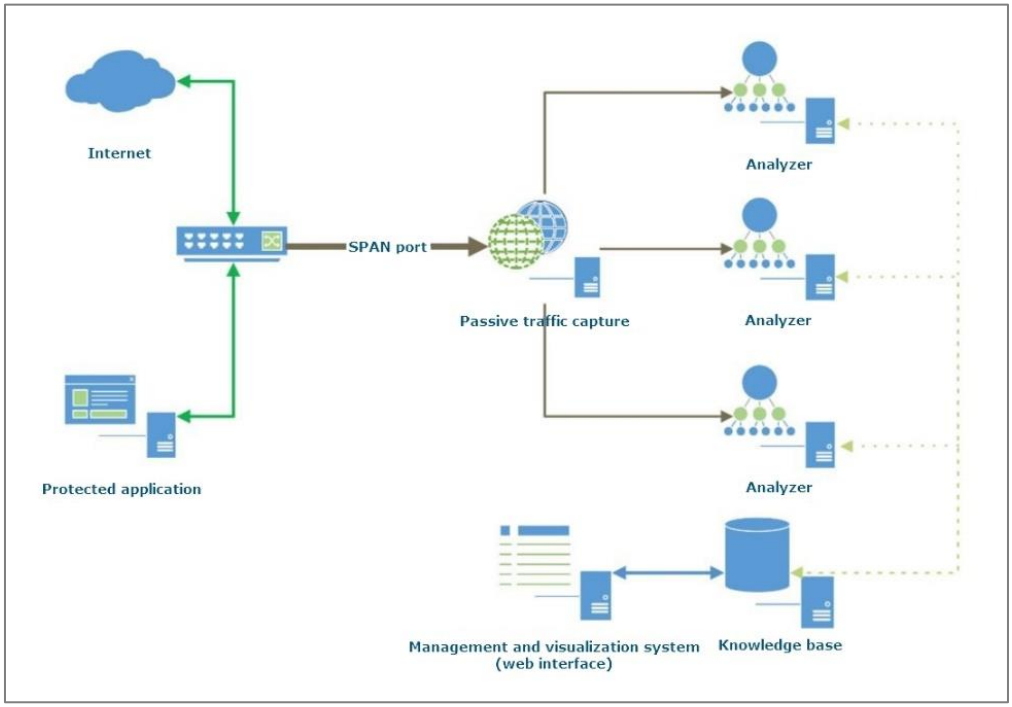
Note. If necessary, you can change Continent WAF operation mode after the installation and save the collected data (without fully reinstalling the software) by contacting technical support.

To set the operation mode, run the **cd /usr/local** command, and then enter **ls**. The component list appears. If nginx is on this list, then Continent WAF is initialized in active mode, if nginx is not on the list — in passive mode.

Traffic copy mode (passive mode)

The main feature of this mode is the fact that operability and availability of protected apps does not depend on the Continent WAF operability. Continent WAF receives a copy of all traffic from a network device (for example, a switch or a network load balancer) via a SPAN port.

The scheme for integrating Continent WAF in traffic copy mode into some infrastructure is given in the figure below.



In monitoring mode, passive traffic analysis without the option of blocking requests is performed. For Continent WAF to operate in monitoring mode, you need to configure the network interface to which the traffic copy will be sent by using basic OS tools before the start of the installation procedure.

Note. All commands in this section require root privileges.

Let ethX be the name of the network interface on which the traffic should be captured. You need to make sure that this interface is configured in the **/etc/netplan/00-installer-config.yaml** file. If the configuration is missing, add the following strings at the end of the file:

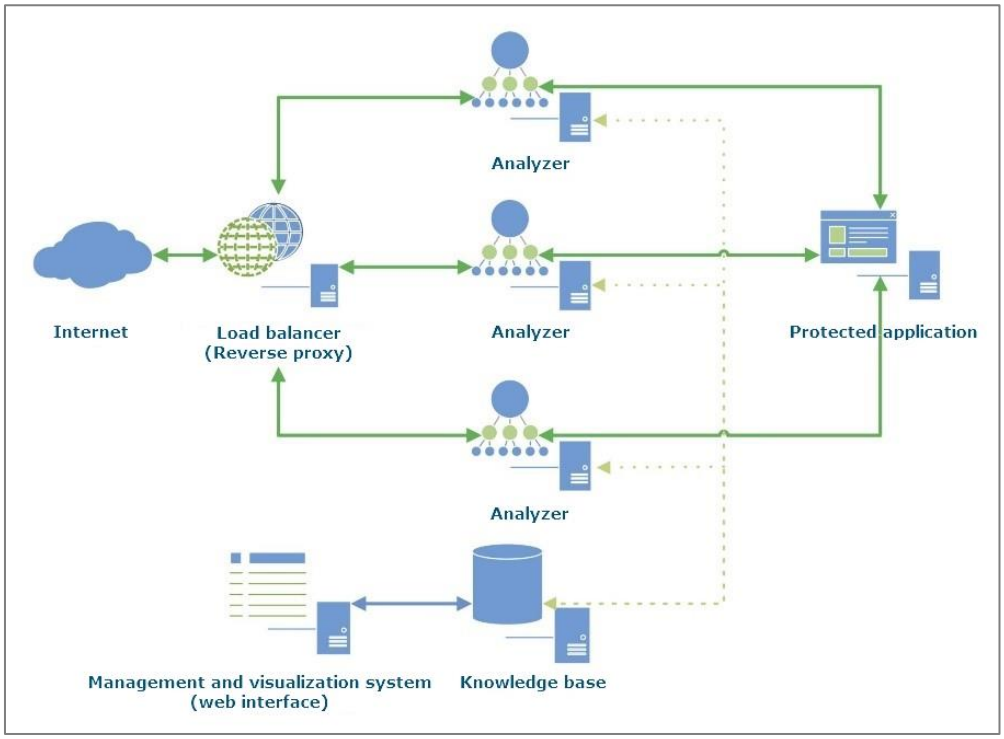
```
auto ethX

iface ethX inet manual
```

You need to check whether the required interface is in the active state using the **ifconfig ethX** command. The command output must contain the words **UP** and **RUNNING**. If the interface is not in the active state (**DOWN**), you need to activate it using the **ifup ethX** command.

Active mode (reverse proxy)

In this mode, incoming traffic passes through Continent WAF which provides the option to affect HTTP requests and responses (block requests, change requests and responses). You can see a scheme for inline integration of Continent WAF into some infrastructure with the option to block attacks.



For Continent WAF to operate in reverse proxy server mode, you need to configure the network interface (interfaces) to which the requests for analysis will be sent in the following way by using basic OS tools before the start of the installation procedure:

- Clients must have network access to the external IP and Continent WAF port (external_ip, external_network) via the external network and external interface.
- Continent WAF must have network access to the IP and port of the protected app. If there are two different interfaces in use, access to the protected app must be performed via the internal interface.

Note. Despite Continent WAF inline installation, the monitoring mode, which does not perform the attack blocking, is set by default for apps created in the system. You can switch to the blocking mode for each protected app separately in the **Webapps** section in the main menu of the management console.

Continent WAF setup order

Continent WAF processes HTTP traffic destined for web apps.

First of all, you need to create a profile for each protected app in the management console. This profile defines rules for requests and responses between the user browser and the web app. For Continent WAF to start processing the application traffic, you need to create an application and add respective tuples.

Then, you need to configure rules and destination addresses to receive notifications about detected attacks.

Information about violations in real time and via reports is sent to the console and email. After looking through these reports, the security administrator can decide to take additional measures to block specific IP addresses, users or networks, edit the app profile, enable, disable or modify response rules and suppress false positives.

Continent WAF processes traffic based on the following principles:

- for an incoming HTTP request, a protected app is determined based on its tuple settings;
- analysis modules process the request according to the application profile, including syntax and structure parsing, identifying actions and their parameters, determining whether the request belongs to a specific session, etc.;
- analysis modules may generate anomalies during processing;
- based on the set of response rules and all the information about the request, including anomalies, a decision whether to block the request or pass it to the protected app is made.

HTTP responses are processed the same way.

Continent WAF has a preinstalled set of rules with the **preinst** tag. If one of the rules is not working correctly, you can disable it by clicking **On** to the right of the rule name.

When web application vulnerabilities are detected, the administrator creates rules to block attempts to exploit these vulnerabilities (virtual patching). If the administrator needs to manually create an allowing or denying rule to process HTTP transactions, they need to go to the **Rules** section in the main menu of the management console and click **Add rule** in the top right corner.

Add a nginx configuration file

Set up proxy

All configuration files must be created in the **/etc/scwaf-nginx/sites-available** directory. Keys and TLS certificates must be stored in the directory **/etc/scwaf-nginx/ssl.** We recommend creating a new configuration file with the application name. The file contains one or several **server** sections. An example of the file is given below:

```
server {
     include static/server.conf; listen 80;
     server_name example.com www.example.com;
     # WAF configuration.
     set $BACKEND http://192.168.1.100;
     location / {
        include static/waf.conf;
     }
}
server {
     include static/server.conf; listen 443 ssl;
     ssl_certificate ssl/cert.crt;
```

```
    ssl_certificate_key ssl/cert.key;

    server_name example.com www.example.com;

    # WAF configuration.

    set $BACKEND https://192.168.1.100:443;

    location / {

       include static/waf.conf;

    }
}
```

The **listen** value contains parameters with the port number to which the backend will be proxied. The **ssl** parameter means that traffic must be available through HTTPS.

The **ssl_certificate*** value contains the path to files with SSL certificates and keys. It is necessary in order to terminate encrypted TLS traffic before it reaches Continent WAF.

The **server_name** value contains domain names to which requests will be sent. Note that you need to specify all the domain names for an application, for example, www.example.com, example.com. Otherwise, some requests will be processed incorrectly.

The **set $BACKEND** value contains the full address of backend with the protocol and port number.

Apply configuration

After creating a configuration file, you need to create a symbolic link to it in the **/etc/scwaf-nginx/sites-enabled/** directory for its activation. For example:

```
ln -s /etc/scwaf-nginx/sites-available/example.com

/etc/scwafl-nginx/sites-enabled/example.com
```

Then, you check the syntax by running the command:

```
scwaf-nginx test
```

If you receive a response like in the figure below, you can apply the configuration.

```
scwaf-nginx reload
```

```
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
```

Check proxy

To check proxy, send requests to a protected web application directly and through Continent WAF. In both cases, the responses must be the same. You can send requests any way you like. For example, you can use a web browser. To do that, allow the domain name of the web application in the **hosts** file. Alternatively, you can use the curl tool in the WAF server console.

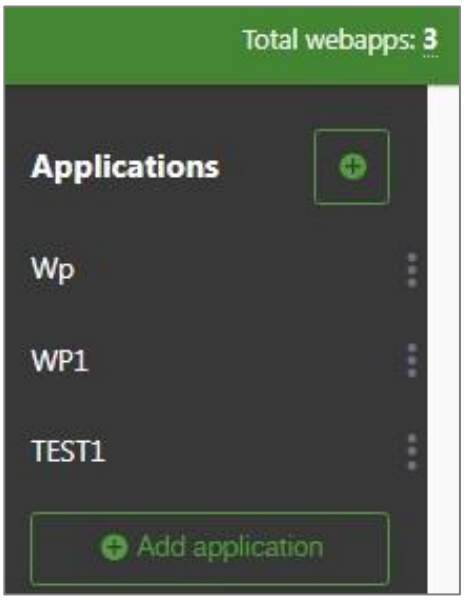
Chapter 4
Operations with protected apps

Create an application profile

A profile is an object that allows you to configure Continent WAF to work with a specific protected web app (a website or a number of websites).

To create an application profile:

1. Go to the **Applications** section in the main menu of the management console.
2. In the appeared second level menu**,** click **Add application** or **+** to the right of the title**.**



3. In the appeared dialog box**,** specify the name of the added web app and click **Create**.

The window containing a list of tuples appears (domain, host, port: parameters of traffic not currently assigned to any profile yet).

Note. This window also allows you to edit the name of the created application and to enable/disable the active protection. For detailed information about editing the name and deleting the application, see **Edit application name** and **Delete applications**.

Possible operations with detected tuples:

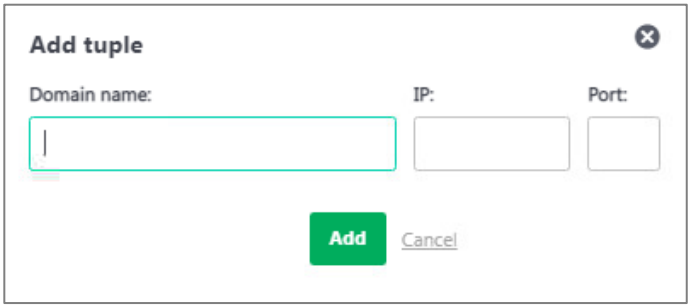
- **Grouping**. You can group tuples using the buttons above the list in the center of the window (the table below).
- **Dynamic search**. You can start entering host name or IP address in the respective field for quick search.

Grouping name	Function
No grouping	Displays all tuples without grouping
Common domain name	Displays tuples grouped by domain name
Common IP	Displays tuples grouped by IP address

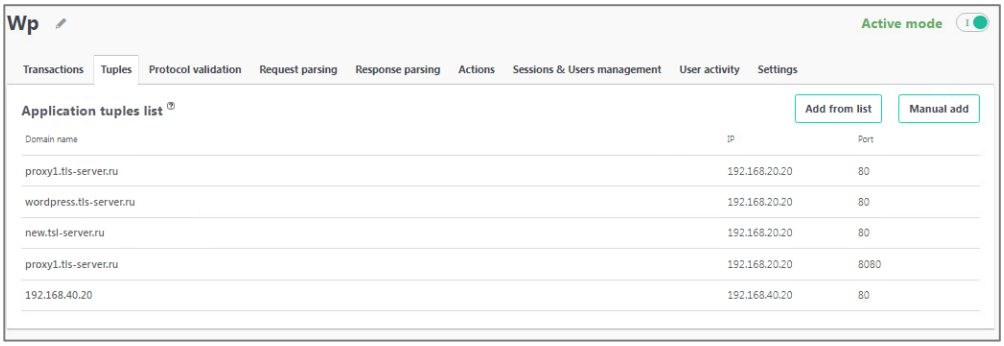
4. Select one or several tuples and click **Add selected**.

   The created application appears in the **Application** list in the second-level menu.

   If there is no required option among the existing, click Add selected. In the next window, select the **Tuples** tab and click **Manual add**. The **Add tuple** window opens.



   Specify the domain name of the protected web app and the port to receive traffic on Continent WAF. The IP address is optional. Do the same for all the domain names and subdomains, as well as ports that were added to the scwaf-nginx configuration file.



Protected app settings

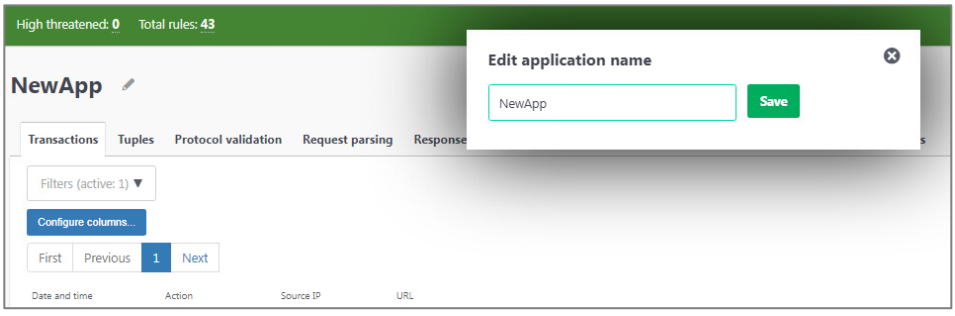
The **Applications** section displays a list of apps registered in the system as well as tools to edit, delete and change Continent WAF operation mode for each of them.

Edit an application name

To edit an application name:

1. Select an application from the menu.
2. Click the  icon to the right of the name.

   The respective dialog box appears.



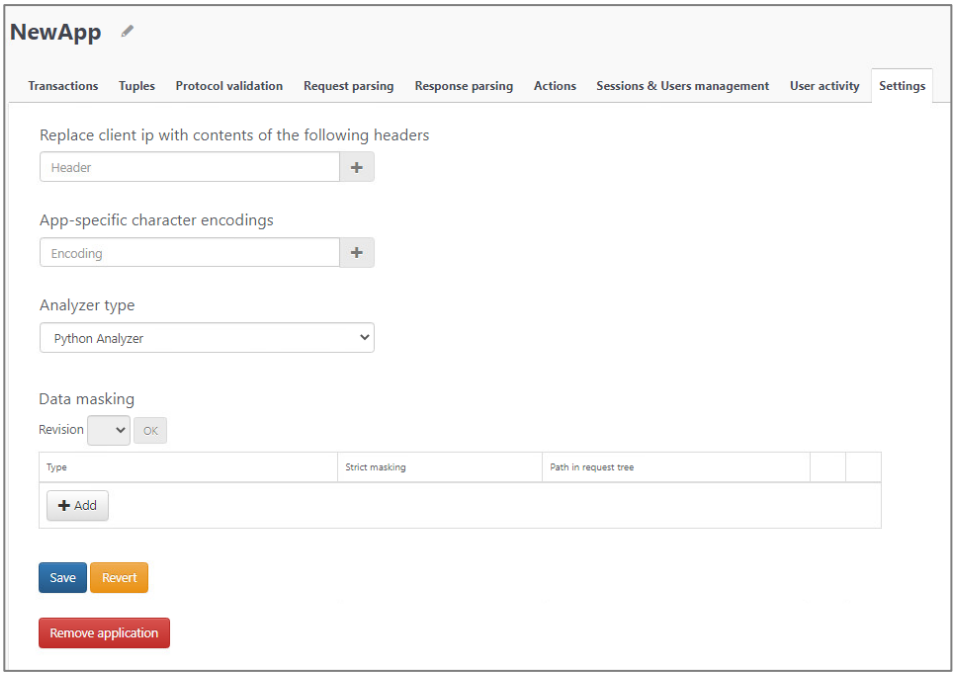
3. Edit the name in the respective field and click **Save**.

Remove an application

To remove an application:

1. Select an application from the list.
2. Go to the **Settings tab** and click **Remove application** at the bottom.

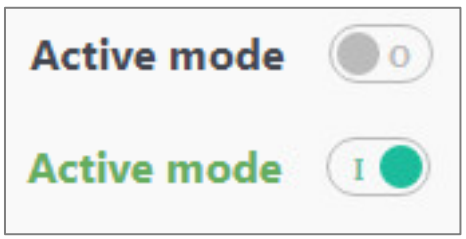
   The system displays a warning message (depending on the web browser used for managing Continent WAF settings, messages may vary).
3. Click **OK**.



Change the Continent WAF operation mode for an application

To enable or disable the active protection mode for an app, select the required app in the list and click the  icon in the top right corner as shown in the figure below

The icon can look different depending on the selected Continent WAF operation mode



Note. When you install Continent WAF in traffic copy mode, the system operates in passive mode regardless of the toggle position.

For detailed information about fine-tuning the application using the tools from the **Transactions** tab, see [**1**].

Configuration version control

Settings configured in **Protocol validation**, **Request parsing**, **Response parsing**, **Actions**, **Sessions & Users management** are saved with version control, meaning if settings are edited, a new revision (version) is created. You can view all settings versions at any time by selecting the version number from a drop-down list.

To apply the selected settings version, click **Reset to original**. A new settings version, which is a copy of the selected version, is created.

Configure proxying of HTTP and HTTPS traffic to a protected application via Continent WAF

In the example below, **app_ip** is a protected web app address available from WAF, **app_port** is a protected app port.

The example uses the following values:

```
app_ip: 127.0.0.1;

app_port: 8080;

application domain name: dvwa;

login: admin;

password: password.
```

To prepare a protected web application:

1. Install the web application using the following command:

```
sudo docker pull vulnerables/web-dvwa
```

2. Run the web application using the following command:

```
docker run --rm -d -p 8080:80 --name dvwa
vulnerables/web-dvwa
```

3. Configure HTTP traffic proxying. To do that, connect to the Continent WAF server via ssh (putty).
4. Make sure the Continent WAF server has network access to the protected app.

```
curl -v http://app_ip:app_port
```

5. Open the scwaf-nginx configuration file for editing in any text editor as a **root** user. For example:

```
sudo vim /etc/scwaf-nginx/sites-enabled/
<application_domain_name>
```

6. Enter settings for proxying HTTP traffic to backend of the protected application. You can see an example of configuration file contents below:

```
server {

   include static/server.conf;

   listen 80;

   server_name <application_domain_name>;
```

```
    # WAF configuration.
    set $BACKEND http://app_ip:app_port;
    location / {
        include static/waf.conf;
    }
}
```

7. After you finish editing, check whether the changes are correct. If there are no errors, restart the scwaf-nginx configuration:

```
sudo scwaf-nginx test
sudo scwaf-nginx reload
```

8. If there are no errors, the new configuration loads and a port on which the scwaf-nginx service listens for incoming connections from the web application clients opens. You can see an example of the **ss -4lnt** command output below:

```
State      Recv-Q Send-Q                                  Local Address:Port
                                  Peer Address:Port
LISTEN     0      100                                         127.0.0.1:6666
                                      *:*
LISTEN     0      128                                         127.0.0.1:6379
                                      *:*
LISTEN     0      128                                               *:80
                                      *:*
LISTEN     0      128                                               *:22
                                      *:*
LISTEN     0      128                                         127.0.0.1:5432
                                      *:*
LISTEN     0      128                                               *:8443
                                      *:*
LISTEN     0      128                                         127.0.0.1:5088
                                      *:*
LISTEN     0      128                                         127.0.0.1:27017
```

9. To check configured settings, you can make a request to Continent WAF using a web browser or curl console web client. You can see an example of the request in the server command line below:

```
curl -H"host:<application_domain_name>" -v localhost
```

10. Configure HTTPS traffic termination. To do that, create a folder to store certificates and keys in:

```
sudo mkdir /etc/scwaf-nginx/ssl
```

11. Generate or receive a key and certificate for SSL termination from the app owner. You can see an example for generating a key and self-signed certificate below:

```
openssl req -x509 -newkey rsa:4096 -keyout key.pem -
nodes -out cert.pem -days 365
```

The output of this command is shown in the figure below.

```
Generating a 4096 bit RSA private key
.....................................................++
.....................................................................
.....................................................................
.....................................................................
.....................................................++
writing new private key to 'key.pem'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:
State or Province Name (full name) [Some-State]:
Locality Name (eg, city) []:
Organization Name (eg, company) [Internet Widgits Pty Ltd]:
Organizational Unit Name (eg, section) []:
Common Name (e.g. server FQDN or YOUR name) []:
Email Address []:
```

12. Place the key and certificate in the created folder:

```
sudo mv cert.pem key.pem /etc/scwaf-nginx/ssl
```

13. Create another scwaf-nginx configuration file:

```
sudo vim /etc/scwaf-nginx/sites-
enabled/<application_domain_name>_ssl
```

You can see an example of configuration file contents below:

```
server {
   include static/server.conf;
   listen 443 ssl;
   ssl_certificate /etc/scwaf-nginx/ssl/cert.pem;
   ssl_certificate_key /etc/scwaf-nginx/ssl/key.pem;
   server_name <application_domain_name>;
   # WAF configuration.
   set $BACKEND http://app_ip:app_port;
   location / {
      include static/waf.conf;
   }
}
```

14. Check and apply the configuration:

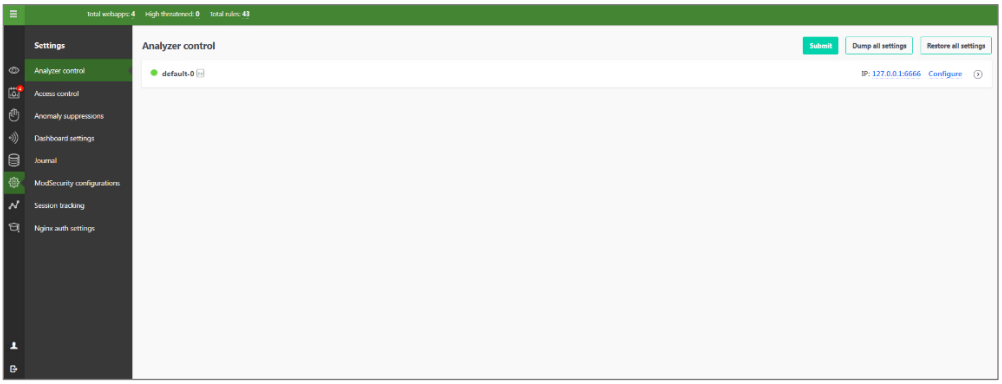
```
sudo scwaf-nginx test
sudo scwaf-nginx reload
```

15. Make sure the application is accessible via Continent WAF:

```
curl -k -v -H"host: dvwa" https://localhost:443
```

Chapter 5
Settings section overview

The main tool of the Continent WAF administrator is the **Settings** section. To switch between the tabs, click the respective pictogram. The general view of the **Settings** section is given in the figure below.



You can do the following in the **Settings** section:

- view the list of analyzers, their states and edit their settings;
- view, add, edit and remove Continent WAF user accounts;
- view the list of anomaly suppressions;
- edit SMTP server settings;
- view the user action log.

This **Settings** section includes the following tabs:

- Analyzer control;
- Access control;
- Anomaly suppression;
- Dashboard settings;
- Journal;
- ModSecurity configurations;
- Session tracking;
- Nginx auth settings.

Analyzer control tab

This tab displays a list of existing analyzers. You can create new analyzers; start, stop, restart, delete and edit the analyzer settings tree. In addition, you can back up the settings of one or all analyzers and restore the settings of a specific analyzer or all analyzers from a backup file.

You can save the settings of the specific analyzer by clicking **Dump**. If you want to save the settings of all analyzers, click **Dump all settings**. The settings are saved according to the browser settings for the downloaded files.

You can restore the settings of a specific analyzer by clicking **Restore**. If you want to restore the settings of all analyzers, click **Restore all settings**.

Add new analyzer

To add a new analyzer:

1. Click **Submit**.

   The **Analyzer** dialog box appears.
2. Specify the parameters listed in the table below.

Filed	Action
Host	Specify the IP address or node domain name on which the analyzer is located
Port	Specify the port of the analyzer. Default value: 6666
Name	Specify the name of the new analyzer

3. Click **Add**.

   A new analyzer appears in the list of analyzers.

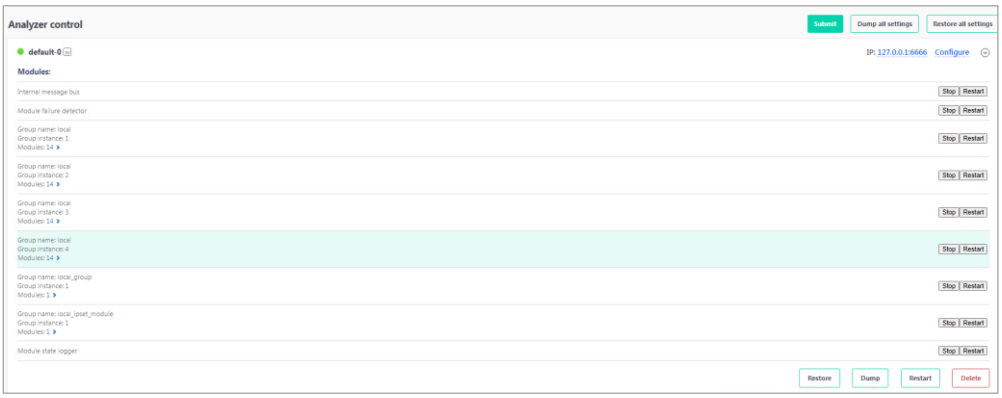
Edit analyzer

The functions available for each analyzer are listed in the table below.

Field/Button	Purpose
IP address field	Change the IP address
Port field	Change the port
> button	View additional information about analyzer state
Configure button	Roll back to the previous configuration or perform the fine-tuning by changing the current settings
State field	View the analyzer state
Delete button	Delete the analyzer
Restart button	Restart the analyzer
Dump button	Create a backup copy of the analyzer configuration and save it to disk in the **config.yaml** file
Restore button	Restore the analyzer configuration from the backup copy. To select a backup copy, click **Browse**

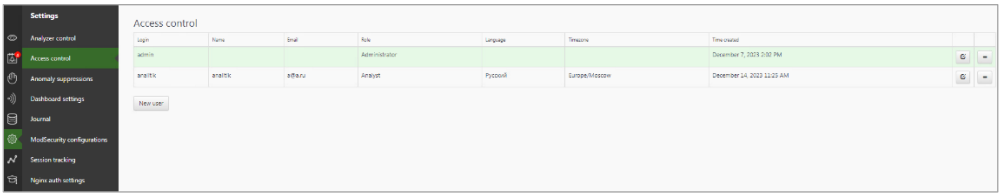
If you click the **>** button, the analyzers additional information is displayed. **Ошибка! Источник ссылки не найден.**

The list of the modules launched in analyzers and their states appear as in the figure below. You can start, stop and restart the modules.



Access control tab

You can create, edit and delete accounts of the users who have access to the Configuration console of Continent WAF in the tab. The general view of the tab is given in the figure below.



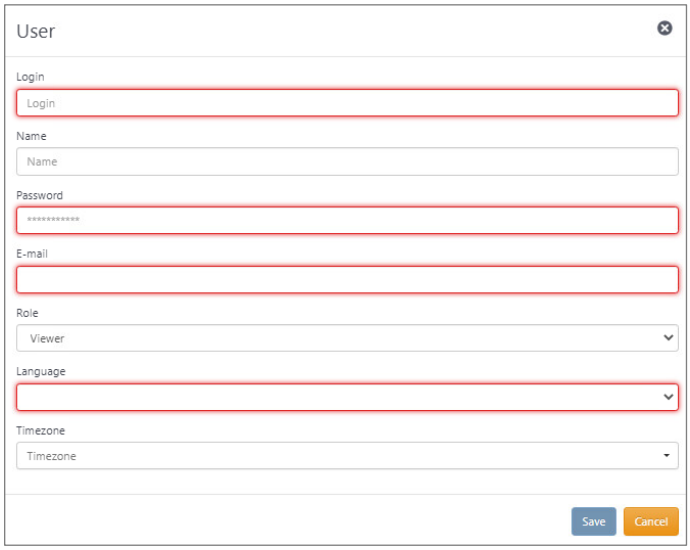
The parameters available for editing and commands supported in the tab are listed in the table below.

Button/Menu command	Purpose
Login field	Change the user login

Button/Menu command	Purpose
Name field	Change the user full name
Email field	Change the user email address
Role field	Change the administrator role (administrator/analyst/user)
Language field	Change the interface language for the selected user
Timezone field	Set the time zone
Time created field	Displays time of the account creation
Password field	Set the user password
 **button**	Edit the user account parameters
 **button**	Delete the user account
Submit button	Apply changes
Cancel button	Discard changes

To add a new user:

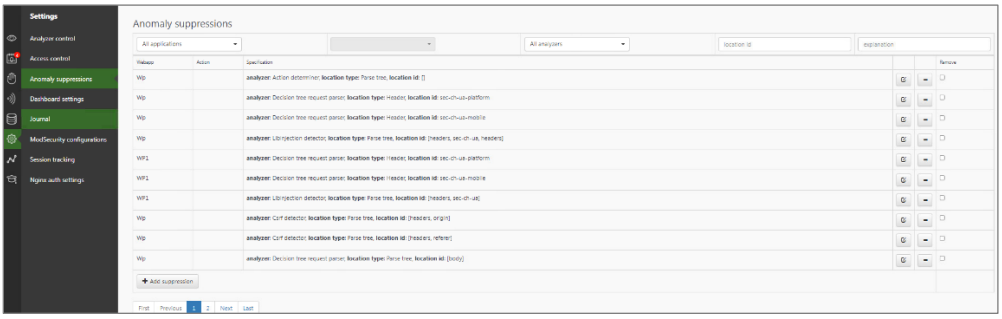
1. Click **New user**.
2. Specify the required information in the respective fields.
3. Click **Submit**.



A new user appears in the **Access control** list.

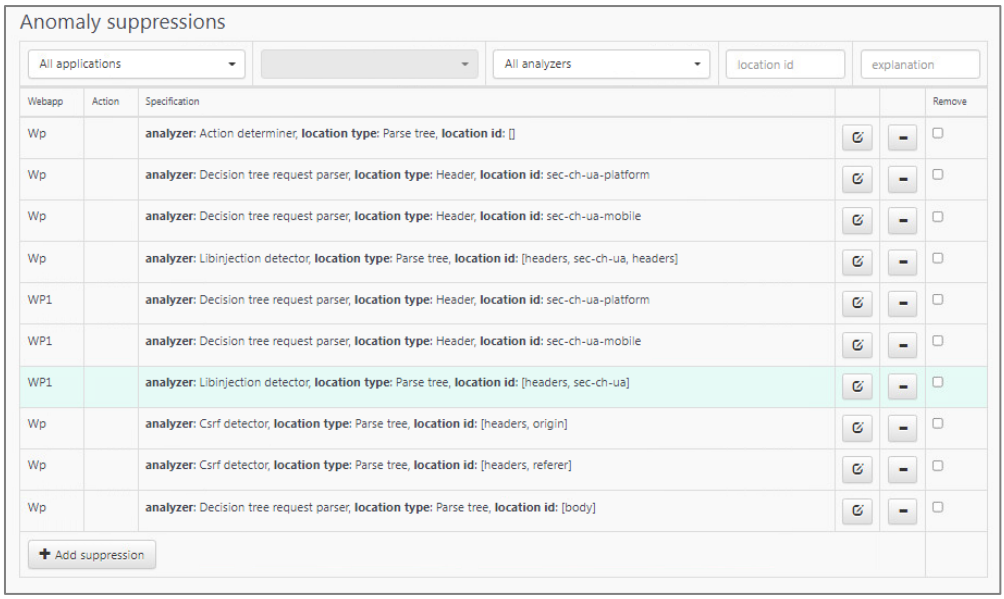
Anomaly suppression tab

You can view, edit, create and delete false positive suppression. The general view of the tab is given in the figure below.



False positive suppression (anomaly suppression) are conditions under which anomalies created during request processing are not considered when deciding whether to block a request or not. Each of these conditions consists of an application, an action and an anomaly specification. For requests to the specified application that are an

instance of the selected action, anomalies that meet the specification are not taken into account. If no application or action is specified, the condition is applied to all actions of all applications or to all actions of a specified application.



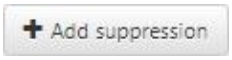
In this section, you can analyze suppressed anomalies, as well as filter them by the following parameters:

- by application;
- by action (becomes active if a target app is selected);
- by analyzer;
- by location type;
- by other data recorded to the database if ModSecurity is triggered or entered in the **Anomaly explanation** field when suppressing anomalies.

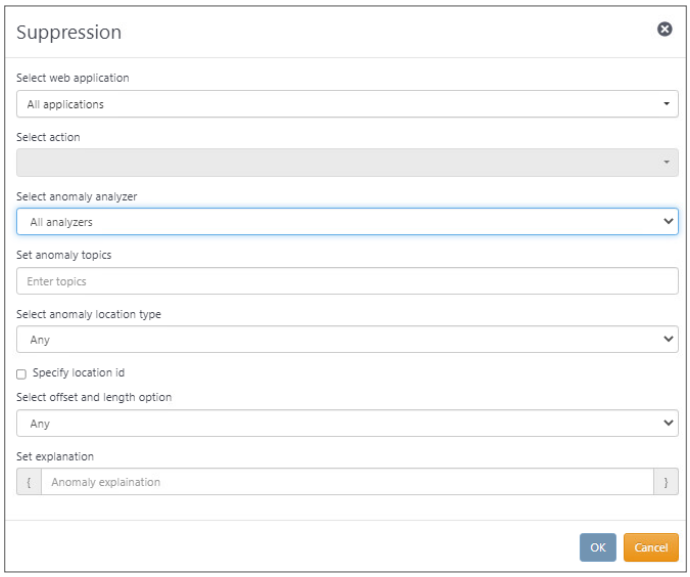
In addition, you can:

- edit already suppressed anomalies;
- remove anomaly suppressions;
- add a new suppression.

To add a new suppression:

1. Click 

button below the **Anomaly suppressions** list.
The **Suppression** dialog box appears as in the figure below.



2. Select the required options from the drop-down lists and specify the required parameters.
3. Click **OK**.

   A new suppression appears in the **Anomaly suppressions** list.

The **Suppression** dialog box includes the following parameters for editing:

- web application;
- action;
- anomaly analyzer;
- anomaly topics (specify tags for which suppression should be performed);
- anomaly location type (specify the more precise path for which suppression should be performed);
- location id (enable the analysis of the location id or the more precise path);
- offset and length option (select where the anomaly is located: value or name);
- explanation (a JSON structure to which you can add any important information. If ModSecurity is triggered, the triggered anomaly ID is specified).

The standard anomaly types are as follows:

PROTOCOL_VIOLATION, PROTOCOL_VIOLATION, XSS, PHP_INJECTION, SQL_INJECTION, FILE_INJECTION, COMMAND_INJECTION, DIR_TRAVERSAL, SESSION_FIXATION, RFI, ERRORS_IIS, ERRORS_PHP, ERRORS_SQL, ERRORS_JAVA, INFO_DIRECTORY_LISTING, SOURCE_CODE_JAVA, SOURCE_CODE_PHP, ENCODING_NOT_ALLOWED, PROTOCOL_NOT_ALLOWED, INVALID_HREQ, HEADER_RESTRICTED, MISSING_HEADER_HOST, METHOD_NOT_ALLOWED, REQUEST_SMUGGLING, EXT_RESTRICTED, SIZE_LIMIT, IP_HOST, CRAWLER, SCRIPTING, EVASION, SECURITY_SCANNER.

Available location types are given in the table below.

Filed/Check box	Value
Any	Do not check the anomaly path
Message	Anomaly can be in any part of the message
Start line	Anomaly can be in the start line of the message
Header	Anomaly can be in the header
Raw body	Anomaly can be in the whole body before decompressing and unpacking
Body	Anomaly can be in the whole body after decompressing and unpacking
URL	Anomaly can be somewhere in URL
Query	Anomaly can be somewhere in the URL request, but not in the specific parameter
Cookie	Anomaly can be in a cookie file with the name and specified ID if any
Parse tree	Path to a specific target parameter with the anomaly
Session data	Session data for suppression (not usually used)
Source	Specific source for suppression (not usually used)
Target/Web application object	Specific target for suppression (not usually used)

The created suppression appears in the Anomaly suppressions list.

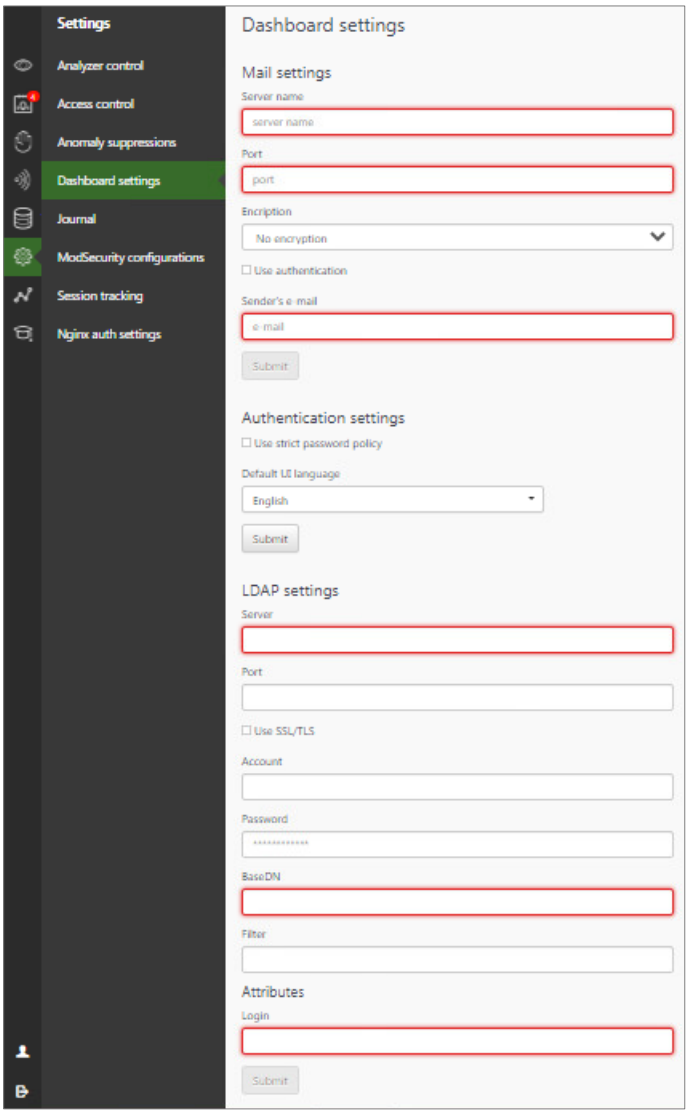
Dashboard settings tab

The **Dashboard settings** tab includes the following settings types:

- Mail settings which include settings to configure a SMTP server used to send notifications.
- Authentication settings which include password policy settings and default interface language settings.

- LDAP settings which include the domain name or IP address of the LDAP server, the port of the LDAP server, user name and password, and the Base DN parameters. If the **Use SSL/TLS** check box is selected, additional parameters become available for editing.

The general view of the tab is given in the figure below.



The parameters available for editing in the tab are listed in the table below.

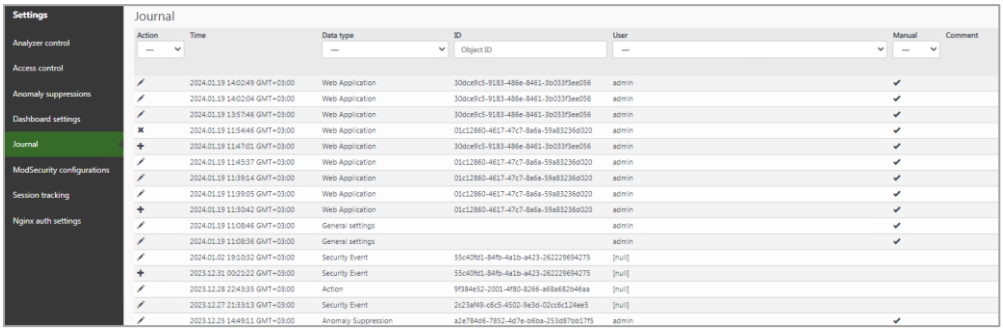
Filed/Check box	Value
Mail settings	
Server name	Domain name or IP address of the SMTP server
Port	SMPT server port number
Encryption	No encryption, SSL/TLS, STARTTLS
Use authentication	If the check box is selected, additional parameters become available for editing
Sender's e-mail	Email address from which notifications are sent
Account	User name
Password	User password
Sender`s email	Email address from which mails are sent
Authentication settings	
Use strict password policy	Adds complexity requirements to passwords
Default UI language	Allows selecting the interface language between English and Russian
LDAP settings	
Server	Domain name or IP address of a LDAP server

Filed/Check box	Value
Port	LDAP server port
Use SSL/TLS	If the check box is selected, additional parameters become available for editing
Check certificate	
Account	User name
Password	User password
BaseDN	Catalog object starting with which the search is performed

To save the settings, click **Submit**.

Journal tab

The tab is a log of user and system actions performed in the Configuration console. The general view of the tab is given in the figure below.



For a convenient search of events, use filters. The filter purposes are listed in the table below. Each column can be sorted using the respective filter.

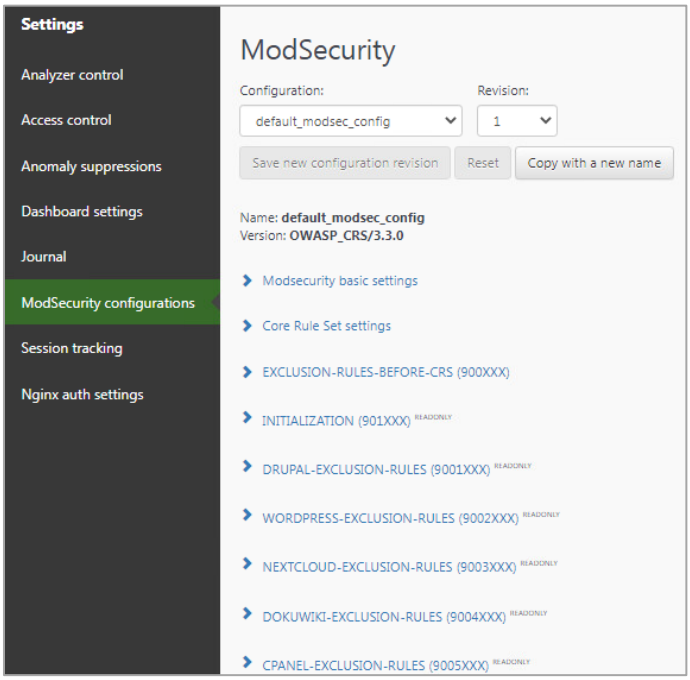
Column	Drop-down list options	Purpose
Action	Add/Edit/Delete	Filters by the action type
Time		Sorts by time (from old to new and backwards)
Data type	Action Predicate Analyzer Anomaly Suppression General settings Decision Rule Contents Decision Rule Information Default session descriptor Login Logout Data masking ModSecurity configuration Notification Parsing Algorithm Static url patter set Action rate limiting settings Action rate limiting settings meta information Request Validation Policy Response Validation Policy Security Event Action chain Session Management Policy Analyzer Settings Tenant WAF User User Management Policy Web Application Actions	Filters by the object type
ID		Object ID search field

Column	Drop-down list options	Purpose
User	Registered user list	Filters by the user who performed the action. For actions that are performed automatically by the system, the field has the [null] value
Manual	Manual/Auto	Allows selecting action performed manually or actions performed automatically by the system
Comment		Groups by the comments

To open the **Difference view** dialog box displaying the changes made, double-click the required table row.

ModSecurity tab

You can specify the configuration to be used by the ModSecurityAnalyzer module in the tab given in the figure below. To do that, select the name of the required configuration in the **Configuration** drop-down list.



On this tab, there are signatures for the ModSecurity signature analyzer grouped into the corresponding groups.

View triggered rules

To view which rule was triggered when a transaction was blocked, open the transaction detail. Then go to **Anomaly** tab, find the respective ModsecurityAnalyzer anomaly and click **...**

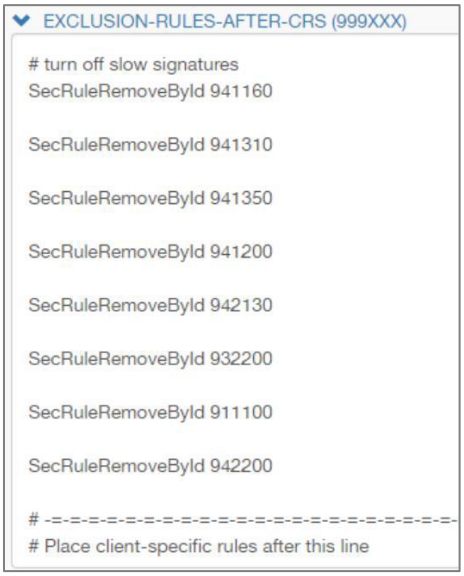
In the additional information, you can see the **id** field. This **id** points to the number of the Modsecurity signature triggered.

To find a rule by ID, go to **ModSecurity configurations** in the **Settings** menu. After a signature name, you can see the numbers of corresponding signatures in parentheses, where **X** is any number.

In this list, find a group in which the signature is; expand it by clicking its name. Then, find the signature by its ID (you can use the key combination <**Ctrl**>+<**F**>). The signature from the example is given in the figure below.

Enable signature for installation

In the **ModSecurity configurations** tab of the **Settings** menu, open the **EXCLUSION-RULES-AFTER-CRS (999XXX)** group. Scroll it to the bottom and add the **SecRuleRemoveById ID** line, where **ID** is the number of the signature to be disabled. Click **Save new synchronization revision**, to save new changes.



Add signature for installation

In the **ModSecurity configurations** tab in the **Settings**, open the **EXCLUSION-RULES-AFTER-CRS (999XXX)** group. Scroll it to the bottom and add the the required line and signature according to the ModSecurity syntax. Click **Save new synchronization revision**, to save new changes.

Session tracking tab

The tab contains a predefined set of key session attributes most commonly used to transfer session information.

The attribute data includes the following information:

1. Cookies with common names (for example, wordpress_logged_in_, JSSESSIONID, ASP.NET_SessionID, etc.);
2. HTTP headers with common names (Authorization, User Agent);
3. Fixed transaction source properties (src_ip).

Each attribute from the predefined set has the following properties:

1. Request variable;
2. Response variable;
3. Flag indicating whether this attribute requires pre-installation by the web application.

Attributes are grouped by priority, ranging from more specific attributes to the most common attributes.

Nginx auth settings tab

You can add a user of the protected application, as well as change a user password, clear a user session or remove a user in the tab.

Chapter 6
Launch Continent WAF services

All Continent WAF services are managed by **systemd** and configured using the **/lib/system/system/<service-name>** files. Typically, services use configuration variables from the **/etc/default/<service-name>** file. The services are configured using configuration files, except for the **scwaf-analyzer** service, which receives an additional configuration from MongoDB.

A detailed description of the services is given in the table below.

Service	Description
scwaf-nginx	It interacts with the **scwaf-analyzer** service via the **scwaf-redis** service in reverse-proxy mode. The interaction is performed via specially developed lua modules
scwaf-redis	It is used to store the rapidly changing local information for the analyzer (user session data) and for communication between **nginx** and **scwaf-analyzer**. Used in PUB-USB mode
scwaf-analyzer	The main service is responsible for anomaly detection in responses and requests and issuing verdicts. It consists of a collection of modules that communicate via ZeroMQ using a process broker. The **scwaf-analyzer** service is configured via the web interface. The configuration is saved to the mongodb database. To read this configuration, the analyzer needs to know the credentials to access the database, as well as the analyzer name (the part that starts with **a- analyzer-name** string). When starting the analyzer, the initial configuration is taken from the **/etc/default/scwaf-analyzer** file (this file contains the name of the analyzer and the address and credentials of the MongoDB database where its configuration is stored). Then the analyzer configuration is read from **mongod**. By default, the scwaf-analyzer debug logs are located in the **/var/log/waf/wafd -{debug/info/error}.log** file. The stdin and stdout output streams can be viewed by running the **journalctl -xe -u scwaf-analyzer** command
scwaf-dashboard	The service is responsible for the web interface. The debug log is located in the **/var/log/waf/wafui.log** file. The contents of the stdout and stderr output streams can be viewed by running the **journalctl -xe -u scwaf-dashboard** command. The service settings are located in the **/etc/scwaf/dashboard/config.yml** and **/etc/default/scwaf-dashboard** files. The **/etc/scwaf/dashboard/config.yml** file contains the initial password for the **admin** user, which is required for the first logon to the web interface. The passwords of the other users, as well as the password of the **admin** user after the first password change, are stored in the database. The **admin** user password stored in the **/etc/scwaf/dashboard/config.yml** file loses its relevance after the first logon
scwaf-suricata	Passive traffic capture module based on the modified Suricata software. The service settings are located in the **/etc/suricata/suricata/suricata-waf.yml** file. The debug log is located in the **/var/log/suricata** directory. The common reason for the failure to start this service is a disconnected network interface (the interface is DOWN)
scwaf-celery	Task execution system. It performs aggregation tasks (creating security events) and sending mail
scwaf-celerybeat	Task scheduler. Responsible for scheduling periodic tasks
postgreSQL	Relational storage for HTTP request and response logs and corresponding meta information
nginx	Web server of the Continent WAF management panel
mongoDB	Document-oriented repository for protected application profiles, security policies and analyzer module settings and settings for modules responsible for analyzing

When using scripts to install Continent WAF, automatic startup of services at boot is enabled by default.

To force restart Continent WAF:

1. Stop services in the following order by running the **stop** command:

```
systemctl stop scwaf-analyzer

systemctl stop scwaf-dashboard

systemctl stop scwaf-celery

systemctl stop scwaf-celerybeat

systemctl stop scwaf-suricata (only for passive mode)

systemctl stop scwaf-nginx (only for active mode)
```

Typically, the restart of DBMS services (postgresql, mongod) and the nginx service is not required.

2. Check that services are stopped correctly. To do that, run the following command:

```
systemctl status <service-name>
```

3. Start services in the following order by running the **start** command:

```
systemctl start scwaf-suricata (only for passive mode)

systemctl start scwaf-nginx (only for active mode)

systemctl start scwaf-celerybeat

systemctl start scwaf-celery

systemctl start scwaf-dashboard

systemctl start scwaf-analyzer
```

4. Wait about one minute. Check that services are started correctly. To do that, run the following command:

```
systemctl status
```

Service configuration

You must check the service settings described below after the installation and launch of Continent WAF services.

Disk space

You must keep track of free disk space. Pay attention to the **/var/log** and **/var/lib** partitions in particular.

Service performance

The following services are required to be started (**UP** status) depending on the operating mode:

- software developed by us:
  - scwaf-analyzer — for all deployment options on an analyzer node. Additionally, check that all main modules operate properly by running the **ps axu | grep modules** command;
  - scwaf-dashboard, scwaf-celery, scwaf-celerybeat — for all installation options on a storage node or interface.
- Third-party software:
  - mongod, postgresql — for all deployment options on a storage node;
  - scwaf-redis — for all deployment options on an analyzer node and data storage node;
  - scwaf-nginx — on active nodes with the role of an analyzer;
  - scwaf-suricata — on passive nodes with the role of an analyzer.

Key elements of scwaf-analyzer logs

- The number of incoming requests in passive mode:

```
$ tail -f /var/log/waf/wafd-debug.log | grep Passive

2015-11-03 13:30:37.361 INFO 32656:MainProcess
[modules.passive_adapter.PassiveHttpZmqAdapter.run]
Requests count (all/related): 835/680

2015-11-03 13:30:37.361 INFO 32656:MainProcess
[modules.passive_adapter.PassiveHttpZmqAdapter.run]
reqs/sec (all/related)

83.189024 / 67.746750
```

- Message rate for each of the components:

```
$ tail -F wafd-debug.log | grep 'Message rate'

2015-11-03 17:15:58.029 INFO 2748:MainProcess
[modules.decision_tree_request_parser.DecisionTreeRequ
estParser.message_generator] Message rate:
174.946590/sec

2015-11-03 17:15:59.108 INFO 2750:MainProcess
[modules.libinjection_detector.LibinjectionDetector.me
ssage_generator] Message rate: 91.034136/sec

2015-11-03 17:16:00.245 INFO 2746:MainProcess
[modules.decision_maker.DecisionMakerModule.message_ge
nerator] Message rate: 779.696171/sec 2015-11-03
17:16:02.048 INFO 3329:MainProcess
[modules.action_determiner.ActionDeterminer.message_ge
nerator] Message rate: 190.329310/sec

2015-11-03 17:16:06.229 INFO 2749:MainProcess
[modules.modsecurity.ModsecurityAnalyzer.message_gener
ator] Message rate: 126.318664/sec
```

- The **decision_maker** error (it must be no more than 0.01 seconds and increases on loaded installations):

```
$ tail -F wafd-debug.log | grep 'DecisionMaker' | grep
lag

2015-11-03 17:17:04.240 INFO 2746:MainProcess
[modules.decision_maker.DecisionMakerModule.message_ge
nerator] Max message lag: 2.254621
```

- Data chunk sizes when writing data to postgresql (note chunk sizes and the writing speed):

```
$ tail -F wafd-debug.log | grep
'common.db.http_transaction'

2015-11-03 17:22:49.965 DEBUG 2747:MainProcess
[common.db.http_transaction.BatchPgTxManager._flush_si
ngle_update] UPDATE 7 values, cols: resp_decision

2015-11-03 17:22:50.880 DEBUG 2747:MainProcess
[common.db.http_transaction.BatchPgTxManager.flush]
Table http_transaction: 6 inserts, 1 updates
```

```
2015-11-03 17:22:50.880 DEBUG 2747:MainProcess
[common.db.http_transaction.BatchPgTxManager._flush_in
serts] INSERT 6 values, cols: body, protocol, raw_uri,
obj_id, uri, method, src_ip, headers, dst_port, time,
src_port, dst_ip, webapp_id, _response_src_port,
_response_time, _response_dst_port, _response_body,
_response_headers, _response_dst_ip, _response_src_ip,
_response_protocol, _response_obj_id,
_response_status, req_tree, req_decision

2015-11-03 17:22:50.881 DEBUG 2747:MainProcess
[common.db.http_transaction.BatchPgTxManager._flush_in
serts] INSERT 18 values, cols: body, protocol,
raw_uri, obj_id, uri, method, src_ip, headers,
dst_port, time, src_port, dst_ip, webapp_id,
_response_src_port, _response_time,
_response_dst_port, _response_body, _response_headers,
_response_dst_ip, _response_src_ip,
_response_protocol, _response_obj_id,
_response_status, req_tree

2015-11-03 17:22:50.881 DEBUG 2747:MainProcess
[common.db.http_transaction.BatchPgTxManager._flush_in
serts] INSERT 23 values, cols: body, protocol,
raw_uri, obj_id, uri, method, src_ip, headers,
dst_port, time, src_port, dst_ip, webapp_id, req_tree

2015-11-03 17:22:50.881 DEBUG 2747:MainProcess
[common.db.http_transaction.BatchPgTxManager._flush_in
serts] INSERT 1 values, cols: body, protocol, raw_uri,
obj_id, uri, method, src_ip, headers, dst_port, time,
src_port, dst_ip, webapp_id, _response_src_port,
_response_time, _response_dst_port, _response_body,
_response_headers, _response_dst_ip, _response_src_ip,
_response_protocol, _response_obj_id,
_response_status, req_tree, resp_decision,
req_decision

2015-11-03 17:22:50.881 DEBUG 2747:MainProcess
[common.db.http_transaction.BatchPgTxManager._flush_in
serts] INSERT 7 values, cols: body, protocol, raw_uri,
obj_id, uri, method, src_ip, headers, dst_port, time,
src_port, dst_ip, webapp_id, _response_src_port,
_response_time, _response_dst_port, _response_body,
_response_headers, _response_dst_ip, _response_src_ip,
_response_protocol, _response_obj_id,
_response_status, req_tree, req_decision,
resp_decision

2015-11-03 17:22:50.881 DEBUG 2747:MainProcess
[common.db.http_transaction.BatchPgTxManager._flush_in
serts] INSERT 48 values, cols: body, protocol,
raw_uri, obj_id, uri, method, src_ip, headers,
dst_port, time, src_port, dst_ip, webapp_id, req_tree,
req_decision
```

- The speed of writing to postgresql (the desired time of one writing cycle is up to 1 second):

```
$ tail -F wafd-debug.log | grep batch
```

```
2015-11-03 17:26:47.861 DEBUG 4968:MainProcess

[common.db.batch.BatchPgWriter.writer] BatchPgWriter
loop start

2015-11-03 17:26:48.021 DEBUG 4968:MainProcess

[common.db.batch.BatchPgWriter.writer] BatchPgWriter
loop finished in

0.159891

2015-11-03 17:26:47.861 DEBUG 4968:MainProcess

[common.db.batch.BatchPgWriter.writer] BatchPgWriter
loop start

2015-11-03 17:26:49.289 DEBUG 4968:MainProcess

[common.db.batch.BatchPgWriter.writer] BatchPgWriter
loop finished in

0.426322
```

Additional configuration after starting the services

DBMS database migrations (PostgreSQL, MongoDB)

You must perform a DBMS database migration (postgresql, mongod) only when updating the system. To do that, run the following commands:

- for the MongoDB DBMS:

```
cd /home/waf/waf

.env/bin/activate

python migrations/mongo/__main__.py --port 27017 -d
waf --host 127.0.0.1 -p password -u waf
```

- for the PostgreSQL DBMS:

```
cd /home/waf/waf

migrations/postgres/migrate.sh
```

Control access to the web interface

To control access to the web interface, you need to assign an IP address from which access is allowed. To do that, add an IP address to the **listen** policy of the **/etc/nginx/sites-enabled/wafui_proxy** configuration file.

Protection from bots

To ensure protection from bots, we recommend using the following methods:

- add the user-agent bots to the **/home/waf/waf/config/modsec/crawlers-user-agents.data** file in the ModSecurity settings. The added bots are blocked automatically. The most common bots are already listed in this file. The respective signature blocks them;
- if the subnets from which bots are sending requests are known, a rule that blocks requests from those subnets can be created;
- if bots send many requests for the same action, you can configure the brute-force detector;

- analyze how bot requests differ from the requests of average users, and filter them using a business logic model. For example, bots may not have browser-specific headers, no cookies, etc.

Note. Using these methods, you can accidentally block search engine bots or automatic systems that check the availability of resources.

SyslogExporter module configuration

The SyslogExporter module is used to create log files for export via the Syslog mechanism. The SyslogExporter module runs as a task for Celery software. Celery is an asynchronous task queue. The SyslogExporter configuration is specified in the **/etc/scwaf/celery/config.yml** file. An example of the configuration is given below:

```
.......
syslog_exporter:
task: 'common.db.syslog_exporter.tasks.SyslogExporter'
schedule:
every: 1
period: 'minutes'
kwargs:
mode: 'BLOCKED'
syslog_address: 'tcp://localhost:6789'

template: '{time} {host_ip} CEF:0|SCWAF|Continent
WAF|2.0|{req.obj_id}|HTTP
Transaction|{severity}|src={req.src_ip}
scrPort={req.src_port} dst={req.dst_ip}
dstPort={req.dst_port}
request_time={req.time} response_time={resp.time}
request={req.raw_uri}
requestCookies={req_tree.headers.cookie}
requestMethod={req.method}
file_path={req_tree.url.path}
in={req_tree.headers.content-length}
out={resp_tree.headers.content-length}'
...........
```

The following key parameters are used in the configuration:

- **schedule** — section that specifies the schedule of task execution (the format of section parameters is the same for all tasks managed by Celery);
- **every** — specifies the frequency with which the task is executed in the specified units;
- **period** — specifies the units in which the frequency is set (seconds, minutes, hours, etc.);
- **kwargs** — section that specifies parameters unique to the task (these parameters are passed to the respective process. In this case, SyslogExporter);
- **mode** — specifies the event logging mode. There are two possible values for the parameter:
  - BLOCKED — only blocked transactions are written to the log;
  - ALL — all transactions are written to the log;
- **syslog_address** — specifies the type and address of the syslog service connection. Examples of data records:
  - **tcp://localhost:6789** — network connection, TCP protocol;
  - **udp://localhost:6789** — network connection, UDP protocol;
  - **file:///dev/log** — file;
- **template** — specifies a logging format that is customizable for export purposes and integration with other systems. The data is output in the format described

by the specified template in the template positions, which in turn are defined by placeholder parameters. The following substitution parameter values are supported:

- **time** — transaction time;
- **host_ip** — IP address of the host that processed the transaction;
- **severity** — transaction severity;
- **req** — for the HTTP response:
  - **obj_id** — unique object ID;
  - **src_ip** — source IP address;
  - **src_port** — request source port;
  - **dst_ip** — destination IP address;
  - **dst_port** — destination port;
  - **time** — request receipt time;
  - **raw_uri** — raw URI;
  - **method** — HTTP method;
  - **protocol** — HTTP protocol;
- **resp** — for the HTTP response:
  - **time** — time the response is sent;
- **req_tree** — for the request tree:
  - **time** — time;
  - **src_port** — source port;
  - **dst_ip** — destination IP address;
  - **dst_port** — destination port;
  - **protocol** — HTTP protocol;
  - **method** — HTTP method;
  - **url** — URL;
  - **headers** — HTTP request headers:
    - **cookie** (header name);
    - …;
- **resp_tree** — for the decision tree:
  - **time** — response time;
  - **src_port** — source port;
  - **dst_ip** — destination IP address;
  - **dst_port** — destination port;
  - **protocol** — HTTP protocol;
  - **method** — HTTP method;
  - **url** — URL;
  - **headers** — HTTP response headers:
    - **cookie** (header name);
    - …;
- **req_decision** — for the WAF decision concerning the specific request:
  - **decision** — decision;
  - **rule_name** — rule triggered;
  - **rule_id** — rule ID;
- **resp_decision** — for WAF decision concerning the specific response:
  - **decision** — decision;
  - **rule_name** — rule triggered;
  - **rule_id** — rule ID;
- **webapp_id** — ID of the protected web application;

- **webapp_name** — name of the protected application;
- **action_id** — action ID in the business logic terms;
- **action_name** — action name in business logic terms;
- **action_parameters** — action parameters;
- **request_session_id** — request session ID;
- **response_session_id** — response session ID.

Examples of the template customization for the most common log file formats processed by SIEM systems are given below.

1. LEEF

```
'{time} {host_ip} LEEF:2.0|SCWAF|Continent
WAF|2.0|{req.obj_id}|src={req.src_ip}\tspt={req.src_po
rt}\tdst={req.dst_ip}\tdpt={req.dst_port}\trequest_tim
e={req.time}\tresponse_time={resp.time}\turl={req.raw_
uri}\tsev={severity}\treq_tree.headers.cookie={req_tre
e.headers.cookie}\trequestMethod={req.method}\treq_tre
e.url.path={req_tree.url.path}\tdstBytes={req_tree.hea
ders.content-
length}\tsrcBytes={resp_tree.headers.content-length}'
```

2. CEEF

```
'{time} {host_ip} CEF:0|SCWAF|Continent
WAF|2.0|{req.obj_id}|HTTP
Transaction|{severity}|src={req.src_ip}
scrPort={req.src_port} dst={req.dst_ip}
dstPort={req.dst_port} request_time={req.time}
response_time={resp.time} request={req.raw_uri}
requestCookies={req_tree.headers.cookie}
requestMethod={req.method}
file_path={req_tree.url.path}
in={req_tree.headers.content-length}
out={resp_tree.headers.content-length}'
```

Restart the Celery processes after changing the configuration.

In the default configuration (when the celery service is not replicated as the StandAlone), restart the service by running the following commands:

```
sudo systemctl restart scwaf-celery

sudo systemctl restart scwaf-celerybeat
```

The system starts generating the log in the required format.

In a cluster configuration, change the configuration file on both nodes in the cluster. To apply the changes, run the following commands:

```
sudo crm resource restart celery

sudo systemctl restart scwaf-celery
```

You can check the transfer of logs by using the tcpdump tool with the specified host and port.

Further integration with other systems (including SIEM) must be performed using the Syslog mechanism in accordance with the documentation for these systems.

Configure Open redirect

With Continent WAF, you can eliminate an open redirect vulnerability, which allows redirecting a user from a trusted domain to any website. This vulnerability can be used for:

- Credentials theft by replacing an authorization webpage with a phishing website;

- Reducing the search engine ranking for a webpage from which open redirection is taking place.

To turn on this module, enable the **OpenRedirectDetector** and **DecisionTreeResponseParser** modules (see p.**50**) and enable the **Open redirect** rule.

This module uses response analysis. The detector analyzes the values of the **location** and **refresh** headers in a response and blocks requests in which these values are different from the value of the **host** header in the request.

Configure CSRF detector

With Continent WAF, you can eliminate a CSRF vulnerability, which allows the attacker to use a website under the name of a registered user.

To turn on this module, enable the CsrfDetector module and enable the **CSRF-attack** rule.

This module blocks **POST** requests in which the domain from the referred header does not match the domain from the **host** header.

Chapter 7
Configuring multi-tenant model

The multi-tenant model makes it possible to distribute installation applications across separate and isolated areas (tenants). Sources, lists, targets, rules, and session attributes made by a user of one tenant are not visible to users of another tenant. The exceptions are the attributes listed above, which are in Continent WAF by default. They are visible to users from all tenants and sources, lists, targets, rules, and session attributes created by the super administrator without specifying the tenant.

The capability to view all installation applications regardless of the tenant they belong to is available to the super administrator.

To enable the multi-tenant model on the installation on the control node (the wizard in the case of a failover configuration), run the following commands:

```
sudo -u waf psql

update "user" set role = 'super_admin' where id = 1;
sudo systemctl restart scwaf-dashboard
```

For the failover configuration run the following command:

```
sudo crm resource restart scwaf-dashboard
```

Chapter 8
Configuring modules

The description of module setup general parameters is shown in the table below.

Parameter	Description
group_name	The parameter for selecting a message exchange queue with ZeroMQ. The default value is local. **1.** BrokerQueue (broker) — working only with the ZeroMQ broker, no message exchange within the process. **2.** LocalQueue (local) — a local queue that transfers data only within a single process. **3.** CombinedQueue (combined) — receiving and sending messages both using ZeroMQ and within a process
max_cache_size	The size of the shared cache between the transaction recording module and the action recording module
num_instances	The number of the processes used for the module. The default value is 4. The number of the processes (pipelines) must be: **1.** The number of CPU cores minus 2 — on a node with a dedicated analyzer. **2.** The number of CPU cores multiplied by 0.5 — for standalone installation
python_ implementation	The type of Python interpreter for the analyzer. The interpreter by default is Cpython, you should not change this value
use_redis_mgmt	Using a shared Redis between analyzers for centralized parameter storage. You should not use this setting together with specifying a separate centralized Redis address within a single module. **1.** false — do not use the shared address specified in the **redis_mgmt** tab of the analyzer settings for centralized Redis (by default). **2.** true — use the shared address specified in the **redis_mgmt** tab of the analyzer settings for the centralized Redis
redis -> url	The address of the centralized Redis for bucket storage of the brute force attack detector, if there is any. You should not use this setting together with use_redis_mgmt, as the specific centralized Redis address is set there, instead of using the shared Redis specified in the **redis_mgmt** tab of the analyzer settings. If there is no centralized Redis, a local one is used on each node

BruteforceDetector

This module is responsible for the operation of the brute-force attack detector.

The list of settings that users are allowed to change is shown in the table below.

Parameter	Description
bucket_size	The bucket size for the brute force attack detector value by default
leak_rate	The number of tokens leaking from the bucket per second by default
relieve_on_time out	Enabling the mode of the timeout lock mode when the bucket is full, in seconds. **1.** false — disabled (by default); **2.** true — enabled
relieve_timeout	The request blocking time in the timeout blocking mode, in seconds
timeout_reset	The Timeout resetting with each new request. **1.** false — disabled (by default); **2.** true — enabled
tokens_per_ action	The number of tokens added to the bucket per request
tokens_per_ failed_action	The number of tokens added to the bucket per failed action
tokens_per_ response_404	The number of tokens added to the bucket per failed action
unrecognized_ action	The detector parameters for unrecognized actions only, the description is the same as listed above

DecisionMakerModule

This module is responsible for making the decision about passing or blocking a transaction based on the analysis result of other analyzer modules.

The list of settings that users are allowed to change is shown in the table below.

Parameter	Description
bucket_size	The log describing the moment when the decision is made by Decision Maker. Adds information to the anomalies table. The information from there is displayed at the bottom of the **Solution** tab when you view the transaction. It is only needed for debugging. **1.** false — disabled (by default); **2.** true — enabled
processing_time out	The time for collecting anomalies before the final decision is made by Decision Maker. You can increase the value of this parameter if there are any heavy requests on installation applications. It is considered a good practice to change it along with the decision_timeout value of the NginxZmqAdapter module settings according to the following formula: processing_timeout + 0.1 = decision_timeout, which allows you to allocate time for sending data by the NginxZmqAdapter module after making the decision by the DecisionMaker and avoid timeout errors

DecisionTreeResponseParser

This module is responsible for parsing the response tree to the request. It is disabled on most installations, as it requires a lot of resources.

The list of settings that users are allowed to change is shown in the table below.

Parameter	Description
parse_unrecognized_action	The parameter responsible for parsing the response in case of an unrecognized Continent WAF action. **1.** false — disabled (by default); **2.** true — enabled
use_webapp_control	The setting to control the response parsing parameter for each application separately. **1.** false — disabled (by default); **2.** true — enabled. In this case, the configuration is performed in the **Settings** tab of each application on the installation. The parameter responsible for this is called **Build a response parsing tree**

DumperBatchModule

This module is responsible for recording transactions in the database (Postgres, MongoDB).

The list of settings that users are allowed to change is shown in the table below.

Parameter	Description
mask	The parameter that enables data masking. **1.** false — disabled (by default); **2.** true — enabled
max_batch_size	The size of the batch of transactions that will be recorded to the database
max_watched_action_num	Buffer size for actions
max_watched_tx_num	Buffer size for transactions
normal_tx_sample_rate	The parameter indicating the sampling rate (partial record to the database) for passed transactions to save disk storage. The parameter indicating the sampling rate (partial record to the database) for passed transactions to save disk storage
store_all_actions	The parameter that enables saving all actions in a separate table in Postgres **1.** false — disabled (by default); **2.** true — enabled

Parameter	Description
store_normal_ anomalies	Whether to save suppressed anomalies (false firings) to the database. **1.** false — disabled (by default); **2.** true — enabled
store_normal_txns	Whether to save normal transactions without triggers to Continent WAF. **1.** false — disabled (by default); **2.** true — enabled
store_unknown	Whether to store transactions with actions unknown to Continent WAF. **1.** false — disabled (by default); **2.** true — enabled
tx_data_wait_time out	The waiting time for transaction data, after which there is a forced entry into the database, in seconds
wait_for_all_data	The waiting time for all data before recording to reduce the number of change requests for to the database, since UPDATE is considered a resource-intensive query. **1.** false — disabled (by default); **2.** true — enabled

IcapClient

This module is used to send data to external systems via ICAP.

The list of settings that users are allowed to change is shown in the table below.

Parameter	Description
icap_server_ address	ICAP server IP address
port	ICAP server port
service	The address of the AV server to which data is sent for verification
socket_timeout	The response waiting time, in seconds

LWSessionTracker

This module is responsible for the operation of lightweight tracking of user sessions based on the specified values of session attributes.

The list of settings that users are allowed to change is shown in the table below.

Parameter	Description
max_users_allowed	The maximum number of times the session identifier can be changed. As a rule, the user's name
session_lifetime	The session lifetime between requests, in seconds

ModsecurityAnalyzer

This module is responsible for working with an optimized signature analyzer based on Modsecurity.

The list of settings that users are allowed to change is shown in the table below.

Parameter	Description
config_file	The path to the configuration for Modsecurity
config_name	The configuration name for Modsecurity

NginxDecisionDumper

This Nginx module is responsible for recording Decision Maker data about decisions into the Postgres database.

The list of settings that users are allowed to change is shown in the table below.

Parameter	Description
dumper -> max_	The cache size for transactions

Parameter	Description
watched_tx_num	
store_normal_ anomalies	Whether to save suppressed anomalies (false alarms) to the database. **1.** false — disabled (by default); **2.** true — enabled
wait_for_all_data	Waiting for all data before recording to reduce the number of change requests for to the database, since UPDATE is considered a resource-intensive request. **1.** false — disabled (by default); **2.** true — enabled

NginxZmqAdapter

This Nginx adapter is for fast asynchronous data delivery to the analyzer using the optimized ZeroMQ messaging library.

The list of settings that users are allowed to change is shown in the table below.

Parameter	Description
decision_timeout	The waiting time from the decision analyzer modules, after which the default decision is made. You can increase the value of this parameter if there are any heavy requests on installation applications. It is considered a good practice to change it together with the **processing_timeout** value of the DecisionMakerModule settings according to the following formula: processing_timeout + 0.1 = decision_timeout, which allows you to allocate time for sending data by the NginxZmqAdapter module after the decision is made by Decision Maker and avoid timeout errors
default_decision _for_unknown_ webapp	The default decision for transactions from unknown applications. If the application traffic got to Nginx, but was not determined to any application in Continent WAF through tuples, it is considered an unknown application. **1.** PASS (by default, Solidwall-nginx for all versions); **2.** BLOCK

SequenceAnomalyDetector

This module is responsible for the operation of user action chains.

The list of settings that users are allowed to change is shown in the table below.

Parameter	Description
anomaly_threshold	The anomaly bucket threshold; when it is crossed, an anomaly is generated
deep_pattern_ violation_score	The number of tokens added to the bucket at deep pattern violation
max_trace_length	The maximum trace length
sequence_ttl	The time of relevance of actions within the chain
session_lifetime	User session lifetime
short_pattern_violation_score	The number of tokens added to the bucket at short pattern violation

SessionAnomalyCounter

This module is responsible for anomaly counting within a user session.

The list of settings that users are allowed to change is shown in the table below.

Parameter	Description
anomaly_threshold	The anomaly bucket threshold, when passed, an anomaly is generated

Note. You must not change the other analyzer settings that are described in this document without approval.

Chapter 9
Configuring integration with third party systems

Continent WAF provides integration with the following third-party systems:

- security events export to a third party SIEM system (syslog: server address and selection from the list of formats);
- integration with issue-tracking systems.

Technical support sets up interaction between Continent WAF and third-party systems (if such a clause is present in the Contract).

Chapter 10
Backup

Continent WAF comes in several modifications. The modifications are as follows:

- appliance;
- virtual device (virtual machine);
- software distribution kit.

Appliance backup and restoration

You can create backups and restore the appliance using the backup tools of the appliance or third-party backup systems by creating disk images. Restoration is performed according to the documentation for the selected backup system.

Virtual machine backup and restoration

You can create backups for the virtual machine (hereinafter — VM) using standard tools of the hypervisor or third-party tools according to the documentation by creating a snapshot of the configured VM (when it is turned off) and saving an image of the whole virtual machine. VM is restored fully according to the documentation.

Software backup and restoration

Create a backup copy

If you have Continent WAF as software, create backup copies of the following files and directories:

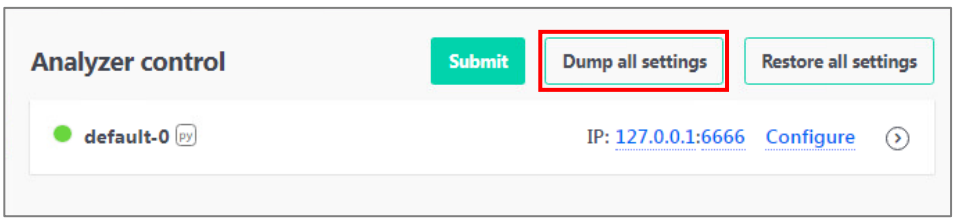
- **/etc/default/scwaf-dashboard**;
- **/etc/default/scwaf-analyzer**;
- **/etc/default/scwaf-celery**;
- **/etc/default/scwaf-celerybeat**;
- **/etc/default/scwaf-suricata**;
- **/etc/default/scwaf-nginx** (whole directory);
- **/home/waf/waf/config** (whole directory).

You also need to create a backup copy of the **waf** database in the PostgreSQL DBMS and a backup copy of the **waf** and **celery** databases in the MongoDB DBMS. Database backup copies are created via standard DBMS tools using the built-in **pg_dump** and **mongodump** tools.

Besides backup and restoration of all data, the system allows you to create and restore backups of analyzer settings.

To export Continent WAF analyzer settings:

1. Open the management console.
2. Go to the **Analyzer control** section in **Settings**.
3. Click **Dump all settings**.



Note. By default, data about previously detected events is not saved during copying and restoration. If you need to save this data, contact technical support.

Restore from a backup copy

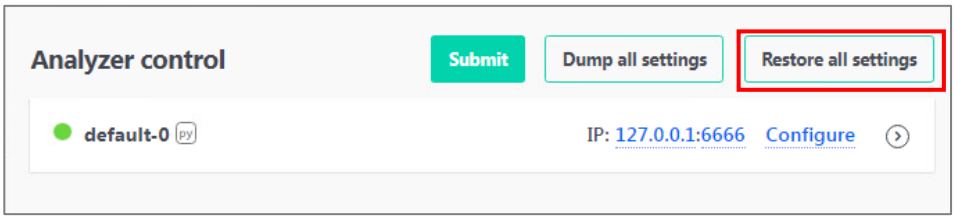
To restore the system from a backup copy:

1. If necessary, install the system software and the Continent WAF software according to the instructions in this manual.
2. Restore the contents of PostgreSQL and MongoDB DBMS via their standard tools using the built-in **pg_restore** and **mongorestore** tools.
3. Transfer the backup copies of the configuration files mentioned above.

Besides backup and restoration of all data, the system allows you to create and restore backups of analyzer settings.

To restore settings of all analyzers:

1. Open the management console.
2. Go to the **Analyzer control** section in **Settings**.
3. Click **Browse at the bottom**.
4. In the appeared dialog box, select the saved configuration file and click **Open**.
5. After the system loads the configuration file, click **Restore all settings**.



Chapter 11
Troubleshooting

Services do not start

scwaf-analyzer, scwaf-dashboard

If the service is missing from the list of processes after its start, you need to find the cause of the error in the **journald** log using the **journalctl -xe -u <service-name>** command.

scwaf-celery / scwaf-celerybeat

Check logs by running the following command:

```
/var/log/waf/scwaf-celery*.log
```

If you did not find the cause of failure of one of the working processes of the **scwaf-celery** service or the **scwaf-celerybeat** process scheduler, you need to start the respective service manually, because the cause of failed start of working processes or process scheduler may not display in the logs due to services starting in **celery multi** mode.

To force the service start:

1. In the executable **scwaf-celery** script **(/usr/sbin/scwaf-celery),** replace **set -e with set -ex** and try to start the service again.
2. During the service startup process, all running commands are displayed.
3. Find the error in **journald**.

Disk space is running out

/var/lib

You need to identify which process takes up space.

To do that, run the following command:

```
sudo du -hs /var/lib/* | sort -rh | head -n 10
```

If postgres takes up space, check which tables in postgres take up space using the following command:

```
sudo -u postgres psql waf

>> SELECT nspname || '.' || relname AS "relation",

pg_size_pretty(pg_total_relation_size(C.oid)) AS
"total_size"

FROM pg_class C

LEFT JOIN pg_namespace N ON (N.oid = C.relnamespace)

WHERE nspname NOT IN ('pg_catalog',
'information_schema')

AND C.relkind <> 'i'

AND nspname !~ '^pg_toast'

ORDER BY pg_total_relation_size(C.oid) DESC

LIMIT 20;
```

/var/log

You need to identify the process which contains large files. To do that, run the following command:

```
sudo du -hs /var/log/* | sort -rh | head -n 10
```

Check whether the **logrotate** rules for components of the identified process are configured correctly in **/etc/logrotate.d**.

You need to adjust the settings so that they correspond to the attached **logrotate.d.tgz** file.

If the disk space is not freed up after you manually clear the old logs, you need to restart the process.

Application is unavailable

If the application is unavailable via Continent WAF, we recommend checking whether the following aspects are correct:

1. Network addressing.

```
ip a
```

2. Name resolution.

```
host example.com
```

3. Routing.

```
ip ro
```

4. Whether the application is available directly.

```
curl -v http://example.com

curl -k -v https://example.com
```

5. The scwaf-nginx service state.

```
sudo systemctl status scwaf-nginx.service
```

6. List of open ports.

```
ss -lnt
```

7. Whether the scwaf-nginx service log contains errors.

```
sudo journalctl -xefu scwaf-nginx
```

8. Whether the scwaf-nginx configuration contains errors.

```
sudo scwaf-nginx test
```

Documentation

1. Continent WAF. Version 2. User Guide