
Whalebone admin guide

Release 3.2.1-12

Feb 29, 2024

DEPLOYMENT OPTIONS

1	Whalebone Peacemaker	2
1.1	Local DNS resolver for ISP	2
2	Whalebone Immunity	3
2.1	Local DNS forwarder	3
3	Cloud deployment	5
3.1	Use existing DNS to forward to Whalebone Cloud DNS	5
3.2	Cloud DNS (direct connection)	6
4	Quickstart	8
4.1	Creating the portal account	8
4.2	DNS traffic view	9
5	Local resolver	10
5.1	Local resolver system requirements	10
5.2	Installation of a new local resolver	12
5.2.1	Verifying the installation	12
5.2.2	Securing your resolver	14
6	Resolver management	15
6.1	Resolvers overview	15
6.2	Deploy configuration	15
6.3	Configure Policy per Network Segment	16
6.4	Configure Blocking Pages	17
6.5	Upgrade/Rollback Resolver	18
7	Security policies	20
7.1	Malicious filtering thresholds	20
7.2	Types of threats	21
7.3	Allow lists	22
7.4	Deny Lists	22
7.5	Regulatory Restrictions	22
7.6	Content Filtering	23
8	DNS resolution configuration	25
9	Knot Resolver - Tips & Tricks	27
9.1	Allow particular IP ranges	27
9.2	Refuse particular IP ranges	28
9.3	Allow list of domains	28

9.4	Deny list of domains	28
9.5	Disable DNSSEC globally	29
9.6	Disable DNSSEC validation for a domain	29
9.7	Disable Query Case Randomization	29
9.8	Disable QNAME Minimization	29
9.9	Disable Domain caching	29
9.10	Enable Prometheus Metrics	29
10	Blocking Pages	30
10.1	Signing blocking pages with a CA	32
11	Resolver agent	34
11.1	Command line interface	34
11.2	Strict mode	39
12	Cloud DNS resolvers	41
13	Uninstalling a local resolver	43
14	Data Analysis	44
14.1	Threats	44
14.1.1	How to search for audit/block events:	44
14.1.2	How to search for a domain:	44
14.1.3	How to search for events based on specific IP address:	45
14.1.4	How to search for events based on specific threat category:	45
14.1.5	How to change the date range of the available data:	45
14.1.6	How to analyze a domain:	45
14.2	DNS Traffic	45
14.2.1	How to view all queries of a specific type:	46
14.2.2	How to view all answers of a specific type:	46
14.2.3	How to search for a domain:	46
14.2.4	How to change the date range of the available data:	46
14.2.5	How to view DGA (Domain Generation Algorithm) indications:	46
14.2.6	Fulltext filtering	46
15	Domain resolution analysis	48
16	Reports	49
17	Alerts	50
17.1	DNS traffic - count of unique requests from IP	50
17.2	DNS traffic - increased percentage of queries	51
17.3	DNS traffic - possible homograph attack	51
17.4	DNS traffic - threshold for unique queries	51
17.5	Resolver - Cloud communication failure	51
17.6	Resolver - Insufficient hardware resources	51
17.7	Resolver - Resolution service failure	52
17.8	Threats - count during intervals	52
17.9	Threats - event detection	52
17.10	Threats - newly blocked domain	52
18	API Integration	53
19	Active Directory Integration	54
19.1	Installation prerequisites	54
19.2	Domain Controller Configuration	55

19.2.1	DC Firewall on Windows	55
19.2.2	DC Firewall Rules	58
19.2.3	Windows Service	58
19.2.4	WMI Remote Configuration	59
19.3	Event Log Forwarder	60
19.3.1	ELF Firewall Rules	60
19.3.2	Install Instructions	60
19.3.3	Configuration Instructions	61
19.3.4	Service Logs	61
20	SNMP Monitoring	62
20.1	High Level Network Diagram	62
20.2	SNMP OID	63
20.2.1	Zabbix Integration	64
20.3	How to import the Whalebone Template	64
20.4	How to add the resolver in Zabbix Monitoring	65
20.5	How to add the Whalebone widget on Zabbix dashboard	68
20.6	How to add triggers on the Zabbix	70
20.7	How to configure the trigger actions	73
20.8	How to view the pre-defined Whalebone dashboard	75
21	User/Organization Management	78
21.1	User Management	78
21.2	Organization Settings	79
21.2.1	Portal Access Policy	80
21.2.2	Password Policy	80
22	Home Office Security Overview	81
22.1	Supported OS	82
23	Step by step installation	83
24	Operation	88
24.1	Devices	88
24.2	States	88
24.3	Security	88
24.4	Service requirements	89
24.4.1	Windows	89
24.4.2	Android	89
24.5	Application Firewall Settings	90
24.6	Application Logs	90
24.7	Uninstalling the app	90
25	Deployment	91
25.1	On-premise resolver deployment	91
25.2	Cloud resolvers	91
26	Configuration	92
26.1	Basic configuration	92
26.2	Security policies	92
26.3	Blocking page configuration	93
26.4	Alerts	93
27	Analysis	95
27.1	Domain analysis	95

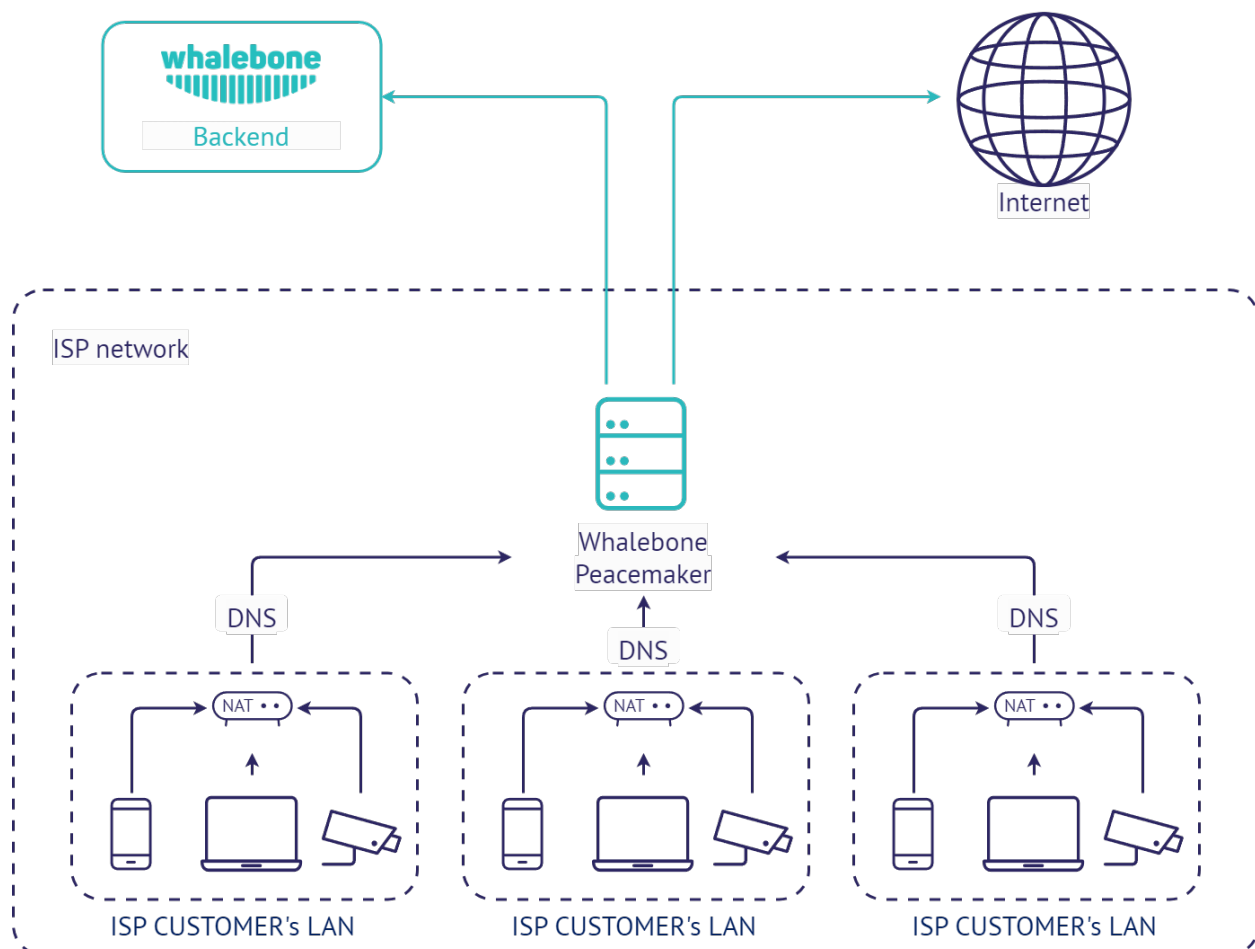
27.2	DNS traffic	95
27.3	Threats	96
27.4	Data Analysis	96
27.5	API	97
27.6	Domain resolution troubleshooting	97
27.7	Domain Tracing	98
28	License Disclaimers	99
28.1	the CRC64 variant with Jones coefficient	99
28.2	the Lightning.NET Library	100

Whalebone is a service for security filtering of DNS traffic. It uses logic on top of own DNS resolvers. Such resolvers could be either cloud ones maintained directly by Whalebone, or on-premise software resolvers using cloud just for threat intelligence updates and reporting. For threat prevention Whalebone relies on external intelligence sources as well as on own methods. More information about the product and company is available on the official [Whalebone website](#).

WHALEBONE PEACEMAKER

1.1 Local DNS resolver for ISP

This deployment scenario uses local Whalebone resolver, that communicates with Whalebone cloud through API. The DNS resolution takes place directly on the resolver and is completely independent on the cloud availability. In the case of resolver not being able to reach the cloud service, it won't be able to update the threat intelligence and to reports any incidents. The main advantage of this deployment is visibility into local network and individual IP addresses and native DNS resolver latency.

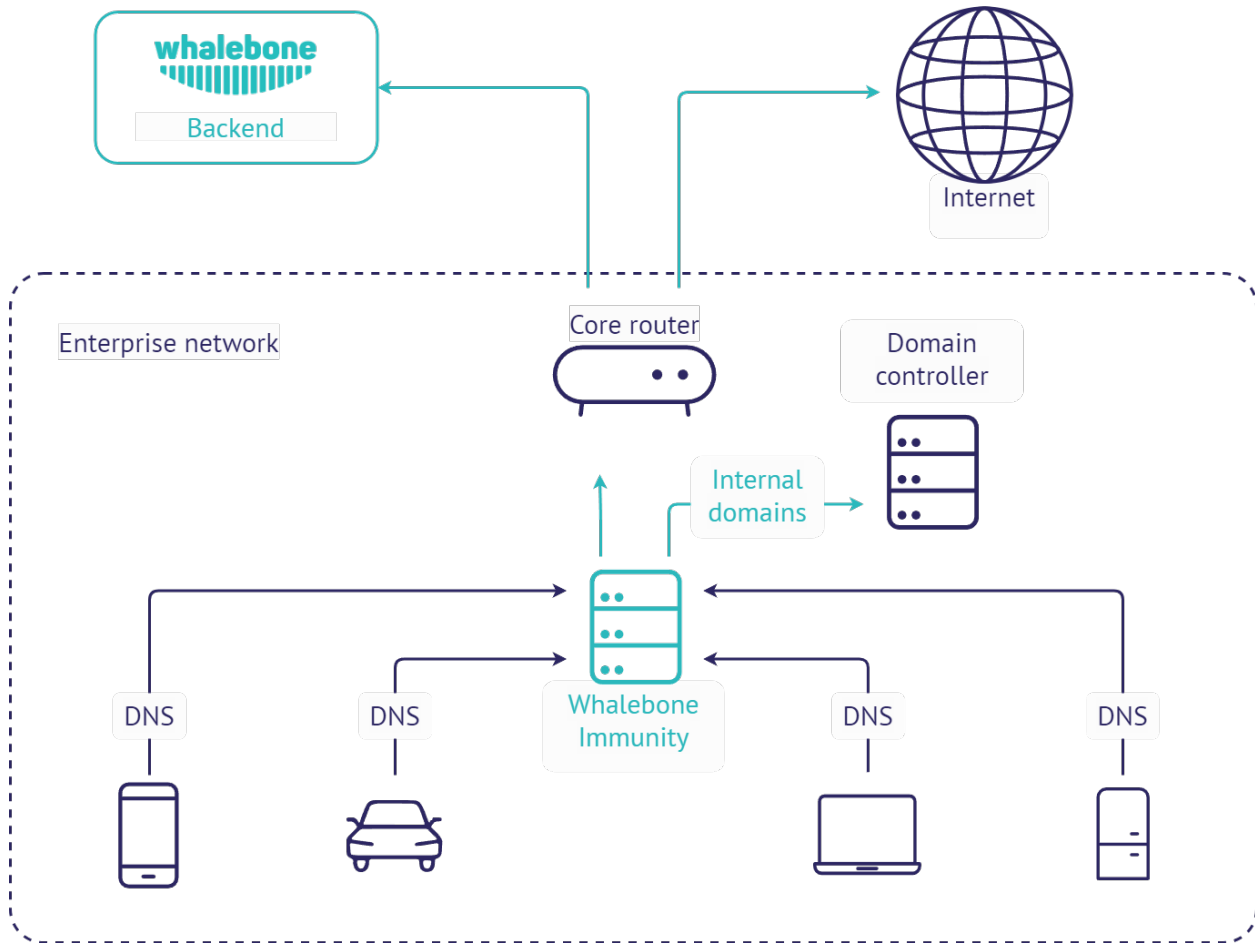


WHALEBONE IMMUNITY

2.1 Local DNS forwarder

Very similar deployment scenario as the local resolver, however Whalebone just forwards the requests for local domains to preconfigured resolvers. This scenario is very useful in case there are local DNS zones that has to be available for the clients (e.g. Active Directory) or cases when the recent resolver configuration is very specific and has to be preserved. This deployment has also lower hardware requirements, roughly half of the CPU and RAM recommended.

Warning: We don't recommend to forward the requests from the local resolver to Whalebone cloud resolvers. Such configuration would result in duplicit incident detection, no added security and unnecessary latency for the clients.



CLOUD DEPLOYMENT

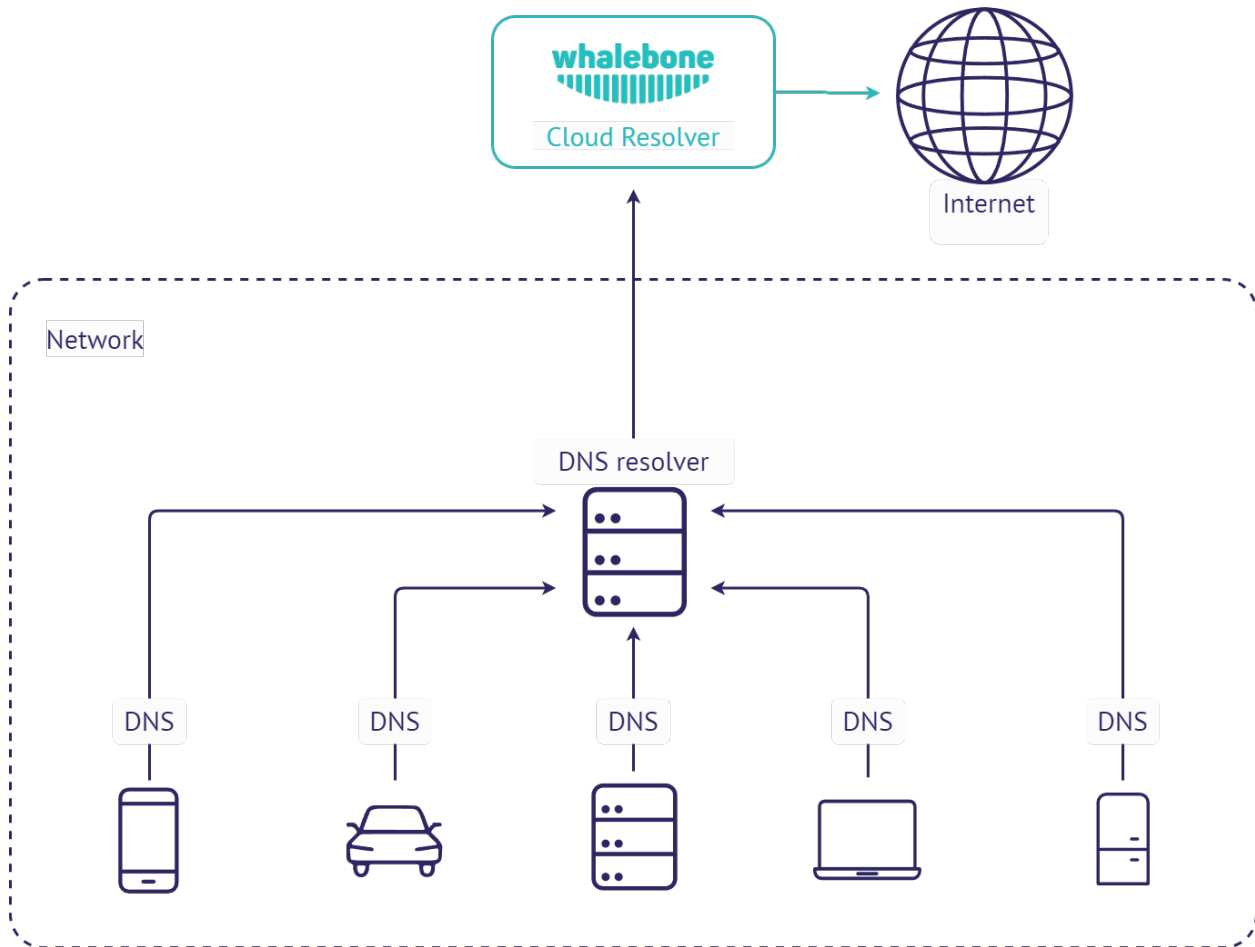
Whalebone could be deployed in several scenarios which can be even combined to fulfill requirements of particular networks. Combination of cloud and local DNS resolver with single management console will serve even complex and distributed networks.

Tip: All of the options below could be combined together. Various network segments and zones could have different requirements and possibilities.

Tip: If neither configuration scenarios below is suitable to your use case, please contact Whalebone Support and we will help you with architecture that will suit your needs.

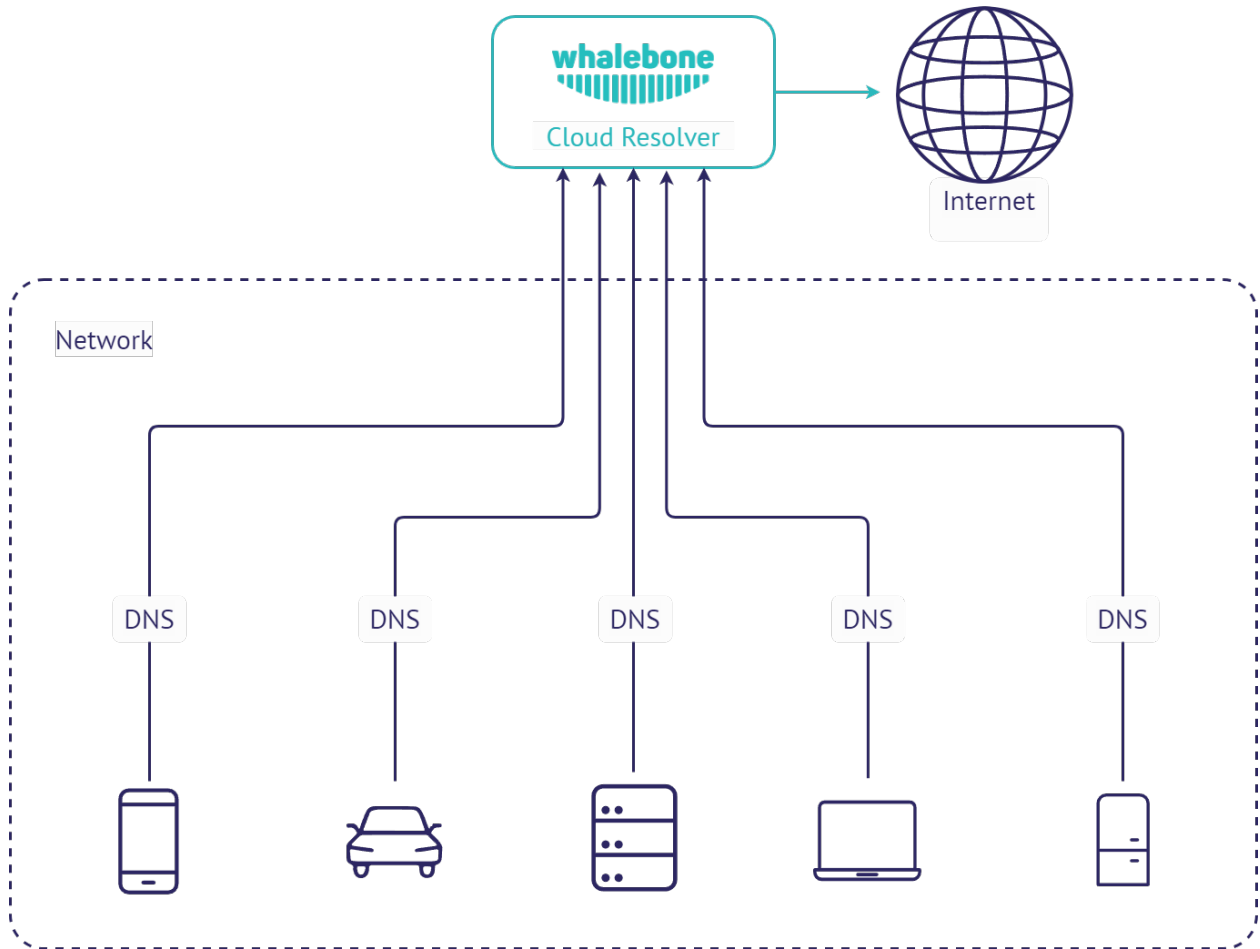
3.1 Use existing DNS to forward to Whalebone Cloud DNS

This is the simplest method of deployment. To use Whalebone filtering, just change the configuration of your recent DNS resolvers and point them to Whalebone cloud resolvers. The downside of this deployment is that all of the incidents will be visible with source IP of the DNS forwarder instead of the original source IP. Still this deployment could come in handy if the priority is to prevent the threats with as low effort and infrastructure changes as possible.



3.2 Cloud DNS (direct connection)



This deployment is similar to forwarding the requests to Whalebone cloud resolvers, but the requests are sent directly to the cloud without local DNS cache. This could be usually set for all endpoints through DHCP. However not using local DNS cache means increased latency introduced by the network communication between the client and cloud resolver. If the individual machines are not hidden behind a NAT, their IP addresses will be directly visible in the Whalebone reporting and the clients can be easily distinguished.



QUICKSTART

4.1 Creating the portal account



After accessing the URL from your activation email, you will be asked to setup the password for your account. We don't enforce any password complexity but we recommend using unique and non-trivial password. An unauthorized access would be a threat to users privacy and could misuse the configuration to harm your network.




Please set your password

Password	<input type="password" value="Web portal password"/>
Password confirmation	<input type="password" value="Your password again"/>

After the password setup you will be asked to login using your username and newly created password.



Your account was activated successfully. You can now login with your password. 

Please sign in

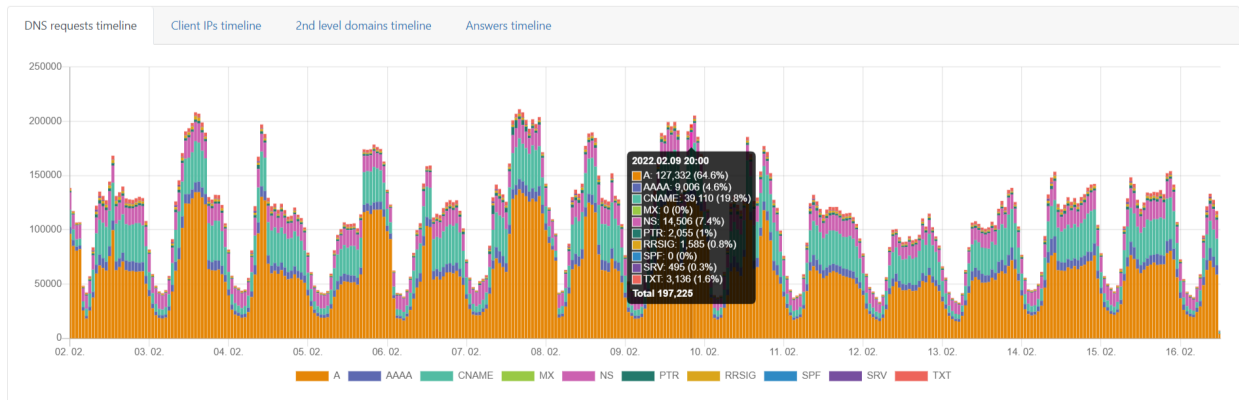
<input type="text" value="username"/>
<input type="password" value="Password"/>

☐ Remember on this device

[Forgotten password?](#)

4.2 DNS traffic view

If the traffic is correctly forwarded to Whalebone DNS resolvers (cloud or local), the DNS traffic will be visible under the menu option **DNS traffic**, where the individual requests and responses are available for further investigation. The traffic should be visible in several minutes after everything has been properly setup. If there is no traffic recorded even in several hours don't hesitate to contact Whalebone support to help you doublecheck the configuration or any sort of network issues.



The DNS resolution check could be also done manually on Windows or Linux machines through nslookup tool. Set the Whalebone resolver IP and try to resolve an existing domain name.

```
localhost:~$ nslookup whalebone.io
Server:          193.32.92.32
Address:         193.32.92.32#53

Non-authoritative answer:
Name:   whalebone.io
Address: 75.2.70.75
Name:   whalebone.io
Address: 99.83.190.102
```

LOCAL RESOLVER

Deploying the Whalebone solution deployed as a **local resolver** brings the advantage of visibility of local IP addresses that send the actual requests. If deploying locally is not a suitable option for you, check out the other Deployment Options.

Whalebone resolver is based on the implementation of [Knot Resolver](#) developed in the CZ.NIC labs.

5.1 Local resolver system requirements

Local resolver is supported on dedicated (hardware or virtual) machine running a supported operating system.

- **Supported operating system** (64-bit, server editions of following distributions):

- Red Hat Enterprise Linux 7, 8, 9
- CentOS Linux 7, 8
- CentOS Stream 8, 9
- Debian 9, 10, 11, 12
- Ubuntu 16.04, 18.04, 20.04, 22.04

- **Supported filesystems**

- ext4
- xfs only with d_type support (ftype=1)

- **Minimum hardware sizing** (physical or virtual):

- 2 CPU cores
- 4 GB RAM
- 40 GB HDD (at least 30 GB in /var partition)

Warning: Please note that Whalebone only supports deployments without desktop environments such as GNOME, KDE or Xfce as those can impact available memory and DNS processing on the server.

- **Network setup requirements** (local resolver needs the following egress ports opened):

Direction	Protocol(s)	Port	Destination IP/Domain	Description
Outbound	TCP+UDP	53	Any	DNS resolution
Outbound	TCP	443	resolverapi.whalebone.io	Threat Database updates
Outbound	TCP	443	stream.whalebone.io	Threat Database updates
Outbound	TCP	443	logger.whalebone.io	Logging stream
Outbound	TCP	443	agentapi.whalebone.io	Resolver management
Outbound	TCP	443	transfer.whalebone.io	Support Log collection
Outbound	TCP	443	portal.whalebone.io	Admin portal
Outbound	TCP	443	harbor.whalebone.io	Resolver updates
Outbound	TCP	443	download.docker.com	Installation Process
Outbound	TCP	443	data.iana.org	DNSSEC keys

Warning: Without communication on port 443 to the domains listed above the resolver won't be installed at all (the installation script will abort).

The main function of the resolver is to get queries from the customers and answer back to them. The answer requires certain ports to be opened on the resolver for the traffic originating from the client subnet or coming to the customer interface.

Direction	Protocol	Port	Source IP/Domain	Description
Inbound	TCP+UDP	53	Customer's subnet range(s)	DNS
Inbound	TCP	853	Customer's subnet range(s)	DNS over TLS (if used)
Inbound	TCP	443	Customer's subnet range(s)	DNS over HTTPS (if used)

The Blocking Pages are being hosted **directly** on the Resolvers so the IP addresses that are advertised to the clients must be used. The clients will then be redirected to the IP address of the resolver upon blocking. It is advised to allow only subnet(s) assigned to customers or trusted networks, otherwise it can be misused for various attacks or unauthorized users.

Direction	Protocol	Port	Source IP/Domain	Description
Inbound	TCP	80	Customer's subnet range(s)	Redirection/Blocking page
Inbound	TCP	443	Customer's subnet range(s)	Redirection/Blocking page

The resolver's processes need to communicate on localhost. In case some firewall is in place please make sure that the traffic is allowed, i.e. `iptables -A INPUT -s 127.0.0.1 -j ACCEPT`

Direction	Protocol	Port	Source IP/Domain	Description
Inbound	TCP	ANY	127.0.0.1	Resolver's processes communication

Note: For hardware sizing estimation of large ISP or Enterprise networks feel free to contact Whalebone. Whalebone local resolver will need approx. twice the RAM and CPU than usual resolver (BIND, Unbound).

5.2 Installation of a new local resolver

You can watch step-by-step video guide about the installation procedure [here](#).

In menu **Resolvers** press the button **Create new**. Choose a name (identifier) for your new resolver. The input is purely informative and won't affect the functionality. Once you've entered the name, click **Add resolver** button. After clicking the button an informative window will pop up with list of supported platforms and the one-line command for the installation. Copy the command and run on the machine dedicated for the local resolver. The command will run the installation script and will pass the one time token used for the resolver activation (the same command can't be used repeatedly).

Once the command is run the operating system is being checked and requirements installed. Script will inform you about the progress and it creates a detailed log named `wb_install.log` in current directory. Successful run of the installation script is ended with the notification ``Final tuning of the OS`` with value `[OK]`. Right after the installation also the initialization takes place and it could take several minutes before the resolver starts the services.

Warning: Local resolver is configured as an open resolver. It will respond to any request sent. This is quite comfortable in terms of availability of the services, but also could be a risk if the service is available from the outside networks. Please make sure you limit the access to the local resolver on port 53 (UDP and TCP) from the trusted networks only, otherwise it can be misused for various DoS attacks.

Important: The resolver's processes need to communicate on localhost. In case some firewall is in place please make sure that the traffic is allowed, i.e. `iptables -A INPUT -s 127.0.0.1 -j ACCEPT`

5.2.1 Verifying the installation

Whalebone resolvers come with a set of testing domains for the verification of the installation and the Security filtering. These domains can be used in order to ensure that you are effectively using a Whalebone resolver:

- `http://malware.test.attacker.online`
- `http://c2server.test.attacker.online`
- `http://spam.test.attacker.online`
- `http://phishing.test.attacker.online`
- `http://coinminer.test.attacker.online`

Upon visiting these domains a blocking page similar to the following should be presented:

In case you come across the page below, it means that the request was not blocked and thus a Whalebone resolver is not being used. Please review your settings and if the issue persists, please contact support.

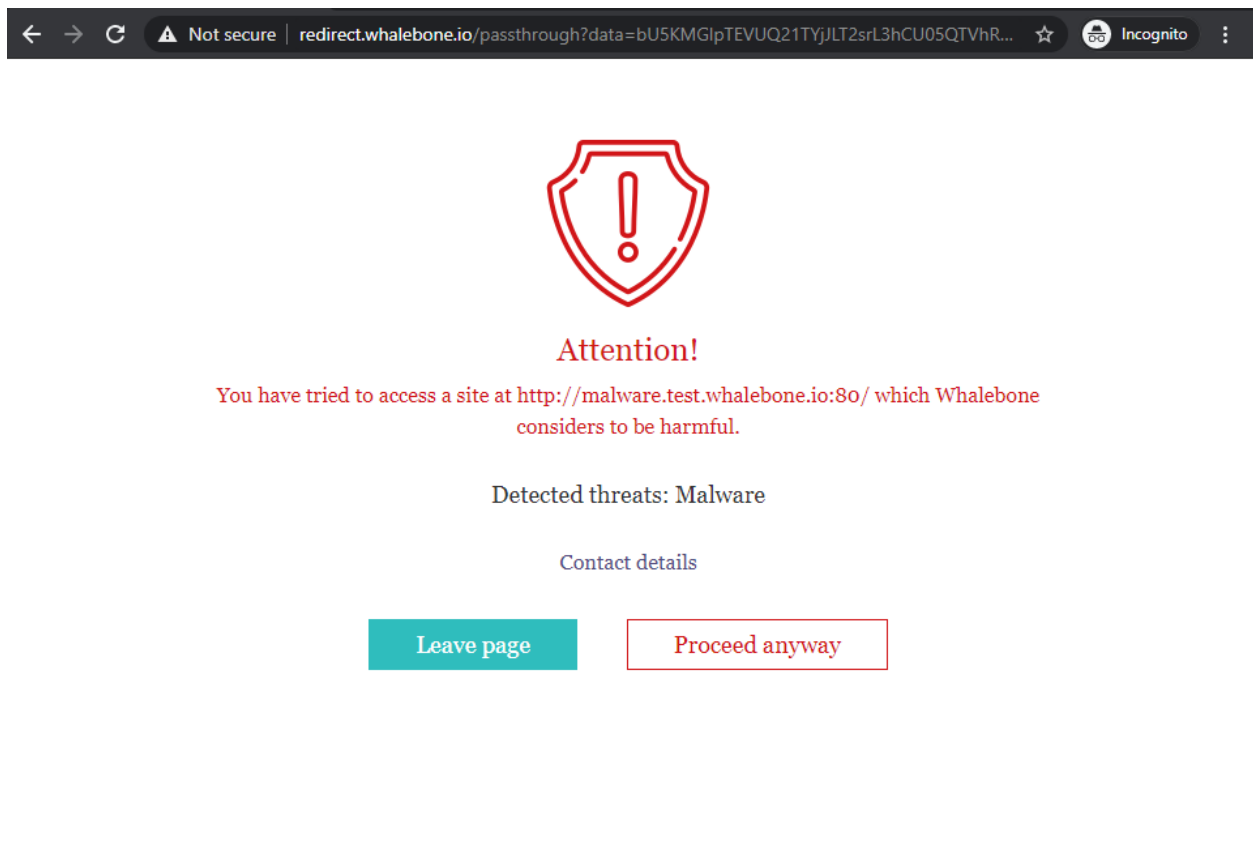


Fig. 1: Blocking Page - Whalebone Resolver is being used.

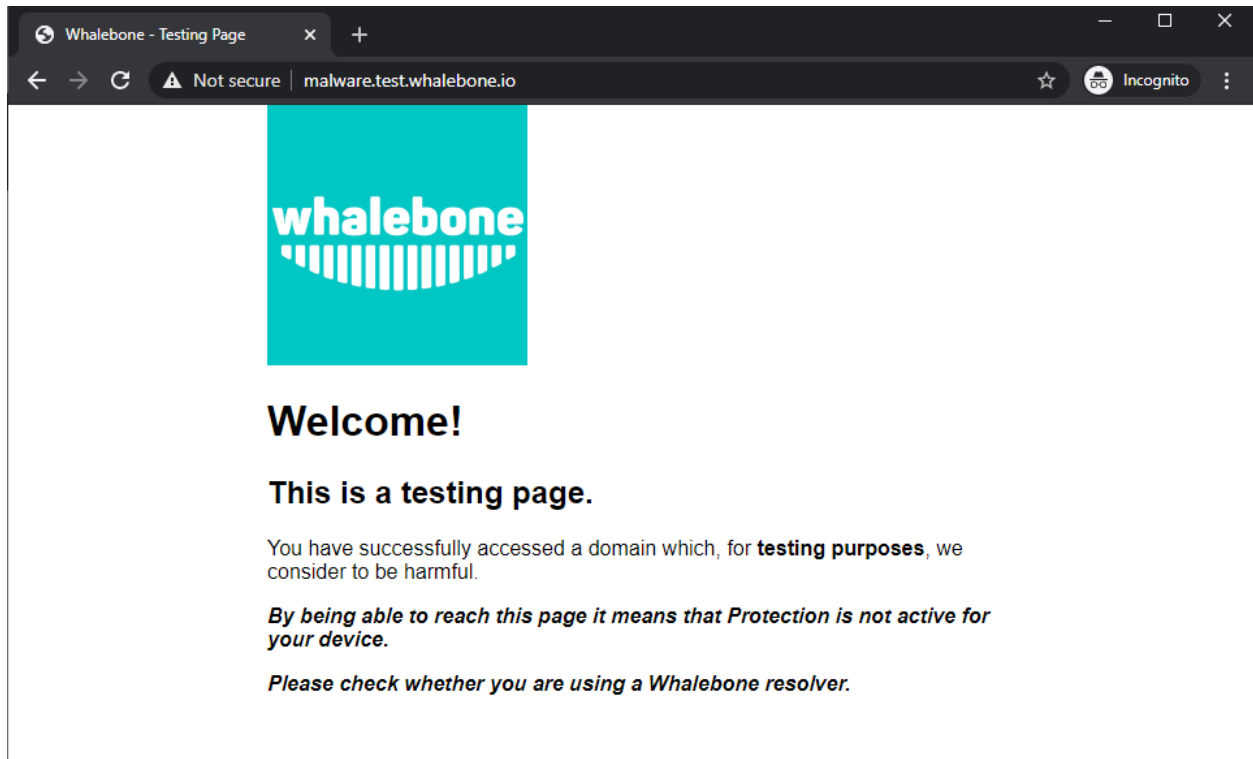


Fig. 2: Blocking Page - Whalebone Resolver is not being used.

5.2.2 Securing your resolver

Upon initial installation, the resolver is configured as an open resolver. It will respond to any request sent to it regardless of where the request originated from. This is quite comfortable in terms of availability of the services, but could also be a risk if the service is available from the outside networks. Please make sure you limit the access to the local resolver on port 53 (UDP and TCP) from the trusted networks only, otherwise it can be misused for various DoS attacks.

RESOLVER MANAGEMENT

On the **Resolvers** page there is an overview of created resolvers. Administrator can adjust the configuration, deploy updates and install new resolvers.

6.1 Resolvers overview

In the main resolver overview there are tiles with resolver details. The overview includes information about operating system and resources as CPU, Memory and HDD usage. There is also the status of the communication channel between the resolver and the cloud indicated by the color-coded dot.

The resolver can be in one of these states:

- **Active** - This is the expected status in production environments and signalizes that everything is running correctly.
- **Resolution problem** - The resolver is unable to translate DNS requests.
- **Unavailable** - The resolver has lost connection with Whalebone Cloud. This state does not affect the DNS translation however the resolver cannot get threat database updates and might not respond to policy or configuration changes initiated from the portal.
- **Upgrading** - An upgrade command has been issued to the resolver. This state should not persist for more than a few minutes.
- **Not installed** - The resolver was not yet installed.

6.2 Deploy configuration

If there been are any configuration changes which affect the DNS resolution, you have to **deploy** the configuration afterwards. Otherwise the changes will not take effect. In case there are any configuration changes available to be deployed, there will be a **red icon** with down right arrow visible on the resolver card. Once clicked, the webpage will ask for confirmation and the successful deployment will be notified in the top right corner.

Note: If the deployment resulted in error, try to repeat the action. The reason for the error could be a short term communication outage between the cloud and the resolver.

6.3 Configure Policy per Network Segment

Security and content policies can be assigned in a granular manner to different segments of the network.

The setting applies per resolver and can be configured under **Resolvers** → Name of the resolver → **Policy Assignment**

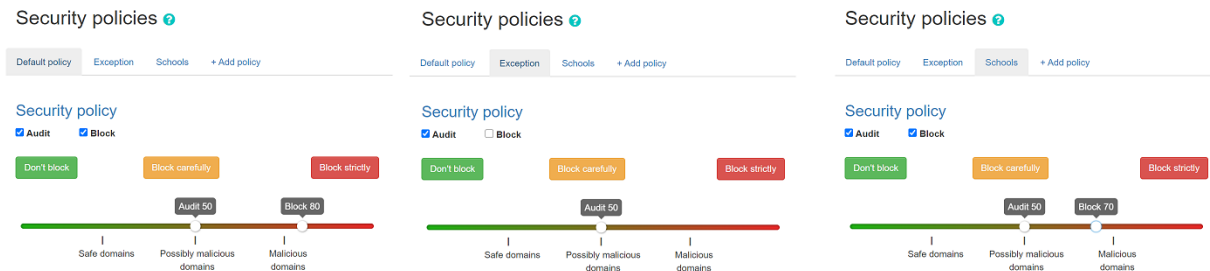
Note: The configuration is **per resolver**. In case you want to apply the configuration to more than one resolvers, please modify all the necessary resolvers.

The policies can be applied by adding IP ranges in the available input form:

In order to provide a better understanding let's consider an example with the network range 10.10.0.0/16.

We have created 3 different policies:

- **Default:** the policy that we want to apply to the whole network, this is the most generic policy
- **Exception:** a policy that must be applied to a specific segment in the network which will have all security and content filtering disabled.
- **School:** a policy that we want to apply to 2 different subnets that have been assigned to school environments. In this case we have chosen to be more strict in the blocking.



Example policy settings. ... note:: The first setting option for policies is for all the undefined ranges. In case different policies affecting same range the more granular is applied.

Let's summarize the requirements in the following matrix:

Policy	Network
Default	10.10.0.0/16
Exception	10.10.10.0/24
School	10.10.20.0/24 and 10.10.40.0/24

The following image shows the process of assigning the policies:

Statistics
Policy assignment
Advanced configuration
Upgrade
Actions log

All changes require deployment of policies to resolver from resolvers list page with button

Blocking page settings

Blocking page location ☒ Whalebone Cloud ☐ On-premise local resolver

Policy matching strategy

Match policy based on IP of ☒ Client ☐ Resolver

Policy assignment

IP Range	Policy	Options
<i>This policy applies to all undefined ranges</i>	Block all	
10.10.0.0/16	Default policy	Remove IP range
10.10.10.0/24	Exception	Remove IP range
10.10.20.0/24 10.10.40.0/24	School	Remove IP range

Add IP range
Save to resolver

Note: After adding the networks you must click on **Save to resolver** in order to take effect. The changes will be then validated and a pop-up message will provide additional information.

In order to assign additional entries to an existing assignment, a new network range can be appended using *newline* as a separator. Building on the previous example, in case we wanted to add the subnet `10.10.30.0/24` to the Exception Policy:

6.4 Configure Blocking Pages

In a similar manner to the Security Policies, the Blocking Pages can also be assigned to particular network ranges.

The first step is to select **On-premise local resolver** for the **Blocking Page Location** option. Two new fields are enabled where the IPv4 and IPv6 addresses of the Blocking Pages must be filled in.

Tip: The Blocking Pages are being hosted **directly** on the Resolvers so the IP addresses that are advertised to the clients must be used. The clients will then be redirected to the IP address of the resolver upon blocking. Please ensure that ports 80 and 443 are accessible on the firewall.

For each IP range that is added, there is a drop-down menu for the Blocking Page that should be assigned.

Important: The first entry in the **Policy Assignment** is considered the Default/Fallback. In case a client accesses the resolver from an undefined IP range, the respective options will apply.

Note: After making the necessary changes to the Blocking Page settings, please check whether the resolvers need to

Local resolver Whalebone ← Back to resolvers page

- Statistics
- Policy assignment**
- Advanced configuration
- Upgrade

Blocking page settings

Blocking page location ☐ Whalebone Cloud ☒ On-premise local resolver

Policy assignment

IP Range	Policy	Blocking page	Options
<div> <div></div> <div>This policy applies to all undefined ranges</div> </div>	Default policy	Default	<input checked="" type="checkbox"/> Enable bypass

+ Add IP range
Save to resolver

Fig. 1: Assign Blocking Page to IP range

be re-deployed.

6.5 Upgrade/Rollback Resolver

When a new version of the Resolver is released, a **red upgrade icon** appears on the resolver's management interface.

Your local resolvers ? + Create new

! There is available upgrade for resolvers. You can find it under the icon 👤

Local Resolver

#0001

🔧
🔧
🗑️

Hostname: whalebone

Operating system: Ubuntu

Status: ● Active

IP 10.10.10.10

🕒 Updated 42 seconds ago

CPU: 0.5 %

RAM: 48.2 %

HDD: 30.6 %

Upon clicking on the **Upgrade** icon, the respective menu is selected and important information about the new release is provided.

Local resolver whalebone

[← Back to resolvers page](#)

Statistics
Policy assignment
Advanced configuration
Upgrade

Upgrade

Available upgrade

Upgrades version

Rollback

2020.10.12 13:54:33 Stable Version 35

[Initiate update](#)

- Software update source for Whalebone resolver is now <https://harbor.whalebone.io> (please check your firewall rules)
- Blocking page is reworked from the scratch (originally referred to as "Sinkhole")
 - You can find the configuration in Configuration -> Blocking pages and the activation can be done in the resolver details in Policy assignment
 - It is hosted directly on the resolver (ports TCP/80,443 has to be reachable from clients)
 - Full access to html code editor
 - Feature "Continue anyway" - user can decide to continue to the destination malicious website on his own
 - Different blocking pages per IP or subnet - could be used to customize the blocking page for a specific customer (school, government office, etc.)
 - Definition of supported languages and a default language (for browsers that do not tell which language they prefer if any)
 - Knot resolver updated to version 5.1.3 (from version 5.1.1)
- Based on DNS Flag Day 2020 recommendation that EDNS buffer size is adjusted to 1232 bytes
- Management Agent for cloud communication is now independently monitored and if there are any issues, it is automatically restarted (no impact on DNS resolution)

[↑ lr-agent 1.4.4](#)
[↑ resolver 5.1.3-3-2](#)
[↑ kresman 3.2.2](#)
[passivedns 1.1.3](#)
[logstream 2.1](#)
[logrotate 1.1](#)
[logcat 1.1](#)
[logcat-content 1.1](#)

Services highlighted in green will be updated

From this menu, the upgrade of the resolver can be initiated.

In case the installation of the new version does not yield the expected outcome, a rollback to the previous version is possible anytime in the **Rollback** tab:

Local resolver whalebone

[← Back to resolvers page](#)

Statistics
Policy assignment
Advanced configuration
Upgrade

Upgrade

Available upgrade

Upgrades version

Rollback

2020.04.28 12:37:22 Stable Version 27

[Back to previous version](#)

[↓ lr-agent 1.4.2](#)
[↓ resolver 5.1.1-1](#)
[↓ kresman 3.1.7](#)
[passivedns 1.1.3](#)
[logstream 2.1](#)
[logrotate 1.1](#)
[logcat 1.1](#)
[logcat-content 1.1](#)

SECURITY POLICIES

You can watch step-by-step video guide of basic security policy configuration [here](#).

The step-by-step video guide with deeper explanation of security policy tuning is [here](#).

To control Whalebone's security filtering you need to configure it's security policies. When you install Whalebone, it comes with a **Default** policy which is set to include all threat types and sets the thresholds to the value of **80/50**. This policy will also be automatically applied to every newly installed resolver. In any policy there are several options to be configured:

7.1 Malicious filtering thresholds

Every domain in our threat intelligence database has certain value of the score. The score represents how malicious we believe that particular domain to be. In the policy you adjust two values related to the score:

- **The blocking threshold** - Domains with a score higher or equal to this value will be blocked by Whalebone and the client request will be answered with an IP address of the blocking page.
- **The audit threshold** - Domains with a score higher or equal to this value, but lower than the blocking threshold will be monitored. The request will be allowed and the answer will be served either from cache or by performing the full DNS recursion. Requests will however be monitored in the Threats dashboard for later investigation, if needed.

Individual actions could be turned off - e.g. turn off the blocking for testing purposes. The slider values define the probability that the particular domain is malicious on the scale from **0** to **100** with **100** being the most malicious.

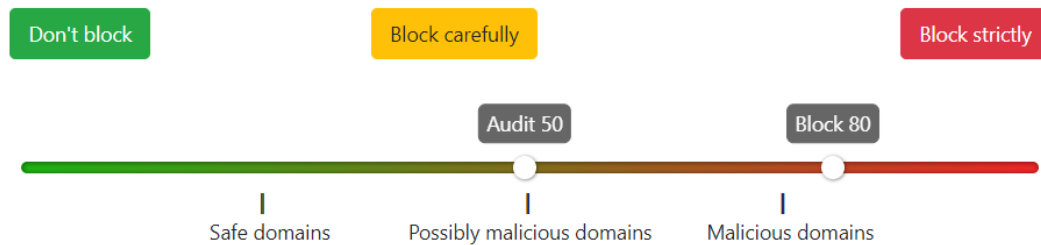
Tip: The default threshold for blocking is set to **80** which is safe even for larger network with liberal policy towards the users. For more restrictive policy we suggest setting threshold for blocking to **70-75**, in very restrictive networks even down to **60**. Audit is purely informative, however setting the threshold too low can result in too many logged incidents.

There are preconfigured policies available that cover the most usual cases. These cases are: **Don't Block**, **Block carefully** and **Block strictly**.

- **Block carefully** setting prioritizes a low false positive rate and is suitable for ISPs.
- **Block strictly** maximizes the detection rate and is suitable for most corporate deployments.
- **Don't block** turns off the blocking entirely and causes Whalebone to operate in a transparent/permissive mode, where it will only log (audit) the incidents but it won't actively block them.

Security policy

☒ Audit

☒ Block


Types of threats

☒ Include all types of threats

You can configure additional policies by clicking the **Add Policy** tab. First you select which of the existing policies you want the new policy to be based upon. Then click the pencil button under the **Name of the Policy** to clearly differentiate it from the others. You can then modify the blocking and auditing sensitivity, add deny lists or set up regulatory filtering. The new policy is not saved until you click the **Save** button.

Tip: The policy is not active unless it is assigned to some resolvers (local or cloud ones). To start enforcing the policy, navigate to **Resolvers** → **Policy Assignment** and assign it to a specific **subnet** or **resolver**.

7.2 Types of threats

The default settings is to include all types of threats. If you want to exclude some you can do so by unchecking the box **Include all types of threats**. From the drop-down menu you can now choose the specific categories of audited/blocked threats. The available categories are: **blacklist**, **c&c**, **coinminer**, **compromised**, **malware**, **phishing** and **spam**.

A full list of what each category includes can be found below:

- **C&C (Command and Control)**: domains that facilitate botnet communication to coordinate its activity. A botnet is a network of infected computers, which are controlled as a group.
- **Malware**: domains that host and distribute any kind of malicious code.
- **Phishing**: domains aiming to trick users and extract sensitive information such as credit card details, login credentials, etc.
- **Blacklist**: domains that are known to serve multiple nefarious purposes at the same time or over a period of time.
- **Spam**: domains that are linked to spreading spam emails and scam schemes.
- **Compromised**: otherwise legitimate domains that have been hacked and are temporarily used for malicious purposes.
- **Coinminer**: domains that hijack processing and energy resources for unsolicited cryptocurrency mining.

Note: Any changes in the Security Policies will be applied to the resolvers in approx. 2-3 minutes. Saved configuration is used during preparation of the threat data package for the resolvers that download and apply those packages at regular intervals.

7.3 Allow lists

- Domains that will never be blocked (unless they are also present in a regulatory compliance feed).
- The allow list has the second highest priority when evaluating how to resolve a domain.
- The allow list is applied to the domain and all of the subdomains, e.g.: allowed domain `whalebone.io` will also allow `docs.whalebone.io`, but not vice versa.
- The list can be configured on the **Allow / Deny List** tab on the left side of **Configuration** page.
- One list can hold up to 10 000 domains.

7.4 Deny Lists

- Domains that will be blocked at all times (unless the same domain is also present on an allow list).
- The deny list is applied to the domain and all of the subdomains, e.g.: denied domain `malware.ninja` will also deny `super.malware.ninja`, but not vice versa.
- The list can be configured on the **Allow / Deny List** tab on the right side of **Configuration** page.
- One list can hold up to 10 000 domains.

The custom lists support a *Lex specialis derogat legi generali* principle, in which a more specific domain listing overrides a more general domain listing. This way, you can have the whole domain `malware.ninja` on a Deny list but if you have `friendly.malware.ninja` on an Allow list, this will take precedence and communication to this site will act as an exception and will be allowed by the resolver.

Warning: After creating an allow or deny list, it needs to be assigned to the specific security policy, or else the changes will not take effect.

7.5 Regulatory Restrictions

- Integrated list of domains that must be applied in order to conform to Regulatory Restrictions of a country.
- Examples of these domains include cases of illegal gambling or child pornography.
- Domains on the regulatory restrictions list will be always blocked, if the list is applied to the security policy.
- They have the highest priority and their filtering cannot be overridden. Not even adding a domain to an allow list will cause the resolver to stop blocking it.

Warning: Each country has different Regulatory lists. In case of multi-country deployments different policies can be used in order to apply the proper Regulatory Restrictions.

7.6 Content Filtering


Particular Content categories can be applied on a per-policy level. This is useful in case different segments of the networks come with different requirements. For example, in case of a School environment all the **Adult** categories can be enabled and access to relevant content can be restricted.

A diverse set of content filtering categories are available:

- **Sexual content:** Sexual and pornographic material,
- **Gambling:** games and activities involving betting money,
- **Weapons:** guns and weapon-related sites,
- **Audio-video:** audio and video streaming services,
- **Games:** online games and gaming websites,
- **Chat:** instant messaging and chatting applications,
- **Social-networks:** social networking sites and applications,
- **Child abuse:** websites related to child abuse dissemination of child pornography,
- **Drugs:** drug related websites including alcohol and tobacco,
- **Racism:** content linked to racism and xenophobia,
- **Violence:** explicit violence and gore,
- **Terrorism:** domains linked to terrorism support,
- **Advertisement:** banners, context advertisements and other advertisements systems,
- **Crypto-mining:** domains connected to crypto-currency mining activities,
- **DoH:** DNS over HTTPS. These are domains that provide obfuscation of the DNS requests in HTTP traffic,
- **P2P:** domains linked to peer to peer networks where multimedia content is shared by the users,
- **Tracking:** web and email tracking systems.

The content filter can also be applied for specific times of the day. When a particular category is ticked, a clock icon will appear next to it. If you click the clock icon, you can add a new schedule for this category. Multiple schedules may be active for the same category. This way, you may only allow access to social networks during the lunch break and after working hours. Finish the settings by clicking **Apply** and **Save** the security policy.

Allow category 'social networks' in the following times



12:00 ▾

–

12:59 ▾

Everyday

Monday

Tuesday


Wednesday

Thursday

Friday

Saturday

Sunday



17:00 ▾

–

23:59 ▾

Everyday

Monday

Tuesday

Wednesday

Thursday

Friday

Saturday

Sunday

+ Add schedule

Apply

Note: By applying the schedule, you are **allowing** access to domains from that content category during that specific time period.

DNS RESOLUTION CONFIGURATION

You can find the options to configure the resolver in the menu **Configuration** and tab **DNS resolution**. This page allows you to do the basic configuration without the knowledge of configuration syntax. Furthermore there is a text area allowing you to define any configuration to the underlying [Knot Resolver](#).

Available configuration options:

- **Enable IPv6**
 - If the system has IPv6 configured properly its is possible to enable IPv6.
 - Otherwise the activation of IPv6 could have negative effects on the performance and latency of the resolver.
- **Forward queries to**
 - This option allows to redirect all or chosen queries to upstream resolvers or authoritative DNS servers (suitable e.g. for forwarding to domain controllers of Active Directory).
 - **Disable DNSSEC**
 - * If checked, the answers from the forwarded queries won't be DNSSEC validated.
 - * We recommend to check this option in case the upstream server don't have DNSSEC configured properly.
 - **All queries to**
 - * Option to forward all queries to one or more resolver.
 - * This option keeps caching all responses!
 - **Following domains** * Option to choose particular domains that should be forwarded to on more resolvers.
 - * Different resolvers could be defined for different domains. * Caching for the selected domains will be turned off!.
- **Static records**
 - Predefined answers that should be returned for particular domains.
 - Could serve for special purposes such as monitoring or very simple substitution of records on authoritative server.
- **Advanced DNS configuration**
 - Text area for advanced configuration.
 - Used for direct configuration of Knot Resolver.
 - [Complete Knot Resolver configuration](#)
 - Supports Lua scripting.

Warning: Faulty configuration can impact stability, performance or security functions of the resolver. In case of wrong syntax the **Deploy Configuration** will result in error code.

Note: Once the **Save** button is pressed changes in DNS resolution are saved and prepared to be deployed to target resolvers. The deployment itself has to be done from the **Resolvers** page. It is possible to do multiple changes and apply all of them at once to minimize the number of deployments to the resolver.

KNOT RESOLVER - TIPS & TRICKS

Advanced configuration of Whalebone resolver allows to apply any Knot Resolver configuration. In this section we are going to describe the most frequent use cases and examples of such configuration snippets. Views, policies and their actions are evaluated in the sequence as they are defined (except special chain actions that are described in the official Knot Resolver documentation). First match will execute the action, the rest of the policy rules is not evaluated. If you are going to combine different configuration snippets, you can load the same module just once at the beginning of the configuration.

9.1 Allow particular IP ranges

Define a list of IP ranges that will be allowed to use this DNS resolver. Queries from all other ranges will be refused.

```
-- load modules
modules = {'policy', 'view'}

--define list of ranges to allow
--127.0.0.1 should always be allowed
allowed = {
    '127.0.0.1/32',
    '10.10.20.5/32',
    '10.30.10.0/24'
}

-- allow list of ranges
for i,subnet in ipairs(allowed) do
    view:addr(subnet, policy.all(policy.PASS))
end

-- block all other ranges
view:addr('0.0.0.0/0', policy.all(policy.DENY))
```


9.2 Refuse particular IP ranges

Define a list of IP ranges that will be blocked to use this DNS resolver. Queries from all other ranges will be allowed.

```
-- load modules
modules = {'policy', 'view'}

--define list of ranges to block
blocked = {
    '10.10.20.5/32',
    '10.30.10.0/24'
}

-- block list of ranges
for i,subnet in ipairs(blocked) do
    view:addr(subnet, policy.all(policy.REFUSE))
end
```

9.3 Allow list of domains

```
-- load modules
modules = {'policy'}

--define list of allowed domains
domains = {
    'example.com',
    'anotherexample.org'
}

-- allow list of domains
for i,domain in ipairs(domains) do
    policy.add(policy.suffix(policy.PASS, {todname(domain)}))
end
```

9.4 Deny list of domains

```
-- load modules
modules = {'policy'}

--define list of denied domains
domains = {
    'example.com',
    'anotherexample.org'
}

-- deny list of domains, while returning NXDOMAIN
for i,domain in ipairs(domains) do
```

(continues on next page)

(continued from previous page)

```
policy.add(policy.suffix(policy.DENY, {todname(domain)}))  
end
```

9.5 Disable DNSSEC globally

```
trust_anchors.negative = { '.' }
```

9.6 Disable DNSSEC validation for a domain

```
trust_anchors.set_insecure({ 'domain.com' })
```

9.7 Disable Query Case Randomization

```
policy.add(policy.suffix(policy.FLAGS('NO_0X20'), {todname('domain.com')}))
```

9.8 Disable QNAME Minimization

```
policy.add(policy.suffix(policy.FLAGS('NO_MINIMIZE'), {todname('domain.com')}))
```

9.9 Disable Domain caching

```
policy.add(policy.suffix(policy.FLAGS('NO_CACHE'), {todname('domain.com')}))
```

9.10 Enable Prometheus Metrics

The resolver can expose its metrics in Prometheus text format. The following script enables the HTTP module and the respective `/metrics` endpoint is made available.

More information and configuration options can be found on [Knot Resolver Documentation](#)

```
modules.load('http')  
function startHttp ()  
net.listen('127.0.0.1', 8453, { kind = 'webmgmt' })  
end  
pcall(startHttp)
```

BLOCKING PAGES

You can watch step-by-step video guide [here](#).

In case of blocking access to a domain (due to security, content or regulatory reasons), the resolvers are answering to the clients with a specific IP address that leads to one of the Blocking pages. While the clients initiate the HTTP(S) connections towards the blocked domain, they are presented with a custom Blocking page with different content based on the reason of the blocking. For the Blocking Pages Whalebone provides sample template, however, they do not have to be followed and virtually every modification, branding and copywriting is possible. The template code is written to be compatible with the widest range of browsers to avoid problems with older versions.

Different versions of the Blocking Pages can be assigned to different segments of the networks in **Resolvers** → **Policy assignment**.

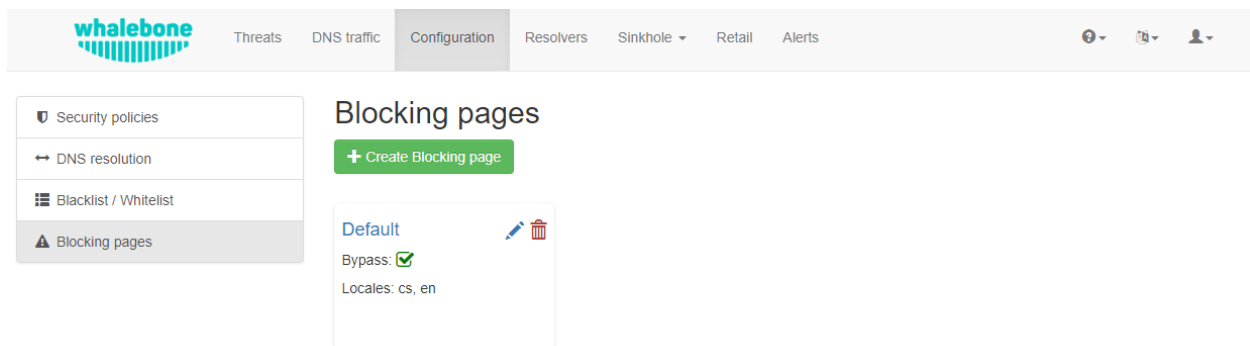


Fig. 1: Blocking Pages Overview

For each version, based on the deployment details, there are four variants of the Blocking Pages that are available and can be configured:

- **Security:** displayed when access is blocked due to security reasons
- **Blacklist:** displayed when access is blocked by the Administrators
- **Regulatory:** displayed when access is regulated due to law or court order
- **Content:** displayed when access is blocked due to the content of the domain

Furthermore, each version can have different localization options. The language that is going to be presented to the user is inferred from the language of the browser that is visiting the Blocking Page. New locales can be seamlessly added as an option.

For each Locale several options are available. In the example above, the English version has the following options:

Blocking pages

[Back to list](#)

Name of Blocking page

Default

Blocking page domain

redirect.whalebone.io

+ Locale

Locale	Security	Blacklist	Regulatory	Content
Czech (cs, *)	22.5kB	21.6kB	21.8kB	21.5kB
English (en)	22.4kB	21.5kB	21.6kB	21.5kB

Save

Fig. 2: Blocking Pages Menu

Option 1. – Use Template:

When using the template option, the information that is provided as input to the following form are injected in the template code. This is the fastest and easiest way to customize the blocking pages.

Note: Setting the blocking page could be done clicking on the **Magic wand** button. Note that it will override the previous version of blocking page.

Option 2. – Set as default locale:

This option can customize the default language of the Blocking Pages. In case some browser does not declare its preferred language, the “Default” language acts as a fallback mechanism. Default locale is indicated via wildcard symbol (*) next to the langue type.

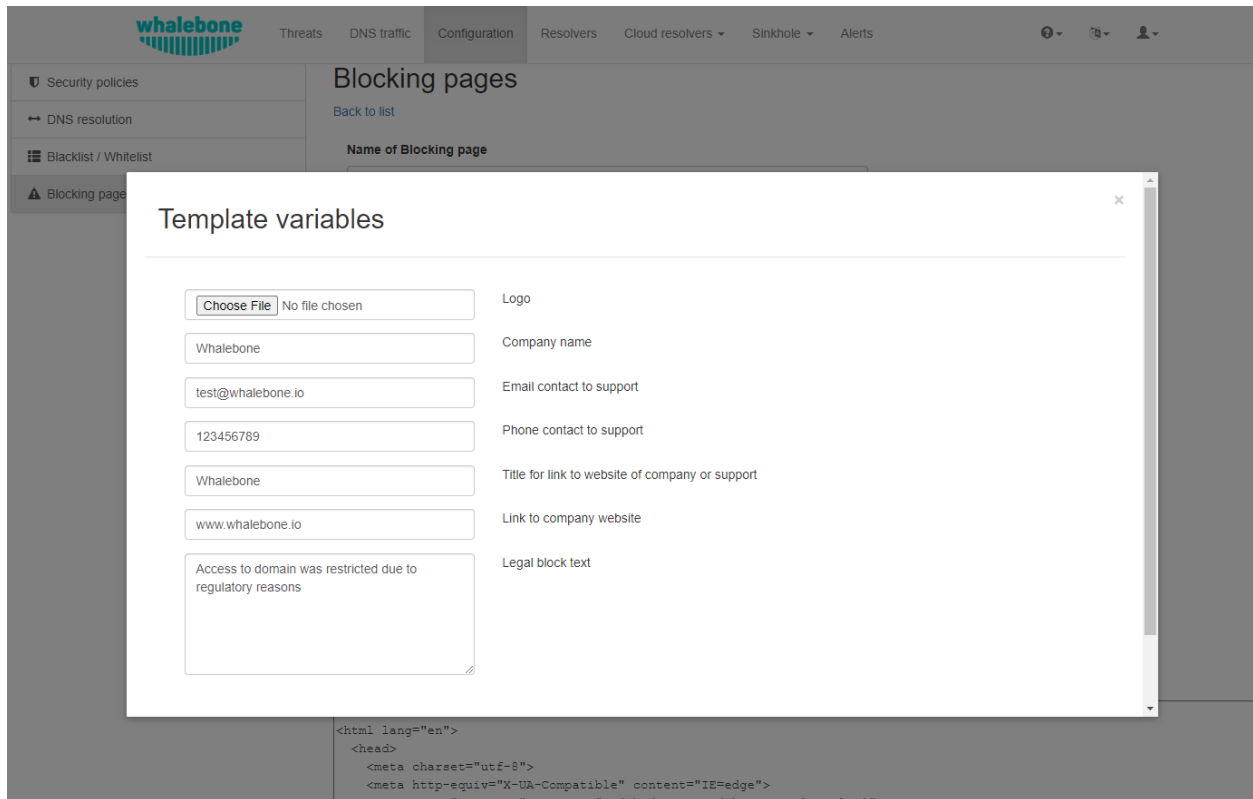
Option 3. – Delete the locale:

Locale could be deleted clicking on **Bin** icon.

Each of the Versions of the Blocking Page (Security, Blacklist, Regulatory, Content) can be customized in more detail by modifying the HTML code. Upon clicking on each version an editor is presented that allows for any required changes.

The editor also exposes a “Verification” interface which parses the final HTML code and checks for the enabled functionalities. The check is based on the id of the specific elements. More information and requirements for each functionality can be found by clicking the respective labels.

Note: Each Version of the Blocking Page has unique characteristics that can be selected. For example, the Security



Blocking Page can include a “Bypass” button which is not available in the respective Regulatory and Blacklist versions.

After editing and saving the changes to the Blocking Pages it is important that they are applied to the individual resolvers.

Tip: The Blocking Pages are served from a web server directly on the Resolver. The pages are expected to be a single file so any additional resources (CSS, images, scripts) must be either embedded directly in the HTML code or served from a publicly accessible web server. The resolver does not provide any option to serve other content.

10.1 Signing blocking pages with a CA

For deployments, where you have control over the endpoints (typically enterprise environment with Group Policy) and you’re able to push self-signed SSL certificates to their trust stores, you can sign the blocking pages on the fly. This results in the browsers going directly to the blocking page without displaying the security warning, which is usually there. The resolver essentially performs a MITM any time it redirects to the blocking pages so the browser warning is expected.

Step 1. – Create “v3_cfg” file with the following contents:

```
[req]
req_extensions = v3_ca_extensions
distinguished_name = req_dn
[v3_ca_extensions]
basicConstraints = CA:TRUE
subjectKeyIdentifier = hash
```

(continues on next page)

(continued from previous page)

```

authorityKeyIdentifier = keyid:always,issuer:always
keyUsage = cRLSign, keyCertSign
subjectAltName = @alt_names
[alt_names]
DNS.1 = localhost
[req_dn]
countryName = Country Name (2 letter code)
countryName_default = US
stateOrProvinceName = State or Province Name (full name)
stateOrProvinceName_default = New York
localityName = Locality Name (eg, city)
localityName_default = New York City
organizationName = Organization Name (eg, company)
organizationName_default = My Organization
commonName = Common Name (eg, your name or your server's hostname)
commonName_max = 64

```

Step 2. – Generate a key:

```
openssl genpkey -algorithm RSA -out /certs/ca.key
```

Step 3. – Create and sign the certificate:

```

openssl req -x509 -new -nodes -key /certs/ca.key -sha256 -days 1024 -out /certs/ca.crt -
↳ config /certs/v3_cfg

```

Step 4. – Export the .pfx file and make sure it is placed in the /certs/ folder:

```

openssl pkcs12 -export -in ca.crt -inkey ca.key -out ca.pfx -certpbe PBE-SHA1-3DES -
↳ keypbe PBE-SHA1-3DES -macal

```

Step 5. – Send the filename and password to Whalebone support to store the configuration persistently on the back-end to ensure that it survives the VM or container restart.

RESOLVER AGENT

11.1 Command line interface

Agent's actions can be invoked using a proxy bash script present at path `/etc/whalebone/cli/cli.sh`. This script calls a python script which handles the execution of the following agent actions:

- **sysinfo** - returns the system status data in JSON format.
 - Parameters: None
 - Output: tested categories on tested key can have two values **OK** and **FAIL**

```
{
  "hostname": "hostname",
  "system": "Linux",
  "platform": "CentOS Linux 7 (Core)",
  "cpu": {
    "count": 4,
    "usage": 28.6
  },
  "memory": {
    "total": 7.6,
    "available": 3.9,
    "usage": 49.2
  },
  "hdd": {
    "total": 50.0,
    "free": 14.4,
    "usage": 71.1
  },
  "swap": {
    "total": 0.0,
    "free": 0.0,
    "usage": 0
  },
  "resolver": {
    "answer.nxdomain": 3284,
    "answer.tc": 35,
    "answer.ad": 849,
    "answer.100ms": 3983,
    "answer.cd": 6,
    "answer.1500ms": 74,
```

(continues on next page)

(continued from previous page)

```

    "answer.slow":215,
    "answer.rd":224337,
    "answer.lms":104683,
    "answer.servfail":215,
    "predict.epoch":24,
    "query.dnssec":6,
    "answer.250ms":14941,
    "query.edns":35498,
    "answer.cached":86713,
    "answer.nodata":3622,
    "answer.aa":2362,
    "answer.do":6,
    "answer.edns0":35498,
    "answer.ra":224337,
    "predict.queue":0,
    "answer.total":224337,
    "answer.10ms":35351,
    "answer.noerror":217216,
    "answer.50ms":59766,
    "answer.500ms":4642,
    "answer.1000ms":653,
    "predict.learned":80
  },
  "docker":{
    "Platform":{
      "Name":""
    },
    "Components":[
      {
        "Name":"Engine",
        "Version":"17.12.1-ce",
        "Details":{
          "ApiVersion":"1.35",
          "Arch":"amd64",
          "BuildTime":"2022-02-27T22:17:54.000000000+00:00",
          "Experimental":"false",
          "GitCommit":"88888fc6",
          "GoVersion":"go1.999.999",
          "KernelVersion":"3.22.66-693.21.1.el7.x86_64",
          "MinAPIVersion":"1.99",
          "Os":"linux"
        }
      }
    ],
    "Version":"19.32.1-ce",
    "ApiVersion":"1.98",
    "MinAPIVersion":"1.12",
    "GitCommit":"7390fc6",
    "GoVersion":"go1.9.4",
    "Os":"linux",
    "Arch":"amd64",
    "KernelVersion":"3.10.0-693.21.1.el7.x86_64",

```

(continues on next page)

(continued from previous page)

```

    "BuildTime":"2018-02-27T22:17:54.000000000+00:00"
  },
  "check":{
    "resolve":"ok",
    "port":"ok"
  },
  "containers":{
    "lr-agent":"running",
    "passivedns":"running",
    "resolver":"running",
    "kresman":"running",
    "pcpy":"running",
    "logrotate":"running",
    "logstream":"running"
  },
  "images":{
    "lr-agent":"whalebone/agent:1.1.1",
    "passivedns":"whalebone/passivedns:1.1.1",
    "resolver":"whalebone/kres:1.1.1",
    "kresman":"whalebone/kresman:1.1.1",
    "logrotate":"whalebone/logrotate:1.1.1",
    "logstream":"whalebone/logstream:1.1.1"
  },
  "error_messages":{
  },
  "interfaces":[
    {
      "name":"lo",
      "addresses":[
        "127.0.0.1",
        "::1",
        "00:00:00:00:00:00"
      ]
    },
    {
      "name":"eth0",
      "addresses":[
        "1.1.1.1",
        "::c8",
        "fe80::",
        "00:00:00:00:00:00"
      ]
    },
    {
      "name":"docker0",
      "addresses":[
        "198.1.1.1",
        "00:00:00:00:00:00"
      ]
    }
  ]
}

```

- **stop - stops up to three containers**

- Parameters: containers to stop (up to 3), Example: `./cli.sh stop resolver lr-agent kresman`

```
{
  'resolver': {'status': 'success'},
  'lr-agent': {'status': 'success'},
  'kresman': {'status': 'success'}
}
```

- **remove - removes up to three containers**

- Parameters: containers to remove (up to 3), Example: `./cli.sh remove resolver lr-agent kresman`

```
{
  'resolver': {'status': 'success'},
  'lr-agent': {'status': 'success'},
  'kresman': {'status': 'success'}
}
```

- **upgrade - upgrades up to three containers, the container's configuration is specified by a docker-compose in agent container (can also be found in a volume `/etc/whalebone/agent`)**

- Parameters: containers to upgrade (up to 3), Example: `./cli.sh upgrade resolver lr-agent kresman`

```
{
  'resolver': {'status': 'success'},
  'lr-agent': {'status': 'success'},
  'kresman': {'status': 'success'}
}
```

- **create - creates containers, the containers are specified by a docker-compose in agent container (can also be found in `/etc/whalebone/agent`)**

- Parameters: None, Example: `./cli.sh create`

```
{'resolver': {'status': 'success'}}
```

- **updatecache - forces the update of resolver's IoC cache (which is used for blocking), this action should be done to manually force the update and refresh of the domains present in the malicious domain cache**

- Parameters: None

```
{'status': 'success', 'message': 'Cache update successful'}
```

- **containers - lists the containers and their information which include: labels, image, name and status.**

- Parameters: None

```
[
  {
    "id": "b8f4489379",
    "image": {
      "id": "c893b4df5ca3",
      "tags": [
        "whalebone/agent:1.1.1"
      ]
    }
  },
]
```

(continues on next page)

(continued from previous page)

```

    "labels":{
      "lr-agent":"1.1.1"
    },
    "name":"lr-agent",
    "status":"running"
  },
  {
    "id":"e433d58f13",
    "image":{
      "id":"2c4b84a7daee",
      "tags":[
        "whalebone/passivedns:1.1.1"
      ]
    },
    "labels":{
      "passivedns":"1.1.1"
    },
    "name":"passivedns",
    "status":"running"
  },
  {
    "id":"2aeec00121",
    "image":{
      "id":"fc442e625539",
      "tags":[
        "whalebone/kres:1.1.1"
      ]
    },
    "labels":{
      "resolver":"1.1.1"
    },
    "name":"resolver",
    "status":"running"
  },
  {
    "id":"662dac2e6c",
    "image":{
      "id":"b37d0d1bd10b",
      "tags":[
        "whalebone/kresman:1.1.1"
      ]
    },
    "labels":{
      "kresman":"1.1.1"
    },
    "name":"kresman",
    "status":"running"
  },
  {
    "id":"05188ac1df",
    "image":{
      "id":"5b50cdc924fc",

```

(continues on next page)

(continued from previous page)

```

        "tags": [
            "whalebone/logrotate:1.1.1"
        ],
        "labels": {
            "logrotate": "1.1.1"
        },
        "name": "logrotate",
        "status": "running"
    },
    {
        "id": "01e64dd697",
        "image": {
            "id": "fffb52c2dadd",
            "tags": [
                "whalebone/logstream:1.1.1"
            ],
        },
        "labels": {
            "logstream": "1.1.1"
        },
        "name": "logstream",
        "status": "running"
    }
]

```

Each of those actions execute similarly named actions and the status of that action, or output of that action, is printed. The **list** and **run** actions are intended for the scenario when a confirmation of a certain action is required. The action list shows the action that should be executed and the changes that would be done by that action for containers specified in that action. This serves as an example of what would happen if the awaiting action would have been executed. The run action then executes the awaiting action and cleans up afterwards.

The actions of **upgrade** and **create** use the docker-compose template present in the agent container to create/upgrade the desired container. This template is mounted in the volume `/etc/whalebone/agent` if the user decides to change it. However this change needs to be done also to the template present at portal.whalebone.io, if not than the local changes will be overwritten from the cloud during next upgrade.

The bash script should be invoked like this: `./cli.sh action param1 param2 param3`. **Action** is the action name and **parameters** are the action parameters. Only actions for container stop, remove and upgrade use these and specify what containers should be affected by the respective action.

11.2 Strict mode

The agent's default option is to execute actions from the cloud management immediately. It is however possible to enable manual confirmation of requests. This gives the administrator control over when and what gets executed. To enable the resolver Strict mode, please create a ticket to Whalebone support.

To list changes the request introduces the cli option **list** option should be used. To execute the request use cli option **run**. There can only be one request pending in the queue. New request from the cloud will overwrite the previous one, but the new one holds the full desired state anyway. To delete waiting request use cli option **delete_request**. The actions that can be persisted are: **upgrade**, **create** and **suicide**. Please see examples of the CLI command usage.

- **list** - lists the awaiting command and the changes that would be made to the containers specified in the awaiting action, this action is intended for human check hence it's format
 - Parameters: None
 - Example: `./cli.sh list`

```

-----
Changes for resolver
New value for label: resolver-1.1.1

      Old value for label: resolver-1.0.0
-----

```

- **run** - executes the awaiting command
 - Parameters: none
 - Example: `./cli.sh run`

```
{'resolver': {'status': 'success'}}
```

- **delete_request** - deletes the awaiting request
 - Parameters: none
 - Example: `./cli.sh delete_request`

```
Pending configuration request deleted.
```

CLOUD DNS RESOLVERS

You can watch step-by-step video guide [here](#). Whalebone Cloud DNS resolver is a service aimed mainly for small or medium customers who can use cloud resolvers for a backup. Typically it is aimed at ISPs that have only one on-premise resolver and to ensure high availability, they use cloud DNS resolvers as a secondary recursive for their customers. One of the prerequisites is to define a public IP address or ranges to Cloud DNS resolver policy assignment so cloud resolver can differentiate and deliver the right filtering policy which you have setup for your network.

Public network range definition serves to distinguish individual customers and their users. It is necessary to include all the public network ranges that will be used by DNS resolver as well as the users browsing the internet. The definition is used to customize the block page appearance (described later). Single customer can manage more network ranges, such ranges can be assigned to localities to easily distinguish between logical network zones in DNS traffic audit and incidents.

Whalebone service DNS server

193.32.92.32 [Copy](#)

Cloud resolver policies assignment

IP Range	Policy	Options
174.215.4.100/32	Strict policy	Remove IP range

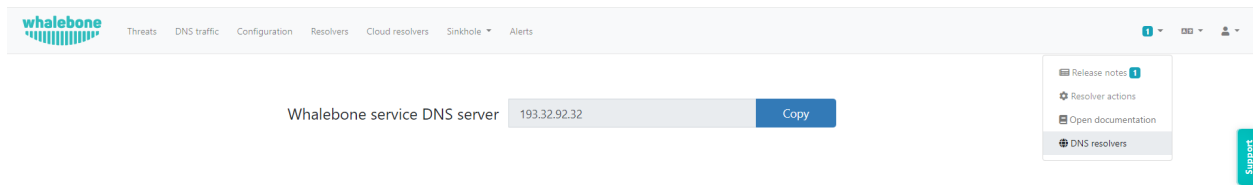
[+ Add IP range](#) [Save to resolver](#)

Warning: If you do not fill in your public network ranges, cloud resolvers will serve as a simple DNS resolvers **without any filtering**. If you use local resolvers, you still have to input your network ranges to display fully customized blocking page to the blocked users.

- Into the field **IP Range** insert one or more network ranges using notation <network address>/<mask>, e.g.: **198.51.100.0/24**.
- Press button **Add IP range** to add more segments of the network.
- Don't forget to save your new setup through the **Save to resolver** button.

Tip: While testing Whalebone (e.g. through adding a testing domain into blacklist) don't forget that many DNS records could be recently in the DNS cache anywhere between the resolver and the user (including the browser, operating system or forwarders). Testing right after the configuration change could therefore fail and the timespan before the protection becomes active could vary based on the TTL of the particular DNS record (should all the caches along the way actually honor the TTL value).

You should forward your DNS traffic towards Whalebone cloud resolvers if this is your preferred deployment option. Cloud resolvers are available on an Anycasted IP address **193.32.92.32** and **193.32.92.33**.



The IP addresses of the resolvers are accessible under **Cloud resolvers** and under the menu **Help → DNS resolvers**.

UNINSTALLING A LOCAL RESOLVER

In order to uninstall a resolver and remove all Whalebone configuration files the following steps should be followed:

Warning: Before starting the process it should be noted that all the individual components that support the resolver functionality are being executed as docker containers. Steps 1 and 2 apply only in case the host server is **dedicated** and **no other services** are running as containers. In case of different situation, please contact us and we will provide an up to date list of the containers that should be removed.

1. Step – Stop and remove all the running docker containers:

```
docker rm -f lr-agent && docker rm -f $(docker ps -q)
```

2. Step – Uninstall Docker:

Please follow the instructions for the applicable operating system:

- [CentOS](#)
- [Red Hat](#)
- [Debian](#)
- [Ubuntu](#)

3. Step – Remove all resolver configuration files and related data:

```
rm -rf /etc/whalebone  
rm -rf /var/whalebone  
rm -rf /var/lib/kres
```

4. Step – Remove DNS traffic and incidents logs:

If you want to fully uninstall the resolver including the logs from DNS traffic and incidents, delete also the log folder. If your intention is just to re-install the resolver but keep the logs, you can skip this step.

DATA ANALYSIS

Whalebone Portal (graphical user interface) gives the user number of possibilities how to analyze what is happening on the DNS resolvers and the network.

14.1 Threats

Threats are special events where there is a DNS request for a domain that is present within the reputation database. There are two types of actions when a threat is detected. The first is to **audit** the event while the second is to **block** it. **Audit** option only logs the domain but access is possible.

The action that is to be implemented depends on the policies that are assigned to the specific resolver. For more on that please refer to [Security Policies](#).

There are some pre-configured filters that can be applied on the data on the portal. Some sample queries can be found below. These queries depict the majority of the use cases but there is no hard limit as the available search engine is **full-text** and *any* query can be compiled impromptu.

You can watch step-by-step video guide [here](#).

14.1.1 How to search for audit/block events:

There are two options for filtering different types of events. The first option, a visual filter can be used. Within the graph, you can click one of the actions (audit, block, allow) to filter it and display only the cases in which the event occurred. Second one is to click next to the **Result's filter** field on the **Filter button** and choose desired filtering option.

14.1.2 How to search for a domain:

The easiest way to search for a domain is by clicking on a specific domain in the log history. The second way is by typing the domain name into the **Result Filter** field.

14.1.3 How to search for events based on specific IP address:

Filtering logs from a specific IP address is possible by selecting a specific source IP address in the log history. The second option is by entering the domain name in the **Result Filter** field.

14.1.4 How to search for events based on specific threat category:

There is a large number of threat categories.

Some of them are: *malware*, *c&c*, *blacklist*, *phishing*, *coinminer*, *spam*, and *compromised*.

A simple way to find attacks is by selecting a specific category from the pie charts or in the log list under the **Threat Categories** column. Another option is to click the **Filter result** button next to the **Filter** field and select the desired filtering option.

14.1.5 How to change the date range of the available data:

The range of data that can be displayed in the portal preview can be changed in several ways. The basic selection method includes choosing predefined time windows (1, 7, 14 or 30 days) in the drop-down list next to the **results filter**. If necessary, a specific time range can be specified using the **Start Date and Time** and **End Date and Time** windows.

14.1.6 How to analyze a domain:

In case to know further information about domain, especially what score Whalebone assigns to particular domain, when was first seen and categorized as malicious, if it falls under regulatory category or what external sources know about it, then watch step-by-step video [here](#).

14.2 DNS Traffic

The DNS Traffic tab contains an overview of the traffic that has been logged on the resolver. It contains all the queries along with some additional information such as the type, the answer and the TTL (time to live) of the answer.

Tip: The data are subject to de-duplication. This means that the resolver logs only unique combinations of query, query type and answer per 24 hour time frame. For this reason, a query might not be available on the portal even though it has been resolved.

You can watch step-by-step video guide [here](#).

Below are some of the most useful filtering options of the available data will be described.

14.2.1 How to view all queries of a specific type:

The easiest way to select queries of a certain type is by clicking the **filter** icon and selecting the desired query type. There are several options to choose from, including A, AAA, CNAME, MX, NS, PTR, RRSIG, SPF, SRV and TXT.

14.2.2 How to view all answers of a specific type:

In the **Answers** window, you can select the desired answer, or in the log list in the **Answer** column, or click the desired answer.

14.2.3 How to search for a domain:

To search for domains, you can use the **Result Filter** text box to enter the name of the domain you are looking for. Other ways to search for a domain is by clicking the domain in the **Tier 2 Domains** section or directly in the log list in the same column.

14.2.4 How to change the date range of the available data:

The range of data that can be displayed in the portal preview can be changed in several ways. The basic selection method includes selecting predefined time windows (1, 7, 14 or 30 days) from the drop-down list located next to the **results filter**. If desired, a specific time range can be specified using the **Start Date and Time** and **End Date and Time** windows.

14.2.5 How to view DGA (Domain Generation Algorithm) indications:

DGA indications can be filtered in a similar way as in the case of displaying queries of a certain type, in this case just select the last record in the list - **DGA**

14.2.6 Fulltext filtering

For more advanced use, you can use the full-text filter and build a compound query. Fulltext filtering only works in the **Threats** panel. .. warning:

The ****content**** and ****DNS traffic**** dashboards does **not** support fulltext filtering at the moment. Only the clickable elements will result in filtering the data in the content dashboard.

These fields can be concatenated using logical operators. AND, OR, NOT, <, > and the wildcard character * are supported. Strings do not have to be wrapped with quotes. An example of the syntax is as follows: action: block AND accu:>70 AND (client_ip: 10.20.30.41 OR 10.20.30.40 OR 192.168.*) AND NOT geoip.country_name: Germany AND matched_iocs.classification.type: malware AND NOT phishing When you run a fulltext query, it updates the content of the entire dashboard.

Threats	Description	Example value
timestamp	The exact time when the resolver registered the DNS request / incident	2022-10-14T12:28:01.000Z
client_ip	The source IP address which made the DNS request / incident	192.168.2.3
domain	The domain in the DNS query	whalebone.io OR whale*one.io
resolver_id	The id of the resolver which handled the event	2404
device_id	The device_id of the HOS agent	MB2A1b40TDin3Xz6DgftAip72v57e
geoup. continent_code	The code of the continent from the php geoIP library	AF AN AS EU NA OC SA
geoup. country_code3	The code of the country from the php geoIP library	RU CZ US CN DE ...
geoup. country_name	The name of the country from the php geoIP library	Russia
ip	The IP in the DNS answer or the IP that would the resolver answer if it didn't block	174.85.249.36 OR SERVFAIL OR NXDOMAIN
action	The action that the resolver took with that specific query	block allow audit
accu	The score of the domain at the time of the event	0..100 < and > operators can be used too
matched_iocs. classification. type	The type of threat	malware c&c phishing coinminer spam compromised blacklist

Tip: Filtering operators are placed statically to the URL address. Therefore, you can create your set of Filters in advance (such as view on individual IPs) and to use them when necessary. Afterwards, you can place them to your CRM for the specific user's account and to access the filtered view immediately. It will help saving your time when customer asks for the support as you can immediately open their details.

DOMAIN RESOLUTION ANALYSIS

There is always chance that every administrator will encounter a situation, when DNS resolution is not successful. Most of the time it is not related to Whalebone's resolver but there is probably an issue with an authoritative server.

ISPs often face complaints that users cannot access the domain, in many cases it is not the ISP's fault. Whalebone solution provides you with information so you can identify the issue.

Steps to be done:

Step 1. – Examine domain in the **Threats** page.

- Check whether domain was blocked by a security feature.

Step 2. – Examine domain in the **DNS traffic**.

- If it was not blocked because of **threats**, go to **DNS Traffic** and check whether it reached the resolver.
- Users often rewrites resolver with public ones and if that resolver faces a issue ISP is blamed to as source of problem, which is not true

You can face three cases:

- Domain was translated correctly.
- NXDOMAIN was returned - it means that the authoritative server responded, but the domain or subdomain does not exist.
- SERVFAIL - no response came from the configured authoritative server. This can mean an outage of server or link issue.

Step 3. – Examine domain using **DNSVIZ** tool.

- Under each domain there is an arrow where you can be redirected to DNSVIZ of a particular domain.
- It shows full resolution process in a human readable way.
- It can show that the DNSSEC validation process was unsuccessful or the authoritative DNS server was not reachable.

You can watch step-by-step video guide [here](#).

Whalebone administration portal provides ability to trace the domain. This feature is available in **Resolvers** under each resolver's three dots. This feature shows what information is passed to resolver when resolving particular domain.

You can watch step-by-step video guide [here](#).

REPORTS

Reporting capabilities can be configured from the drop-down menu under a user's account. The properties that can be customized, include the frequency that the reports are being delivered, the preferred day of the week, the language and the recipients.

Note: The default recipient is the owner of the account and the reports are delivered to their respective registered email address.

ALERTS

Whalebone alerting provides live updates about key information such as resolver's health, resolution status, hardware usage and it also informs about crucial security incidents and many more. All of these information can be passed through multiple channels e.g. email, slack, syslog or webhook. You can create new alert from predefined templates and alerts can be then customized by editing their parameters. You can watch step-by-step video guide [here](#)

Note: In order to turn on an alert, you first need to assign a destination for it. Click the alert name to expand it in detail and select the destination from the box. Multiple destinations may be selected by shift-clicking the addresses.

Note: When the alerting channel is syslog, by default TCP or TLS is supported as the transport layer protocol.

Note: The Syslog or Webhook alerts are sent from the following IP addresses: 159.100.247.142 and 159.100.247.58. If you select one of these channels, make sure to make an incoming TCP traffic exception on your firewall to be able to receive the message.

17.1 DNS traffic - count of unique requests from IP

This alert is triggered when a single source IP reaches the limit of unique requests with defined attributes. Parameters:

- **MINUTES:** time window in minutes (Default=15)
- **TRESHOLD:** number of events in timeframe to trigger alert (Default=100)
- **QUERY_TYPE:** Filter by DNS query type (Default=*)
- **RESPONSE_TYPE:** Filter by DNS response (Default=*)
- **IP_WILDCARD:** Include only these comma-separated IP addresses in the alert (Default=*)
- **IP_WILDCARD_IGNORE:** Ignore these comma separated domains in the alert (Default=none)
- **DOMAIN_WILDCARD:** Include only these comma separated domains in the alert(Default=*)
- **DOMAIN_WILDCARD_IGNORE:** Ignore these comma separated domains in the alert (Default=none)
- **DGA:** Filter by domain generation algorithm - Only DGA, Without DGA or both (Default=*)

17.2 DNS traffic - increased percentage of queries

This Alert will be sent when number of DNS traffic logs is percentually greater over configured time period. Parameters:

- **MINUTES:** Timeframe - time window (Default=15)
- **PERCENT:** Percentage increase (e.g. 200%) - difference between two intervals (Default=50)
- **QUERY_TYPE:** Filter by DNS query type (Default=*)
- **RESPONSE_TYPE:** Filter by DNS response (Default=*)

17.3 DNS traffic - possible homograph attack

This Alert is sent when a possible homograph attack for a specified domain is detected Parameters:

- **DOMAIN:** Domain to monitor for possible homograph attacks (One alert can monitor only one domain)
- **DISTANCE:** Number of characters that can differ in the phishing domain (Default=1)
- **DOMAIN_WILDCARD_IGNORE:** Ignore this list of comma separated domains in the alert. In case DISTANCE is larger than 1, there will be a detection on domains that support both a global and regional top level formats. It is recommended to add the legitimate domains in the whitelists in order to avoid unnecessary alarms. (Default=none)

17.4 DNS traffic - treshold for unique queries

This Alert is sent when threshold for filtered unique DNS logs is reached Parameters:

- **MINUTES:** Timeframe - time window (Default=15)
- **TRESHOLD:** Threshold - number of events in timeframe to trigger alert (Default=100)
- **QUERY_TYPE:** Filter by DNS query type (Default=*)
- **RESPONSE_TYPE:** Filter by DNS response (Default=*)

17.5 Resolver - Cloud communication failure

This Alert is sent when the backend does not receive any message from the local resolver agent for more than 20 minutes.

17.6 Resolver - Insufficient hardware resources

This Alert is sent when the local resolver agent detects that the hardware utilization has increased over the defined treshold. The parameters are expressed as percentages of utilized compared to total resources. As an example, if you want to be alerted when the host uses 80 % of total disk space, set the THRESHOLD_HDD to 80.

- **THRESHOLD_CPU:** Utilization of CPU (Default=80)
- **THRESHOLD_MEMORY:** Utilization of RAM (Default=90)
- **THRESHOLD_HDD:** Utilization of HDD (Default=80)

17.7 Resolver - Resolution service failure

The resolver periodically performs checks to test resolution of well-known domains. Google.com, facebook.com, microsoft.com and apple.com are checked every minute. The default setting of the parameters is very strict, so even if resolution of one of the four domains during a 10 minute time interval fails, the alert is sent. Parameters:

- **TRESHOLD:** number of events to occur during the timeframe to trigger the alert (Default=1)
- **MINUTES:** timeframe in minutes (Default=10)

17.8 Threats - count during intervals

This alert is sent if the percentage of threat records is higher than the set time period. Parameters:

- **MINUTES:** time window in minutes (default=15)
- **TRESHOLD:** number of events in the time window for triggering the alert, this is a percentage change (default=100).
- **LOG_TYPE:** (default=*): filters by event type (audit/block)

17.9 Threats - event detection

This alert is sent in the case of a new entry in the threats page according to the specified threat type and action performed. Parameters:

- **LOG_TYPE:** (Default=*): filters by action type (audit/block)
- **THREAT_TYPE:** (Default=*): filters by type of threat detected

17.10 Threats - newly blocked domain

This alert is sent if the resolver detects a newly blocked threat within the specified time frame. Parameters:

- **DAYS:** Number of days on which newly blocked domains will be searched (default=30)
- **DOMAIN_WILDCARD:** Include only the following comma-separated domains in the alert(Default=*)

API INTEGRATION

Whalebone API is a practical way to access all the data that are gathered by Whalebone's resolvers and integrate them to external systems. The API documentation has two separate schemas. One for getting the events from Whalebone and one for getting and configuring the settings.

- If you want to retrieve incidents, DNS traffic data and resolver metrics use [this schema](#).
- If you want to configure the resolver, update policies, add domains to allow/deny lists or get the settings, use [this schema](#).

In order to authenticate to the API, every user needs a set of **Access Key** and **Secret Key**. These can be managed from the option **API keys** on the dropdown menu, under the user's account.

You can watch step-by-step video guide [here](#).

- **API Key Generation**

The generation of the API key can be achieved by clicking the **Generate new key** button.

Note: Make sure to copy the *Key secret* as it cannot be retrieved again.

- **API Key Revocation**

In case an API key gets lost or compromised, its revocation can be achieved by the same menu by clicking the red trash bin icon. Every key is tightly-coupled with the user ID and there is no central key management. In order to invalidate a key that you do not have access to, the respective user needs to delete the key himself or the entire user must be deleted.

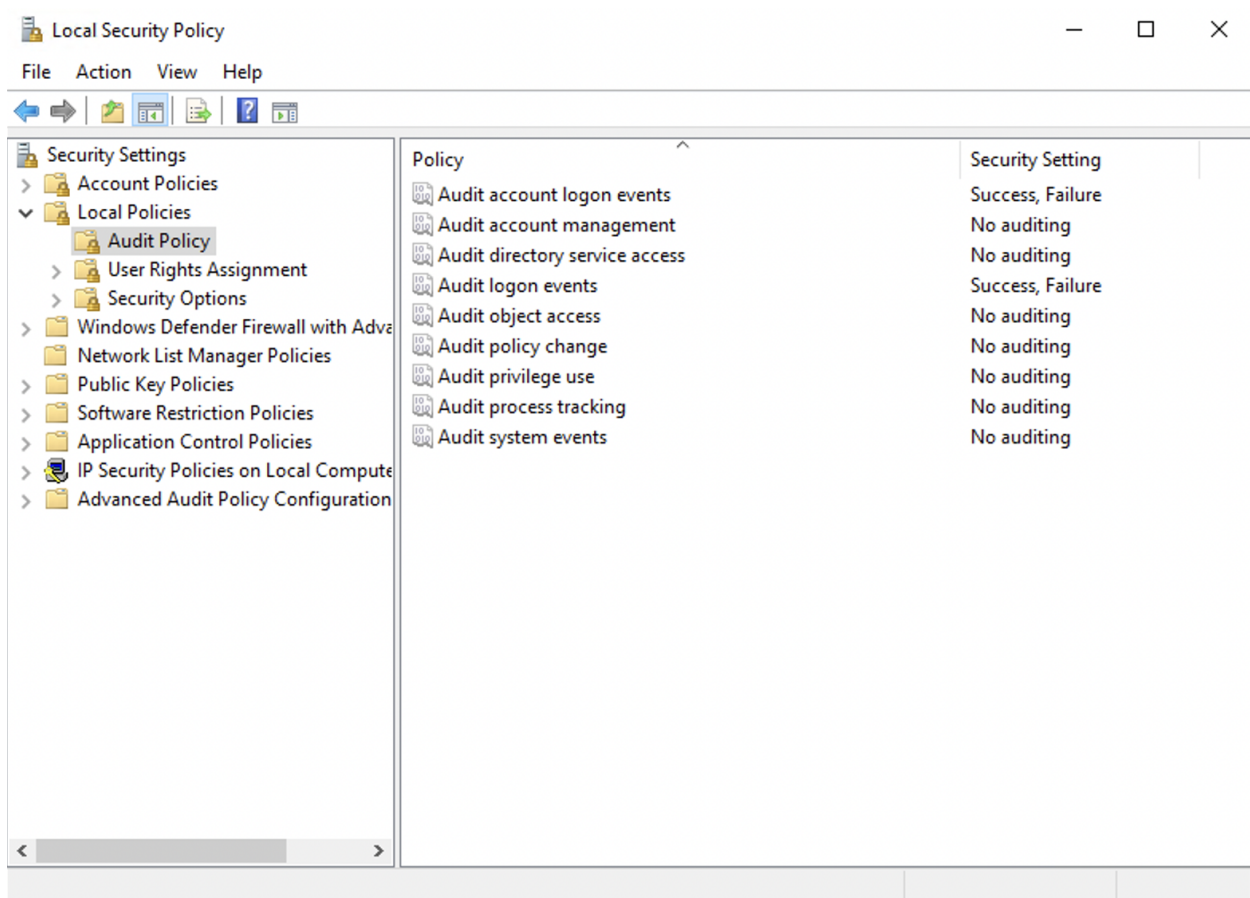
ACTIVE DIRECTORY INTEGRATION

19.1 Installation prerequisites

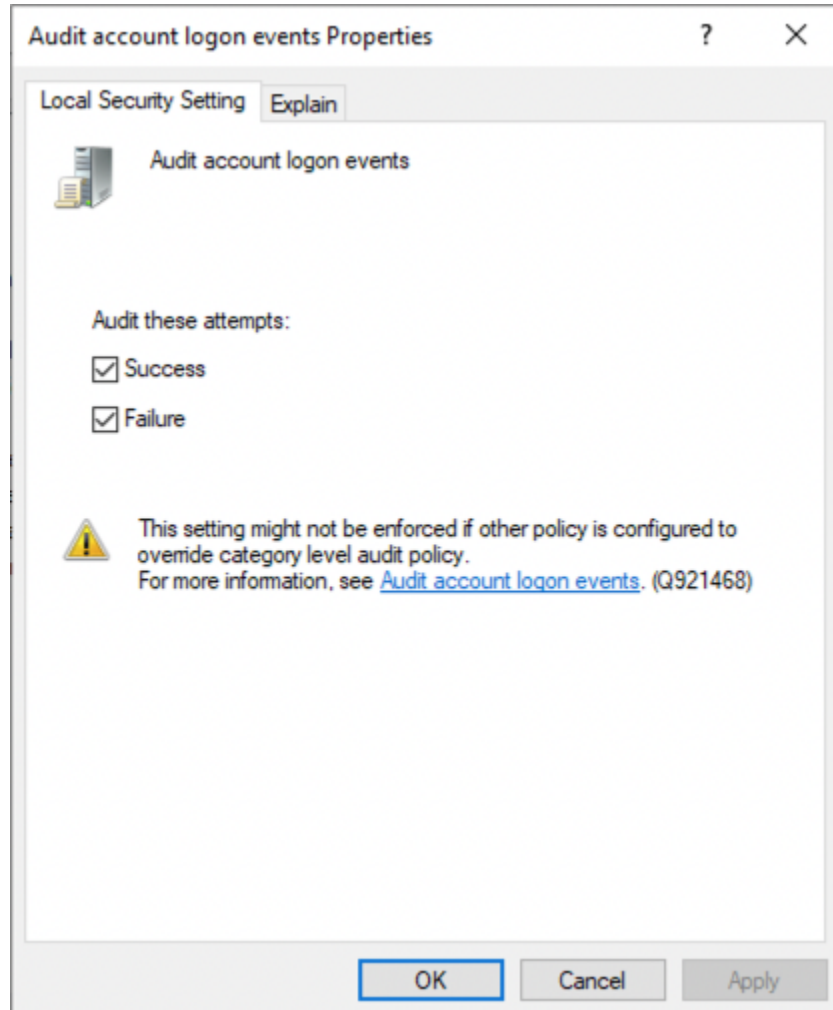
Before you install and Event Log Forwarder (ELF) on one or more of your devices, please ensure that you have enabled audit of events.

On each of your Domain Controllers (DC) go to: Windows Administrative Tools → Local Security Policy, and then Security Settings → Local Policies → Audit Policy, and there find Audit account logon events, Audit account sign-in events and Audit logon events.

Some settings may differ in name or be missing, based on your Windows version.



Check both Success and Failure boxes.



You may need to reload configured policy. To reload policy, please run following command:

```
gpupdate /force
```

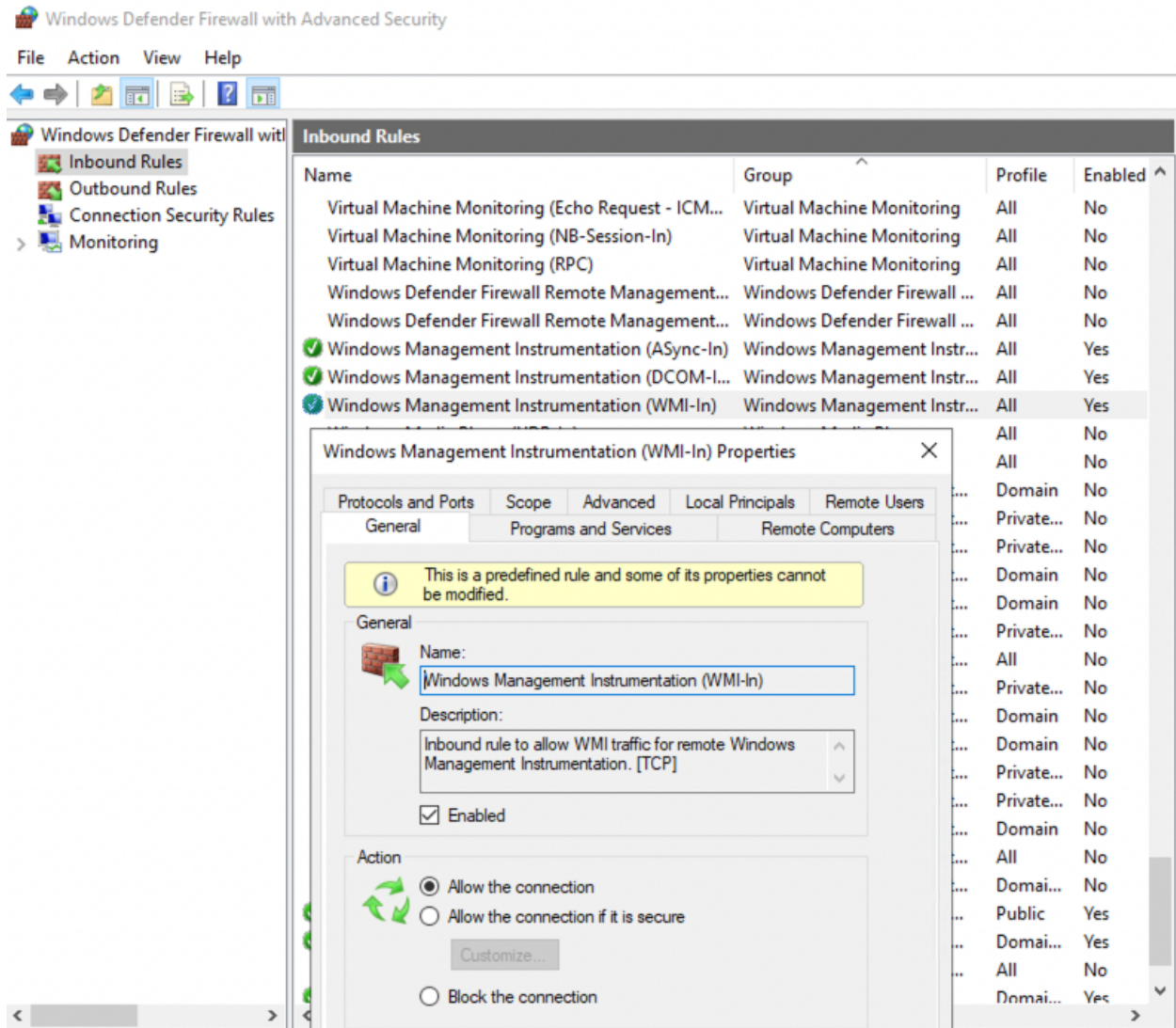
19.2 Domain Controller Configuration

19.2.1 DC Firewall on Windows

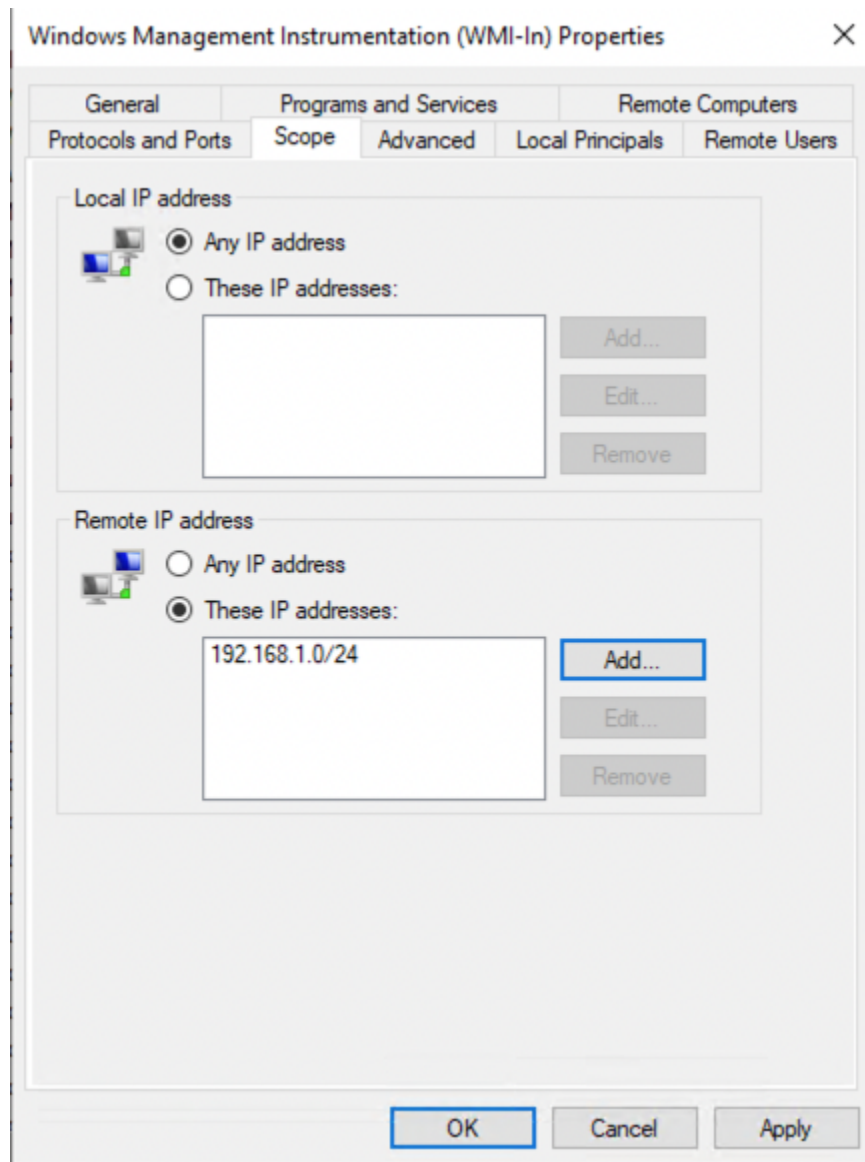
Ensure that Event Log can be accessed through your Firewall configuration using WMI.

On each of your Domain Controllers go to: Windows Defender Firewall → Windows Defender Firewall with Advanced Security on Local Computer Inbound Rules → Windows Management Instrumentation (WMI-In)

ensure the rule allows connections



set up a scope of allowed addresses that may connect. In this example a remote address 192.168.1.0/24 is allowed.



Or, alternatively you can use command line:

```
netsh firewall set service RemoteAdmin enable
```

19.2.2 DC Firewall Rules

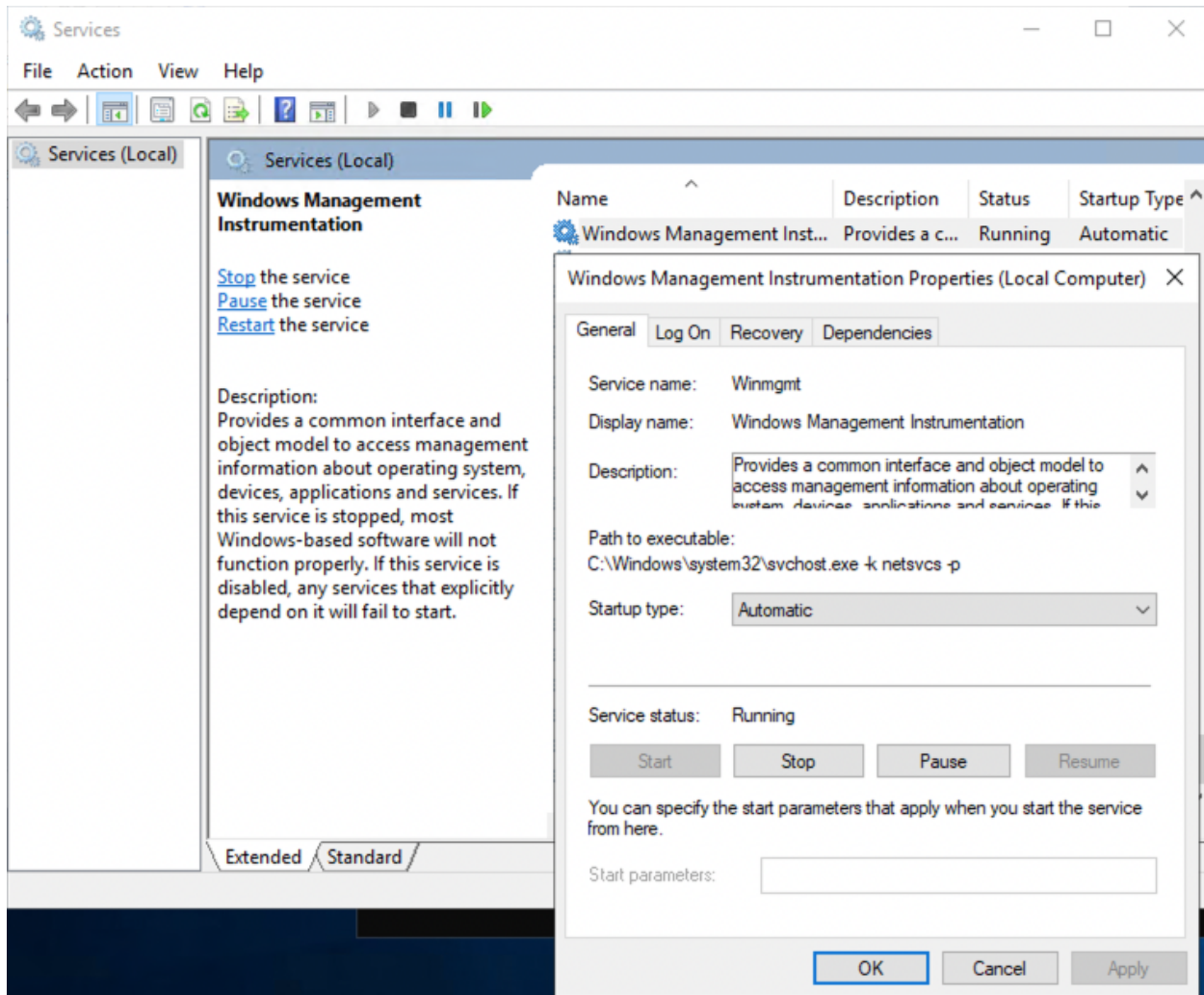
Source	Direction	Destination	Port	Protocol	Reason
DC	—>	local netwk	135	TCP/UDP	Microsoft RPC
DC	—>	local netwk	445	TCP	Microsoft MQ
DC	—>	local netwk		ICMP	

19.2.3 Windows Service

Please ensure that Windows Management Instrumentation service is running.

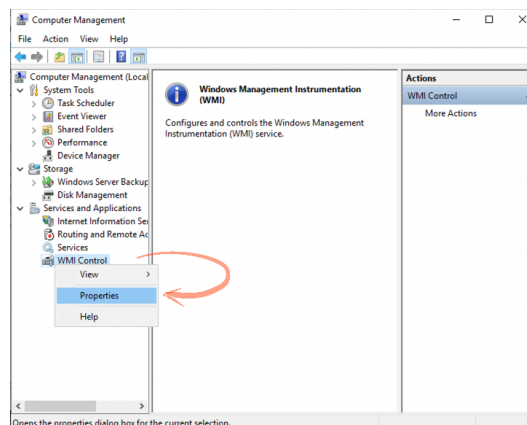
```
C:\Users\Administrator>sc query Winmgmt
```

```
SERVICE_NAME: Winmgmt
        TYPE               : 30  WIN32
        STATE                : 4  RUNNING
                           (STOPPABLE, PAUSABLE, ACCEPTS_SHUTDOWN)
        WIN32_EXIT_CODE       : 0  (0x0)
        SERVICE_EXIT_CODE    : 0  (0x0)
        CHECKPOINT           : 0x0
        WAIT_HINT            : 0x0
```

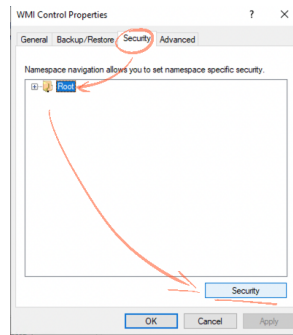


19.2.4 WMI Remote Configuration

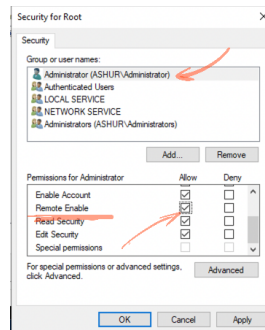
If you chose to install ELF on another Windows PC, ensure that it can use WMI remotely. To enable Remote WMI for the account which will be used to connect to Domain Controller, go to: Computer Management → Services and Applications → WMI Control Right click on it and select Properties



Select Security tab, then choose the Root namespace and hit Security button.



Add user to the list or select a group it belongs to, check Remote Enable permission.



19.3 Event Log Forwarder

You can install ELF locally on the DC or on another Windows PC. ELF uses following connections:

19.3.1 ELF Firewall Rules

Source	Direction	Destination	Port	Protocol	Reason
ELF	—>	DC	135	TCP/UDP	
ELF	—>	resolver	4222	TCP	NATS Message Queue

19.3.2 Install Instructions

Install or Update:

```
msiexec /i "Whalebone.Event.Log.Forwarder.Installer.msi" ui="true"
```

Uninstall:

```
msiexec /x "Whalebone.Event.Log.Forwarder.Installer.msi"
```

19.3.3 Configuration Instructions

Installer shall open configuration Window automatically. You may access configuration from favourite web browser using command:

```
start http://localhost:55225/Configure/AD
```

The screenshot shows the 'Whalebone ELF Configurator' web interface. On the left is a dark sidebar with a menu containing 'Active Directory' (selected), 'NATS', 'Support', 'Charts', and 'Status'. The main content area displays four steps for configuration:

- Step 1:** Enter your Active Directory username and password for Windows Event Log. Username cannot contain „@„ character or domain prefix. Thanks to the verification of your identity, we can then connect to the Domain controller. Event Log is not accessible by default, check support documentation to learn how to allow in on Windows Firewall easily.
- Step 2:** Enter the hostname or IP address of your Domain Controller or another Windows PC that serves Event Log replica.
- Step 3:** By clicking on „Test“ we will make a test connection to Active directory and make a test download of messages.
- Step 4:** By clicking on „Save“ the credentials will be encrypted and saved in the config. Polling will start.

Below the steps are input fields for configuration:

- Username:
- Password:
- Domain:
- Server (hostname or IP address):

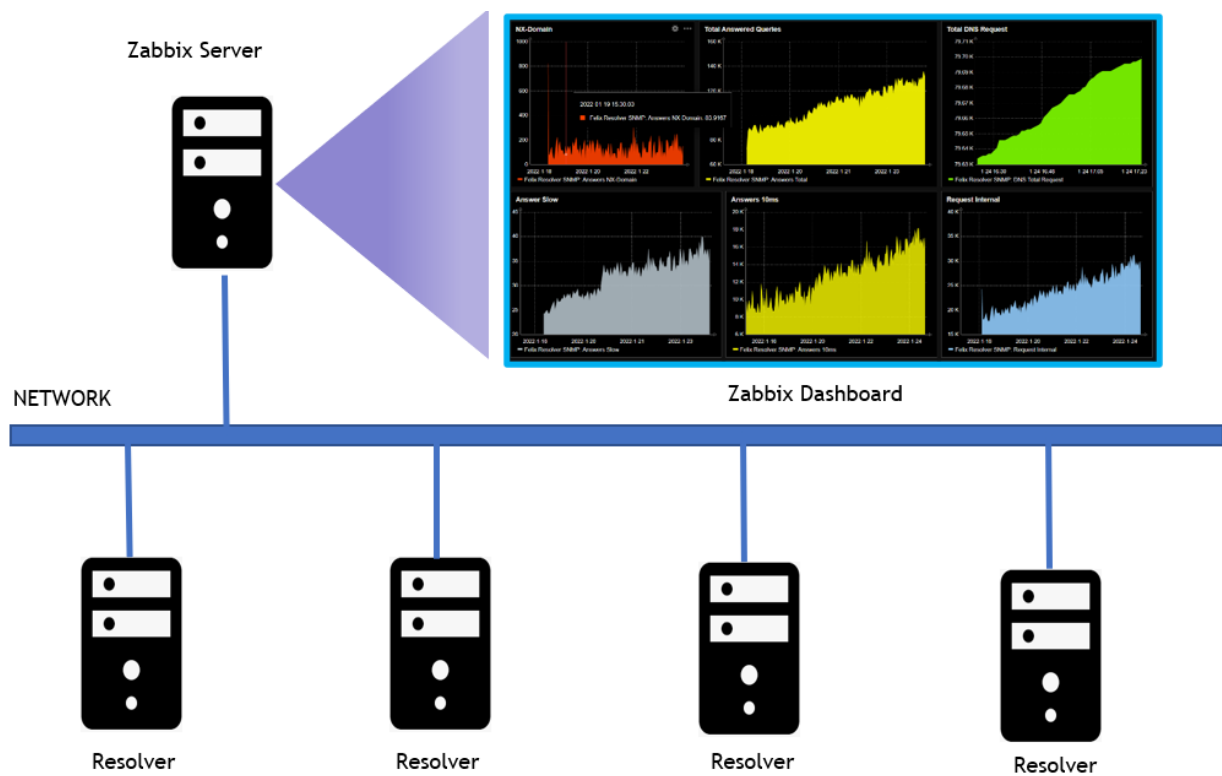
At the bottom are two buttons: 'Test' (blue) and 'Save' (orange).

19.3.4 Service Logs

Service logs can be found at `c:\ProgramData\Whalebone\Event Log Forwarder\`, which contain detailed information about service state. In case you encounter unexpected service behaviour please include this folder along inside your support ticket.

SNMP MONITORING

20.1 High Level Network Diagram



Whalebone SNMP agent is enabled in the resolvers to actively monitor the local resources, queries and statistics.

20.2 SNMP OID

SNMP OID stands for Object Identifiers for creating an SNMP Template for Network Monitoring tool. Below table is the Whalebone SNMP OID's.

Property	ID	SNMP OID
Hostname	1	.1.3.6.1.4.1.8072.1.3.2.4.1.2.9.119.104.97.108.101.98.111.110.101.1
Check Port	2	.1.3.6.1.4.1.8072.1.3.2.4.1.2.9.119.104.97.108.101.98.111.110.101.2
Check Resolve	4	.1.3.6.1.4.1.8072.1.3.2.4.1.2.9.119.104.97.108.101.98.111.110.101.3
CPU Count	6	.1.3.6.1.4.1.8072.1.3.2.4.1.2.9.119.104.97.108.101.98.111.110.101.4
Memory Available	7	.1.3.6.1.4.1.8072.1.3.2.4.1.2.9.119.104.97.108.101.98.111.110.101.6
Memory Total	8	.1.3.6.1.4.1.8072.1.3.2.4.1.2.9.119.104.97.108.101.98.111.110.101.7
Memory Usage	9	.1.3.6.1.4.1.8072.1.3.2.4.1.2.9.119.104.97.108.101.98.111.110.101.8
HDD Free	10	.1.3.6.1.4.1.8072.1.3.2.4.1.2.9.119.104.97.108.101.98.111.110.101.9
HDD Total	11	.1.3.6.1.4.1.8072.1.3.2.4.1.2.9.119.104.97.108.101.98.111.110.101.10
HDD Usage	12	.1.3.6.1.4.1.8072.1.3.2.4.1.2.9.119.104.97.108.101.98.111.110.101.11
Swap Free	13	.1.3.6.1.4.1.8072.1.3.2.4.1.2.9.119.104.97.108.101.98.111.110.101.12
Swap Total	14	.1.3.6.1.4.1.8072.1.3.2.4.1.2.9.119.104.97.108.101.98.111.110.101.13
Swap Usage	15	.1.3.6.1.4.1.8072.1.3.2.4.1.2.9.119.104.97.108.101.98.111.110.101.14
Timestamp	16	.1.3.6.1.4.1.8072.1.3.2.4.1.2.9.119.104.97.108.101.98.111.110.101.15
Requests Total	17	.1.3.6.1.4.1.8072.1.3.2.4.1.2.9.119.104.97.108.101.98.111.110.101.16
Requests Internal	18	.1.3.6.1.4.1.8072.1.3.2.4.1.2.9.119.104.97.108.101.98.111.110.101.17
Requests UDP	19	.1.3.6.1.4.1.8072.1.3.2.4.1.2.9.119.104.97.108.101.98.111.110.101.18
Requests TCP	20	.1.3.6.1.4.1.8072.1.3.2.4.1.2.9.119.104.97.108.101.98.111.110.101.19
Requests DoT	21	.1.3.6.1.4.1.8072.1.3.2.4.1.2.9.119.104.97.108.101.98.111.110.101.20
Requests DoH	22	.1.3.6.1.4.1.8072.1.3.2.4.1.2.9.119.104.97.108.101.98.111.110.101.21
Requests XDP	23	.1.3.6.1.4.1.8072.1.3.2.4.1.2.9.119.104.97.108.101.98.111.110.101.22
Answers Total	24	.1.3.6.1.4.1.8072.1.3.2.4.1.2.9.119.104.97.108.101.98.111.110.101.23
Answers cached	25	.1.3.6.1.4.1.8072.1.3.2.4.1.2.9.119.104.97.108.101.98.111.110.101.24
Answers No error	26	.1.3.6.1.4.1.8072.1.3.2.4.1.2.9.119.104.97.108.101.98.111.110.101.25
Answers No data	27	.1.3.6.1.4.1.8072.1.3.2.4.1.2.9.119.104.97.108.101.98.111.110.101.26
Answers NX-Domain	28	.1.3.6.1.4.1.8072.1.3.2.4.1.2.9.119.104.97.108.101.98.111.110.101.27
Answers SERVFAIL	29	.1.3.6.1.4.1.8072.1.3.2.4.1.2.9.119.104.97.108.101.98.111.110.101.28
Answers 1ms	30	.1.3.6.1.4.1.8072.1.3.2.4.1.2.9.119.104.97.108.101.98.111.110.101.29
Answers 10ms	31	.1.3.6.1.4.1.8072.1.3.2.4.1.2.9.119.104.97.108.101.98.111.110.101.30
Answers 50ms	32	.1.3.6.1.4.1.8072.1.3.2.4.1.2.9.119.104.97.108.101.98.111.110.101.31
Answers 100ms	33	.1.3.6.1.4.1.8072.1.3.2.4.1.2.9.119.104.97.108.101.98.111.110.101.32
Answers 250ms	34	.1.3.6.1.4.1.8072.1.3.2.4.1.2.9.119.104.97.108.101.98.111.110.101.33
Answers 500ms	35	.1.3.6.1.4.1.8072.1.3.2.4.1.2.9.119.104.97.108.101.98.111.110.101.34
Answers 1000ms	36	.1.3.6.1.4.1.8072.1.3.2.4.1.2.9.119.104.97.108.101.98.111.110.101.35
Answers 1500ms	37	.1.3.6.1.4.1.8072.1.3.2.4.1.2.9.119.104.97.108.101.98.111.110.101.36
Answers slow	38	.1.3.6.1.4.1.8072.1.3.2.4.1.2.9.119.104.97.108.101.98.111.110.101.37
Answers AA	39	.1.3.6.1.4.1.8072.1.3.2.4.1.2.9.119.104.97.108.101.98.111.110.101.38
Answers TC	39	.1.3.6.1.4.1.8072.1.3.2.4.1.2.9.119.104.97.108.101.98.111.110.101.39
Answers RA	40	.1.3.6.1.4.1.8072.1.3.2.4.1.2.9.119.104.97.108.101.98.111.110.101.40
Answers RD	41	.1.3.6.1.4.1.8072.1.3.2.4.1.2.9.119.104.97.108.101.98.111.110.101.41
Answers AD	42	.1.3.6.1.4.1.8072.1.3.2.4.1.2.9.119.104.97.108.101.98.111.110.101.42
Answers CD	43	.1.3.6.1.4.1.8072.1.3.2.4.1.2.9.119.104.97.108.101.98.111.110.101.43
Answers DO	44	.1.3.6.1.4.1.8072.1.3.2.4.1.2.9.119.104.97.108.101.98.111.110.101.44
Answers ENDS0	45	.1.3.6.1.4.1.8072.1.3.2.4.1.2.9.119.104.97.108.101.98.111.110.101.45

continues on next page

Table 1 – continued from previous page

Property	ID	SNMP OID
Queries EDNS	46	.1.3.6.1.4.1.8072.1.3.2.4.1.2.9.119.104.97.108.101.98.111.110.101.46
Queries DNSSEC	47	.1.3.6.1.4.1.8072.1.3.2.4.1.2.9.119.104.97.108.101.98.111.110.101.47
Predict Epoch	48	.1.3.6.1.4.1.8072.1.3.2.4.1.2.9.119.104.97.108.101.98.111.110.101.48
Predict learned	49	.1.3.6.1.4.1.8072.1.3.2.4.1.2.9.119.104.97.108.101.98.111.110.101.49
Predict Queue	50	.1.3.6.1.4.1.8072.1.3.2.4.1.2.9.119.104.97.108.101.98.111.110.101.50

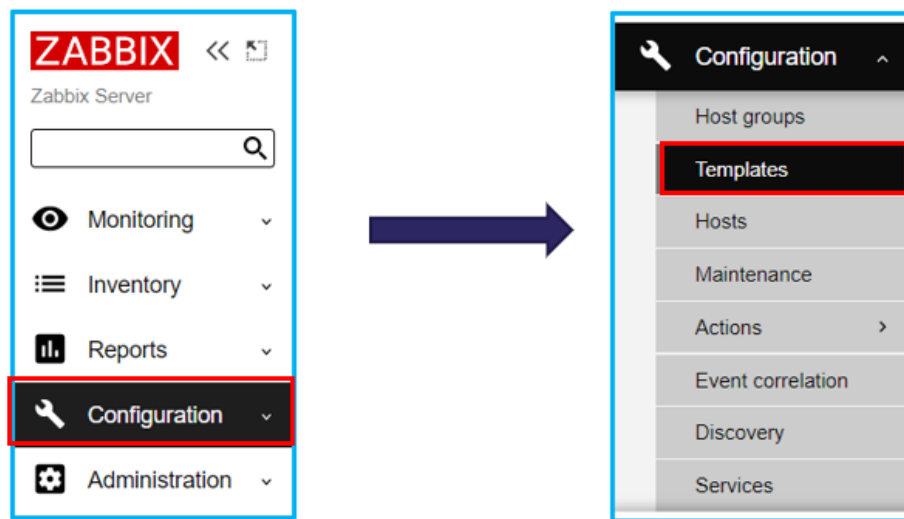
20.2.1 Zabbix Integration

The agent gathers operational information locally and reports data to Zabbix server for processing. Moreover, Zabbix offers excellent reporting and data visualization features based on the stored data from the resolver.

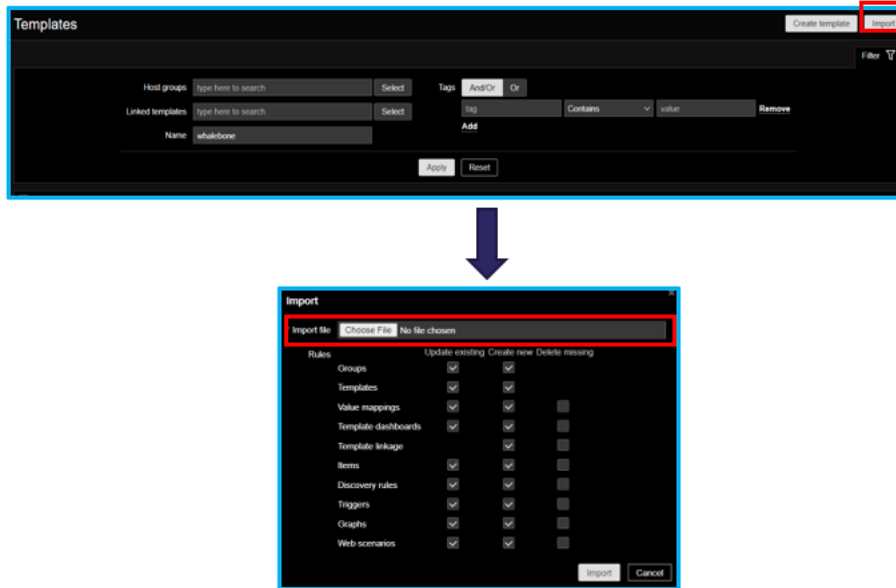
Zabbix is a monitoring tool that provides performance metrics such as network utilization, CPU and memory utilization. It also monitors network disconnection and server unavailability.

20.3 How to import the Whalebone Template

- To import the Whalebone template, go to Zabbix **Configuration**. Under the configuration go to **templates**.



- On the **Templates** tab, select **Import** and select the template file.



20.4 How to add the resolver in Zabbix Monitoring

- To add the host, go to Configuration then click **hosts**. Click **create host** then provide the hostname, groups. After that add the resolver ip address.
- Under the interface select the **SNMP** → Provide the **SNMP Ip address** → Port **161** → SNMP version **SNMPv2** then add the **SNMP Community**.
- After adding the host go to **templates** tab → Select the whalebone template. Click **select** and **add**.
- After selecting the Whalebone template go back to **host** and click **add**. On the hosts tab we can see that the resolver has been added on the Zabbix.

Note: SNMP data from the resolver to Zabbix will take time to initialized. Wait the Zabbix to gather data from the server. Always observe the availability on the right corner to see if it's green. Green means its already connected to the whalebone resolver.

This screenshot shows the top right corner of the Whalebone admin interface. A red box highlights the 'Create host' button. Other visible elements include an 'Import' button, a 'Filter' dropdown, and a 'Monitored by' section with tabs for 'Any', 'Server', and 'Proxy'. Below these are input fields for 'Proxy', 'Tags' (with 'And/Or' and 'Or' options), and a 'Contains' filter with a 'value' input and a 'Remove' button. At the bottom are 'Apply' and 'Reset' buttons.



This screenshot shows the 'Hosts' configuration page. A red box highlights the 'Host name' input field. Another red box highlights the 'Groups' dropdown menu, which has a 'Select' button. Below the groups is an 'Interfaces' section with the text 'No interfaces are defined' and a red box around the 'Add' button. The 'Description' field is a large text area. At the bottom, there is a 'Monitored by proxy' dropdown (set to 'no proxy'), an 'Enabled' checkbox (checked), and 'Add' and 'Cancel' buttons.

This screenshot shows the 'Interfaces' configuration page. A table lists the interfaces with columns: 'Type', 'IP address', 'DNS name', 'Connect to', 'Port', and 'Default'. The first row is for 'SNMP' with IP address '127.0.0.1', 'IP' as the connection type, and '161' as the port. A red box highlights the 'IP' and '161' values. Below the table, there is a red box around the 'SNMP version' dropdown (set to 'SNMPv2') and another red box around the 'SNMP community' input field (containing '{\$SNMP_COMMUNITY}'). At the bottom is a checked checkbox for 'Use bulk requests'.

Interfaces	Type	IP address	DNS name	Connect to	Port	Default
SNMP	127.0.0.1		IP	DNS	161	Remove

Hosts

Host **Templates** IPMI Tags Macros Inventory Encryption Value mapping

Linked templates: Name Action

Link new templates: type here to search Select

Add Cancel



Templates

Host group: Templates Select

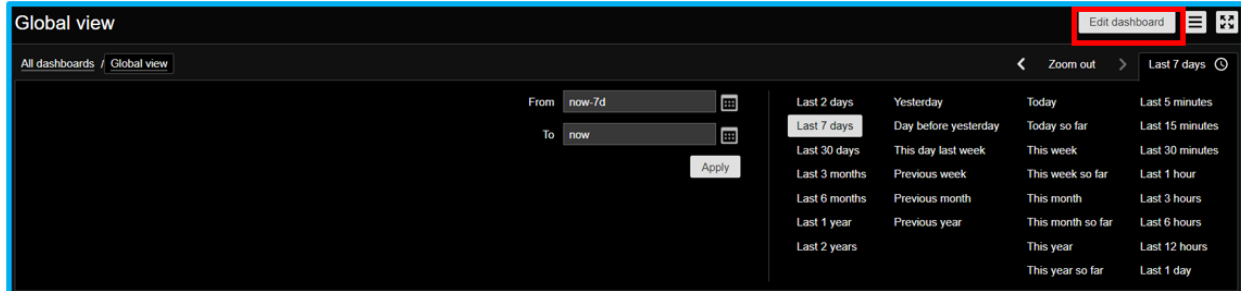
- ☐ VMware Guest
- ☐ VMware Hypervisor
- ☐ VMware macros
- ☐ VMware SD-WAN WebCloud by HTTP
- ☐ Windows configure by Zabbix agent 2
- ☒ **Whalebone Resolver**
- ☐ Wildfly Domain by JMX
- ☐ Wildfly Server by JMX
- ☐ Windows by Zabbix agent
- ☐ Windows by Zabbix agent active
- ☐ Windows CPU by Zabbix agent
- ☐ Windows CPU by Zabbix agent active
- ☐ Windows filesystems by Zabbix agent
- ☐ Windows filesystems by Zabbix agent active
- ☐ Windows generic by Zabbix agent
- ☐ Windows generic by Zabbix agent active
- ☐ Windows memory by Zabbix agent
- ☐ Windows memory by Zabbix agent active

Select Cancel

Name	Items	Triggers	Graphs	Discovery	Web	Interface	Proxy	Templates	Status	Availability	Agent encryption
Felix Resolver SNMP	Items 07	Triggers 15	Graphs 53	Discovery 5	Web	78.47.79.134.161		Whalebone Resolver	Enabled	SNMP	None
Zabbix server	Items 124	Triggers 64	Graphs 26	Discovery 3	Web	127.0.0.1:10050		Linux by Zabbix agent (Linux block devices by Zabbix agent, Linux CPU by Zabbix agent, Linux filesystems by Zabbix agent, Linux generic by Zabbix agent, Linux memory by Zabbix agent, Linux network interfaces by Zabbix agent, Zabbix agent), Zabbix server health	Enabled	78X	None

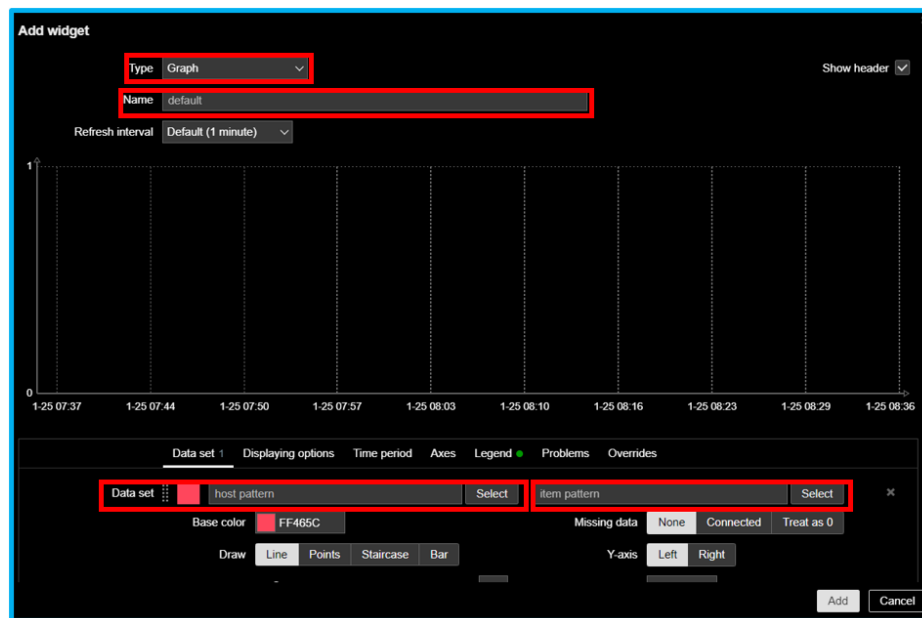
20.5 How to add the Whalebone widget on Zabbix dashboard

- To add the dashboard, go to **Monitoring** then **Dashboard**. On the dashboard Global view, we can see the **edit dashboard**. Click the edit dashboard to add new graphs.



Note: Before adding graphs on the dashboard make sure that the host already detected the graphs. You can find the graphs on the configuration > hosts > graphs.

- Click the **edit dashboard** and **add widget** → Select **type** → **Graph**. Provide a name of the widget.



- Select a **data set** which is the hostname and select the **item pattern** where we can find the whalebone template.
- Select the items you want to add on the widget for graphical visualization. After the adding **item pattern**. Select base color for graphs then you can adjust the width, point size, transparency, and fill.
- On here we successfully added a widget on the dashboard. To edit or change the widget, click the gear icon.

Hosts

Host group Linux servers Select

☐ Name

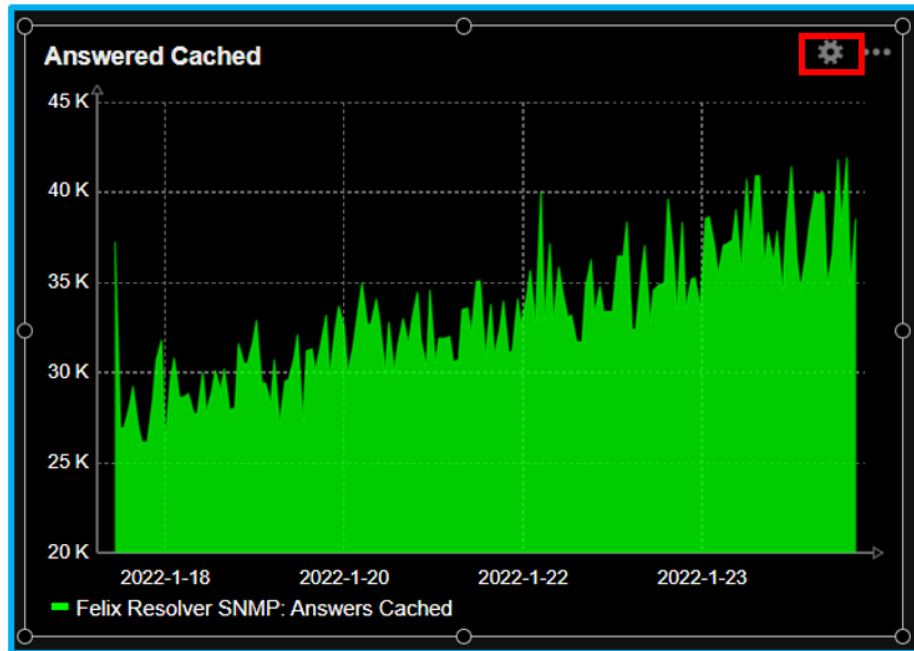
☐ **Felix Resolver SNMP**

Select Cancel

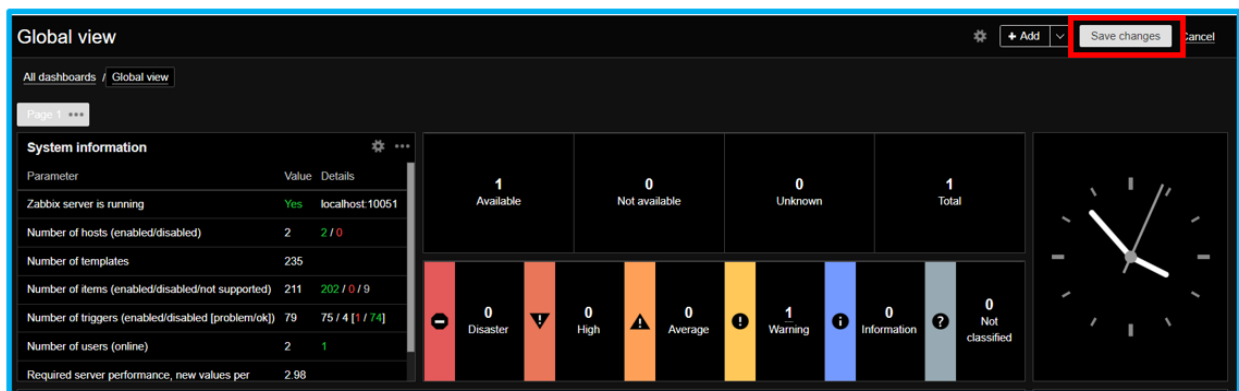
Items

Host Felix Resolver SNMP Select

<input type="checkbox"/> Name	Key	Type	Type of information	Status
<input type="checkbox"/> Answers 1ms	answersers1ms	SNMP agent	Numeric (float)	Enabled
<input type="checkbox"/> Answers 10ms	answersers10ms	SNMP agent	Numeric (float)	Enabled
<input type="checkbox"/> Answers 50ms	answersers50ms	SNMP agent	Numeric (float)	Enabled
<input type="checkbox"/> Answers 100ms	answersers100ms	SNMP agent	Numeric (float)	Enabled
<input type="checkbox"/> Answers 250ms	answersers250ms	SNMP agent	Numeric (float)	Enabled
<input type="checkbox"/> Answers 500ms	answersers500ms	SNMP agent	Numeric (float)	Enabled
<input type="checkbox"/> Answers 1000ms	answersers1000ms	SNMP agent	Numeric (float)	Enabled
<input type="checkbox"/> Answers 1500ms	answersers1500ms	SNMP agent	Numeric (float)	Enabled
<input type="checkbox"/> Answers AA	answersersAA	SNMP agent	Numeric (float)	Enabled
<input type="checkbox"/> Answers AD	answersersAD	SNMP agent	Numeric (float)	Enabled
<input type="checkbox"/> Answers Cached	answerscached	SNMP agent	Numeric (float)	Enabled
<input type="checkbox"/> Answers CD	answersCD	SNMP agent	Numeric (float)	Enabled
<input type="checkbox"/> Answers DO	answersDO	SNMP agent	Numeric (float)	Enabled
<input type="checkbox"/> Answers EDNS0	answersEDNS0	SNMP agent	Numeric (float)	Enabled
<input type="checkbox"/> Answers No Data	answersnodata	SNMP agent	Numeric (float)	Enabled
<input type="checkbox"/> Answers No Error	answerserror	SNMP agent	Numeric (float)	Enabled
<input type="checkbox"/> Answers NX-Domain	answersnodomain	SNMP agent	Numeric (float)	Enabled



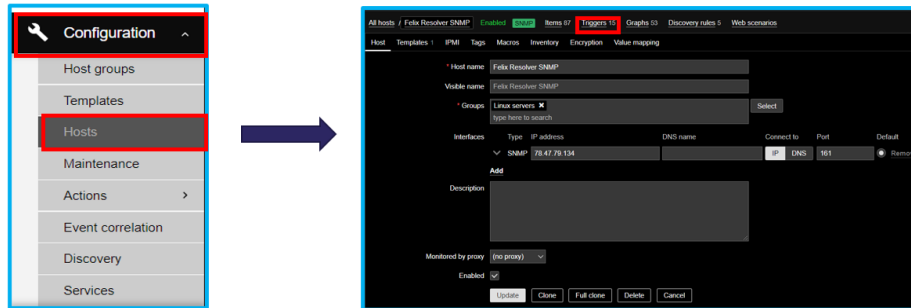
- Don't forget to click the save button on the upper right to save the widget on the dashboard.



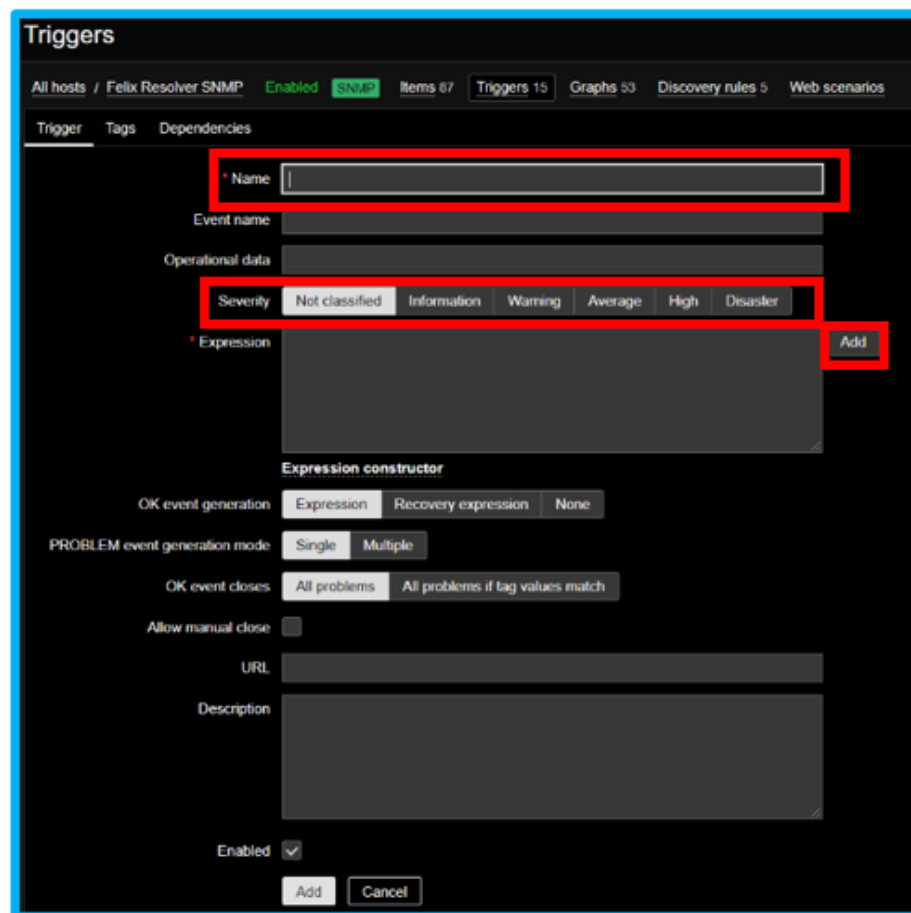
20.6 How to add triggers on the Zabbix

Triggers are logical expression that “evaluate” data gathered by items and represent the current system state. Triggers expression allow to define a threshold of what state of data is acceptable. Therefore, if the incoming data surpass the acceptable state, a trigger is “fired” - or changes status to PROBLEM. Example if the whalebone resolver encountered 1,000 NXDOMAIN the trigger will be initialized to notify us that the data has exceeded from the set threshold.

- To configure the trigger, go to **Configuration - Hosts**. Click the **triggers** tab.



- Create **trigger** → Provide **name** then add an expression. Let say we want to trigger if the resolver **NXDOMAIN** exceeds more than **60**. Select severity for this trigger.



- Click **add** → On the **condition** tab → **Select the item**. On here let's select the **NXDOMAIN**.
- On the **condition** tab, set the **count** → **time shift - now-h** → **result**. On the **result** select an operand then set the value to **60**. This condition will trigger if the **NXDOMAIN** exceed to 60.
- Click **insert** and save the triggers. Make sure the trigger is enabled on the template.

Condition

Item

Select

Function

last() - Last (most recent) T value

Last of (T)

Count

Time shift

now-h

Time

Result

=

0

Insert

Cancel

Condition

Item

Whalebone Resolver: Answers NX-Domain

Select

Function

last() - Last (most recent) T value

Last of (T)

Count

Time shift

now-h

Time

Result

>

60

Insert

Cancel

Sev	Severity	Name	Operational data	Expression	Status	Tags
!	Warning	Answers 1ms less than 40000		last(Whalebone Resolver/answers1ms)<40000	Enabled	
!	Warning	Answers 100ms higher than 5000		last(Whalebone Resolver/answers100ms)>5000	Enabled	
!	Warning	Answers NXDOMAIN higher than 50		last(Whalebone Resolver/answersnxdomain)>50	Enabled	
!	Warning	HDD usage more than 17GB		last(Whalebone Resolver/hddusage)>18	Enabled	

- On the **problems** tab, check the **NXDOMAIN** that exceeds the threshold.

Time	Info	Host	Problem • Severity	Duration	Ack
01/21/2022 05:54:07 AM	Warning	Felix Resolver SNMP	Answers NXDOMAIN higher than 50	4d 3h 45m	No
01/21/2022 05:53:07 AM	Warning	Felix Resolver SNMP	Answers NXDOMAIN higher than 50	4d 3h 46m	No
01/21/2022 05:52:07 AM	Warning	Felix Resolver SNMP	Answers NXDOMAIN higher than 50	4d 3h 47m	No
01/21/2022 05:51:07 AM	Warning	Felix Resolver SNMP	Answers NXDOMAIN higher than 50	4d 3h 48m	No
01/21/2022 05:50:08 AM	Warning	Felix Resolver SNMP	Answers NXDOMAIN higher than 50	4d 3h 49m	No
01/21/2022 05:49:07 AM	Warning	Felix Resolver SNMP	Answers NXDOMAIN higher than 50	4d 3h 50m	No
01/21/2022 05:48:07 AM	Warning	Felix Resolver SNMP	Answers NXDOMAIN higher than 50	4d 3h 51m	No
01/21/2022 05:47:07 AM	Warning	Felix Resolver SNMP	Answers NXDOMAIN higher than 50	4d 3h 52m	No
01/21/2022 05:46:07 AM	Warning	Felix Resolver SNMP	Answers NXDOMAIN higher than 50	4d 3h 53m	No
01/21/2022 05:45:07 AM	Warning	Felix Resolver SNMP	Answers NXDOMAIN higher than 50	4d 3h 54m	No

- On the dashboard, the **NXDOMAIN** that exceeds the threshold is identifiable.

Time	Info	Host	Problem • Severity	Duration	Ack
10:55:07 AM		Felix Resolver SNMP	Answers NXDOMAIN higher than 50	1m	No
10:55:07 AM		Felix Resolver SNMP	HDD usage more than 17GB	1m	No
10:54:07 AM		Felix Resolver SNMP	Answers NXDOMAIN higher than 50	2m	No
10:54:07 AM		Felix Resolver SNMP	HDD usage more than 17GB	2m	No
10:53:07 AM		Felix Resolver SNMP	Answers NXDOMAIN higher than 50	3m	No
10:53:07 AM		Felix Resolver SNMP	HDD usage more than 17GB	3m	No
10:52:07 AM		Felix Resolver SNMP	Answers NXDOMAIN higher than 50	4m	No

20.7 How to configure the trigger actions

Triggers action are logical expression that “evaluate” data gathered by items and represent the current system state. Trigger expression allow to define a threshold of what the data is “acceptable”. Therefore, if the incoming data surpass the acceptable state, a trigger is “fired” or changes status to **PROBLEM**. For this example, lets say the **NXDOMAIN** exceeds to 60. The trigger will initialize an email for the admin reporting or notification.

- First step to set up a trigger action by using an email. Go to **Administration** and **Media types**. Create media type and provide Name → SMTP server → port → SMTP email > user and pass.
- After you setup the email → Go to **Configuration** → **Actions** → **Action triggers**. On the trigger **Actions** → **Create Action** → Provide a name → **Add a condition**.
- On the **new condition** window, select the Type: **Trigger Operator: equals** triggers: Select the **NXDOMAIN**.
- Select the **NXDOMAIN** for Action Triggers. Click Add.

Media types

Media type Message templates Options

* Name

Type

* SMTP server

SMTP server port

* SMTP helo

* SMTP email

Connection security ☐ None ☐ STARTTLS ☒ SSL/TLS

Authentication ☐ None ☒ Username and password

Message format ☐ HTML ☐ Plain text

Description

Enabled ☒

Actions

Action Operations

* Name

Conditions	Label	Name	Action
<input type="button" value="Add"/>			

Enabled ☒

* At least one operation must exist.

New condition

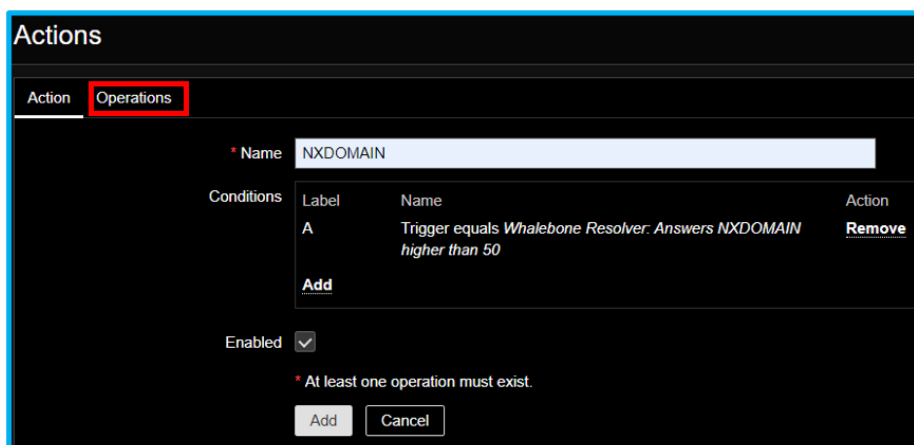
Type

Operator

Triggers



- On the **Actions** → Click the **Operation** → Select the default operation to 1 min. On the operations click **add**.



- Select the step duration to 1 minute. On the operation click **add** → **Send to users** → **Add the Zabbix admin** → **Send only to** → **Email**.

20.8 How to view the pre-defined Whalebone dashboard

For reference, the Whalebone template has a sample dashboard that overviews the data from the resolver.

- To access the dashboard, go to **monitoring** → **hosts**. Then in the **host** click the dashboard.
- This is the overview of the pre-defined Whalebone dashboard.

Operation details

Operation: Send message

Steps: 1 - 1 (0 - infinitely)

Step duration: 0 (0 - use action default)

* At least one user or user group must be selected.

Send to user groups:

User group	Action
Add	

Send to users:

User	Action
Admin (Zabbix Administrator)	Remove
Add	

Send only to: Email

Custom message: ☐

Conditions:

Label	Name	Action
Add		

[Update](#) [Cancel](#)

Graphs	Dashboards
Graphs 52	Dashboards 1
Graphs 53	Dashboards 1
Graphs 52	Dashboards 1
Graphs 25	Dashboards 3
Disp	



USER/ORGANIZATION MANAGEMENT

21.1 User Management

The users can be managed under the respective tab on the **User Menu**.

Under this menu, an Administrator is able manage user accounts by adding, removing or disabling them. Additionally they are presented with an overview of last login and last password change details per account.

Tip: When a user is invited to join an organization and does not already have a Whalebone account, a new account is created for them and an activation link is being sent to their registered email address.

The two types of users that are supported are:

Users: users that have their primary account registered under the specific organization.

External Users: (If available) users that belong to another organization but can be assigned a role under a different Whalebone Portal tenant. e.g. resellers

Tip: Each user can be assigned one or more roles which can be combined to shape their final role. The permissions are additive (stackable).

Below are described the different roles and the actions that they are able to perform.

Action	Read Traf- fic	Read Threats	List Ed- itor	Se- curity policy Admin	API Cre- den- tials	Read only	Oper- ations Read Only	DNS Ad- min	Home- Office Security admin	Users ad- min	Ad- min
View Threat Data											
View DNS Traffic											
View Whitelists/Blacklists											
Edit Whitelists/Blacklists											
View Security Policies											
Edit Security Policies											
View Resolver Configuration											
Edit Resolver Configuration											
View API To- kens											
Generate API Tokens											
View Network Configuration											
Edit Network Configuration											
View Alerts											
Edit Alerts											
View Reports											
Edit Reports											
HOS device management and policy settings											
Manage user accounts											

21.2 Organization Settings

The Organization Setting can be found under the **User Menu**.

21.2.1 Portal Access Policy

Portal Access Policy defines security mechanism for users accessing Whalebone's Portal. The following settings can be configured:

Allowed IP Ranges: IPv4 or IPv6 ranges in CIDR notation, e.g. 10.0.0.0/24 that are allowed to access Whalebone Portal.

Account Lockout: If enabled, it can limit the number of failed login attempts.

The available options are:

- **Failed Login Limit:**

Number of unsuccessful login attempts before locking the account. Default is 5.

- **Lockout Duration:**

Time duration in minutes for disallowing login requests.

- **Lockout Reset Time:**

Time duration in minutes before resetting the number of failed attempts.

- **CAPTCHA Threshold:**

Number of unsuccessful login attempts before enabling the CAPTCHA verification.

Multi Factor Authentication: Require users to use a two factor authentication (2FA) application and enter additional tokens upon logging to the portal.

21.2.2 Password Policy

The following password settings can be configured:

Expiration Time: Number of days before a password needs to be changed.

Password history: Number of old passwords that cannot be reused when setting up a new passwords.

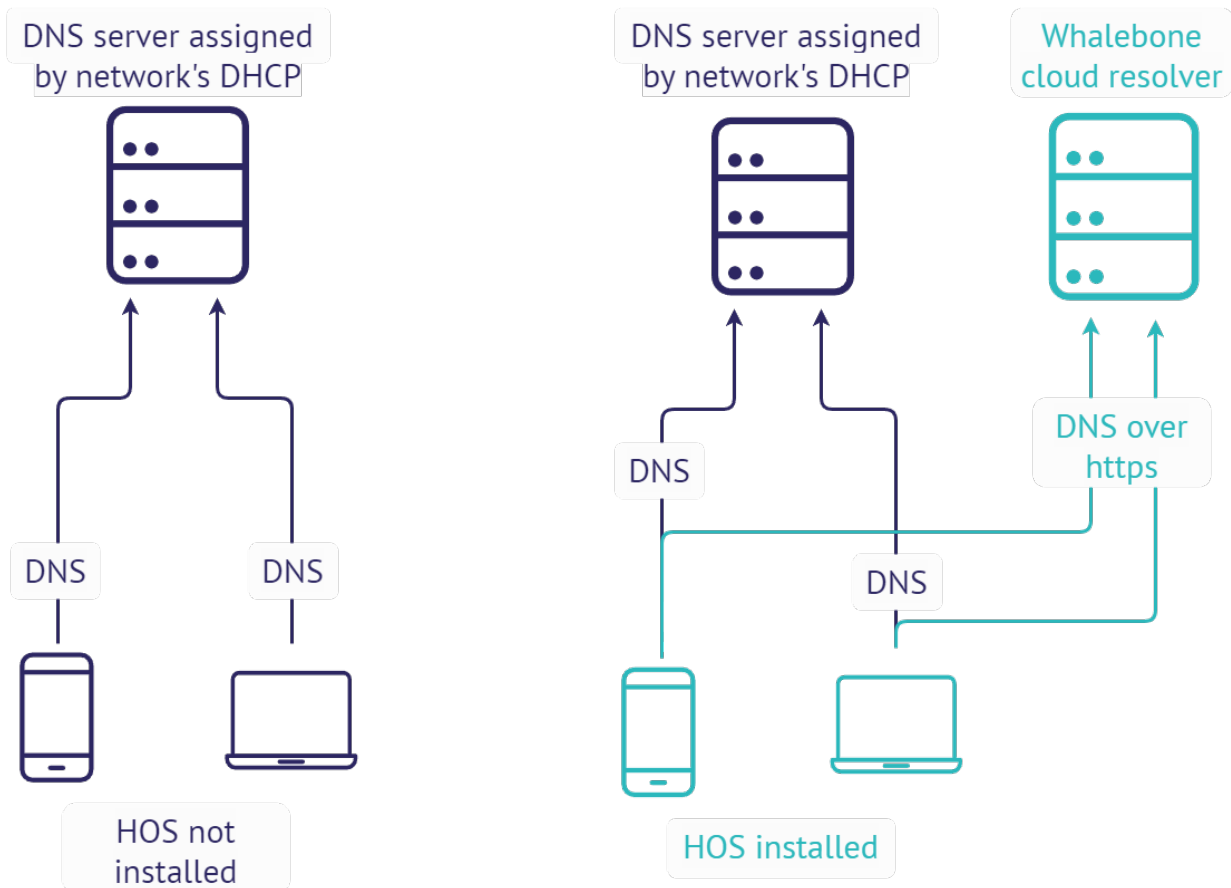
Password Attributes: The attributes that a new password should have.

The attributes that a new password can have are the following:

- Minimum Length
- Number of Digits
- Number of lowercase characters
- Number of uppercase characters
- Number of special characters

HOME OFFICE SECURITY OVERVIEW

Whalebone Home Office Security (HOS) provides an off-network DNS filtering functionality for desktop and mobile devices. It intercepts DNS traffic and inspects it before sending network packets to the wild. It protects the device from network threat by scanning every DNS packet. At the moment, Windows, Android and iOS devices are supported. For detailed OS version support, see below.



HOS comes with Windows Installer for the deployment. No user interaction is required to perform the installation, however the installer requires a token.

The default target directory is:

C:\Program Files (x86)\Whalebone\Home Office Security\

For Android the default install location is:

/storage/emulated/0/Android/io.whalebone.securedns.corp/

22.1 Supported OS

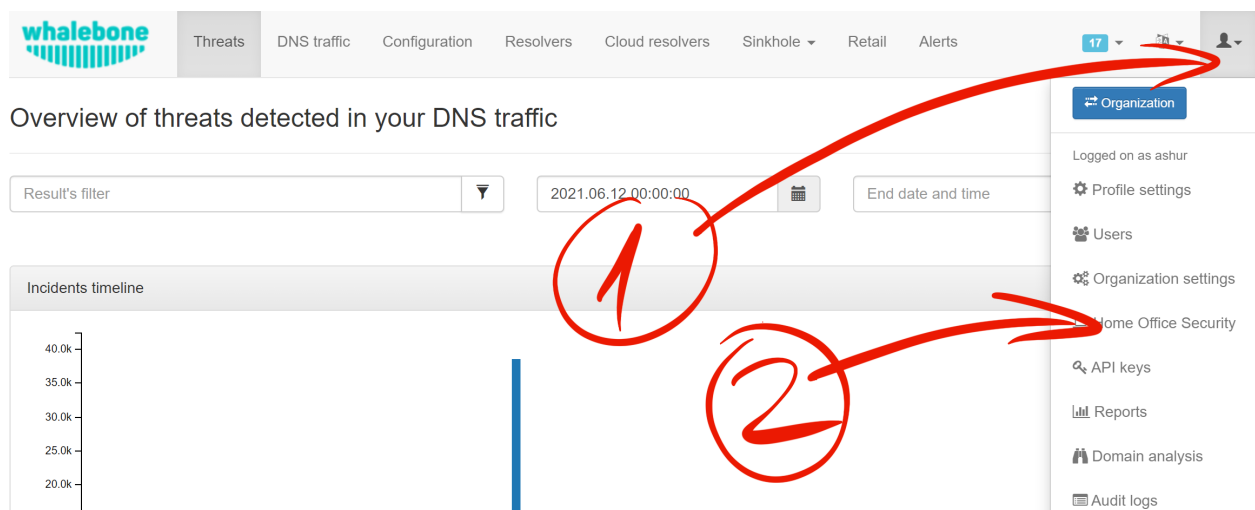
Windows Desktop	Windows 7 or higher
Windows Server	Windows Server 2012 or higher
Android	Android 5 or higher
iOS	All versions
MacOS	Not supported
Linux	Not supported

Windows 7 systems must be up-to-date or at least have KB3033929 installed.

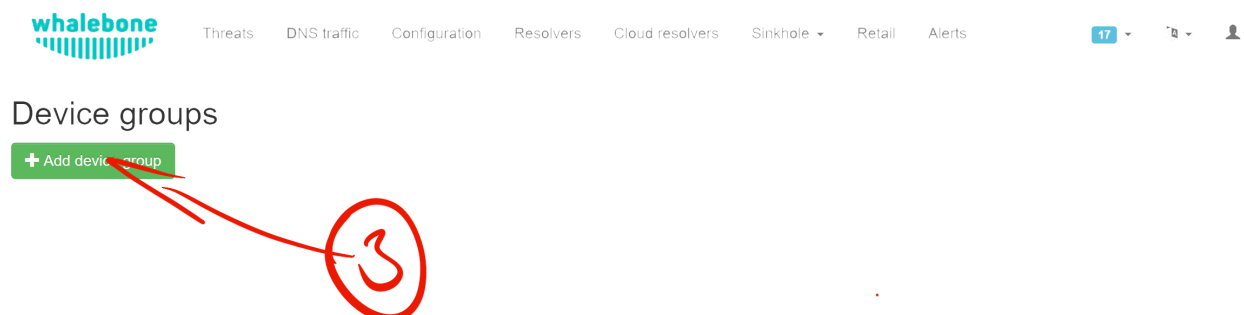
Windows Server 2016 systems must have secure boot disabled.

STEP BY STEP INSTALLATION

To install HOS on device you need to configure it first. Open **Whalebone Portal** web page and use (1) **User menu** to navigate to (2) **Home Office Security**.



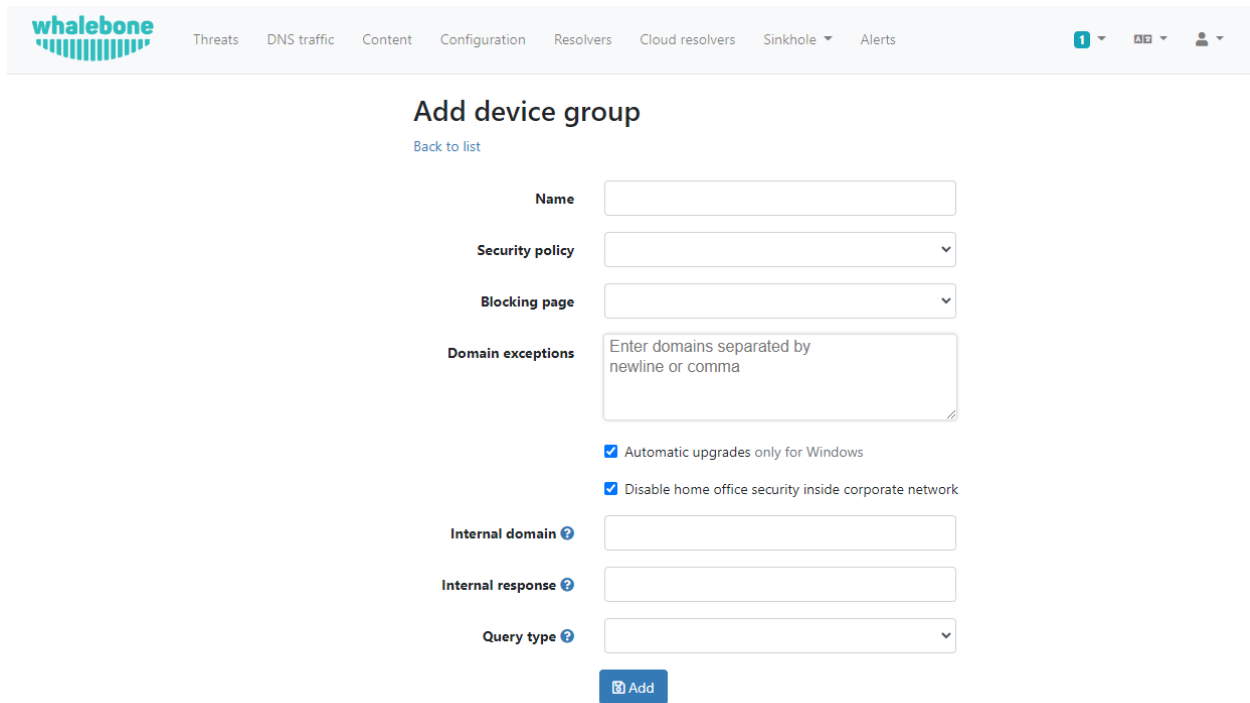
A Default Device group should already exist. If not, create one by clicking the (3) a + **Add device group** button.



- **Name:** This should clearly identify the device group to differentiate it from others. If you only use one, you may leave its name as Default Group.
- **Policy:** corresponds to the policies you create in the Configuration menu. It is a set of rules that instructs how to operate. Based on policy the device or the local/cloud resolver decides what to during DNS resolution. This set of rules persist on the device and is updated initially and later synchronized. Because of this Portal provides monitoring of these devices.
- **Blocking page:** corresponds to the blocking pages you create in the Configuration menu.
- **Domain exceptions:** HOS service will not divert any DNS queries that contain question for domain on the exception list. E.g. when `example.com` is specified, the DNS request will be resolved as usual on the resolver configured by operating system. A same rule applies for question `subdomain.example.com`.
- **Automatic upgrade:** When this configuration option is checked, HOS application on Windows will update to latest production version when a newer version is available to download. This option takes effect on Windows only, on mobile upgrades are performed by the vendor ecosystem.
- **Disable HOS inside the corporate network: When this option is checked, 3 more text boxes will appear. The configuration allows the HOS to be disabled within the corporate network based on a query-response process.**
 - **Internal Domain:** Specifies which internal domain HOS will periodically query.
 - **Internal Response:** HOS expects the response specified in this field after sending a query to the internal domain.
 - **Query Type:** According to the selected query type (A, AAAA and MX), the record on the internal domain controller must be configured correctly.

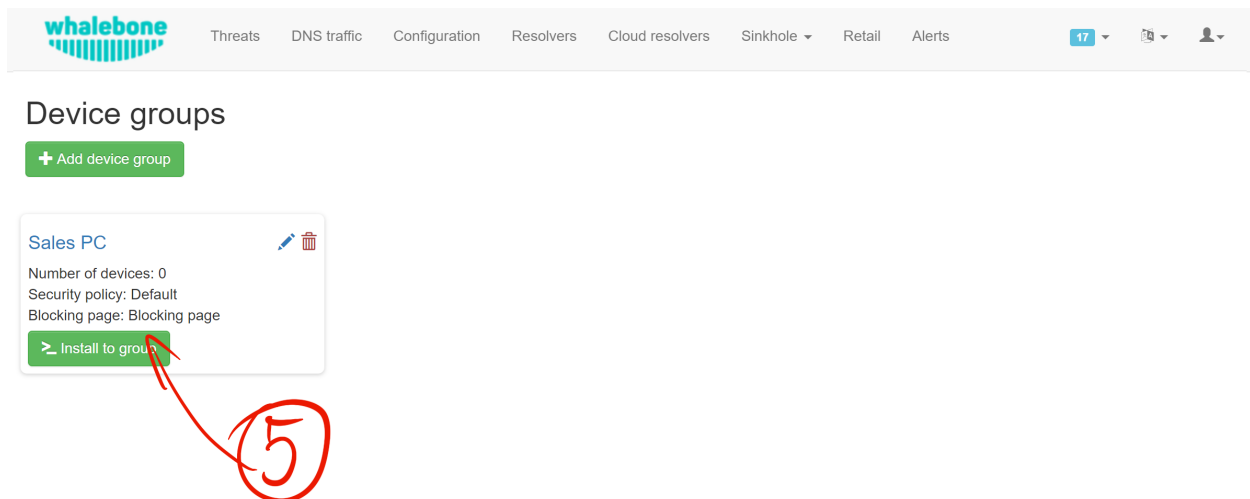
Warning: Please note that two settings mentioned above (Automatic upgrade and Domain exception) are featured in version 2.10.0 for Windows only. If you are running earlier version, please update to 2.10.0 manually.

When you're done, click **Add** button to create this group.



The screenshot shows the 'Add device group' form in the Whalebone admin interface. The form includes fields for Name, Security policy, Blocking page, and Domain exceptions. There are also checkboxes for 'Automatic upgrades only for Windows' and 'Disable home office security inside corporate network'. Below these are fields for Internal domain, Internal response, and a Query type dropdown. An 'Add' button is at the bottom.

Click (5) **Install to group** button to see installation instructions and/or get download link to the HOS installer.



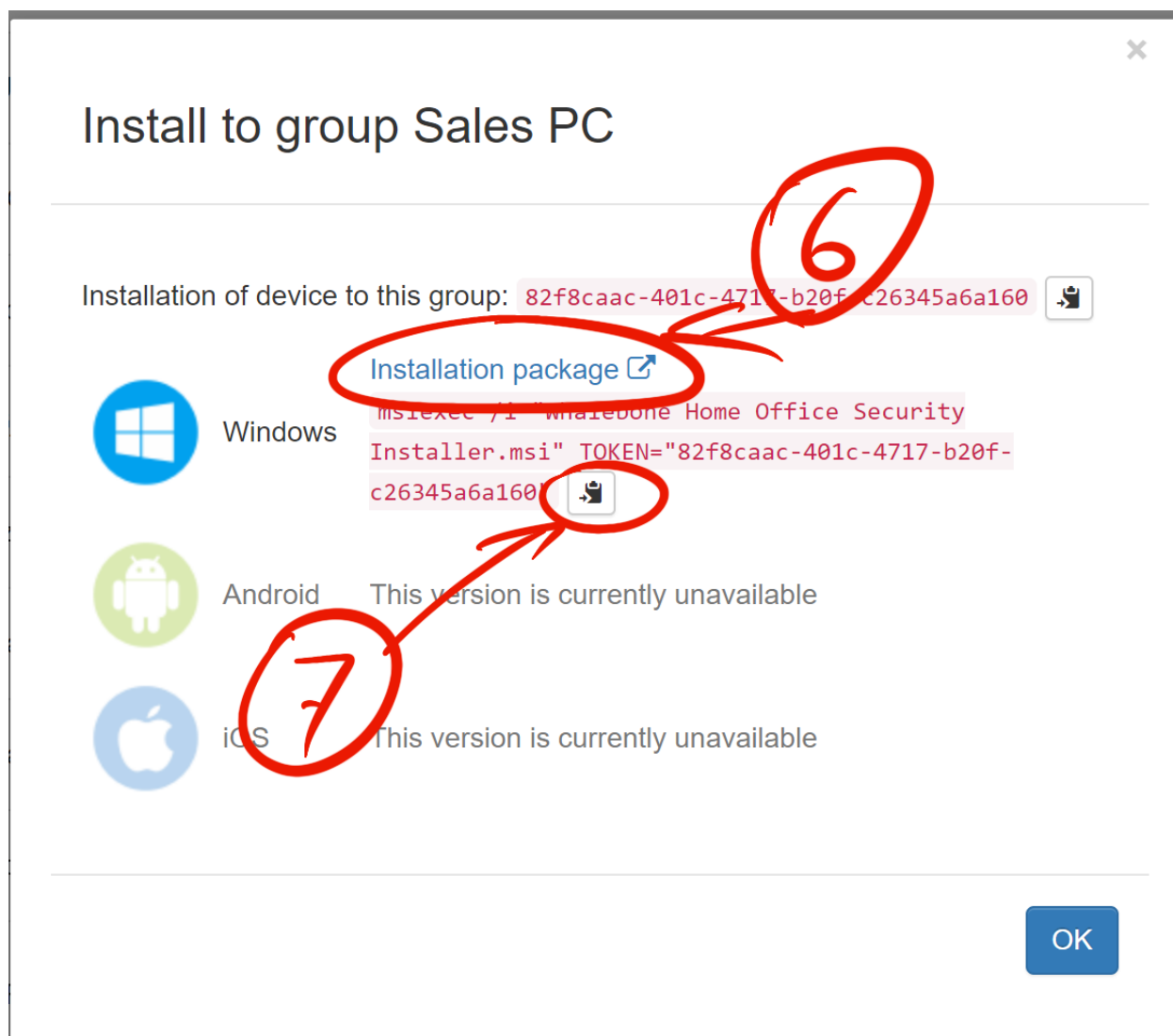
The screenshot shows the 'Device groups' section of the Whalebone admin interface. It features a '+ Add device group' button and a list of device groups. The first group is 'Sales PC', which has 0 devices and a default security policy. A red arrow points from a circled number '5' to the 'Install to group' button within the 'Sales PC' group card.

If you haven't already download the installer (6). While the installer is being downloaded please copy the installation command to clipboard (7). To install or Update:

```
msiexec /i "Whalebone.Home.Office.Security.Installer.msi" TOKEN="60d5806e-07fe-432a-a4ad-7797d82782b3"
```

Uninstall:

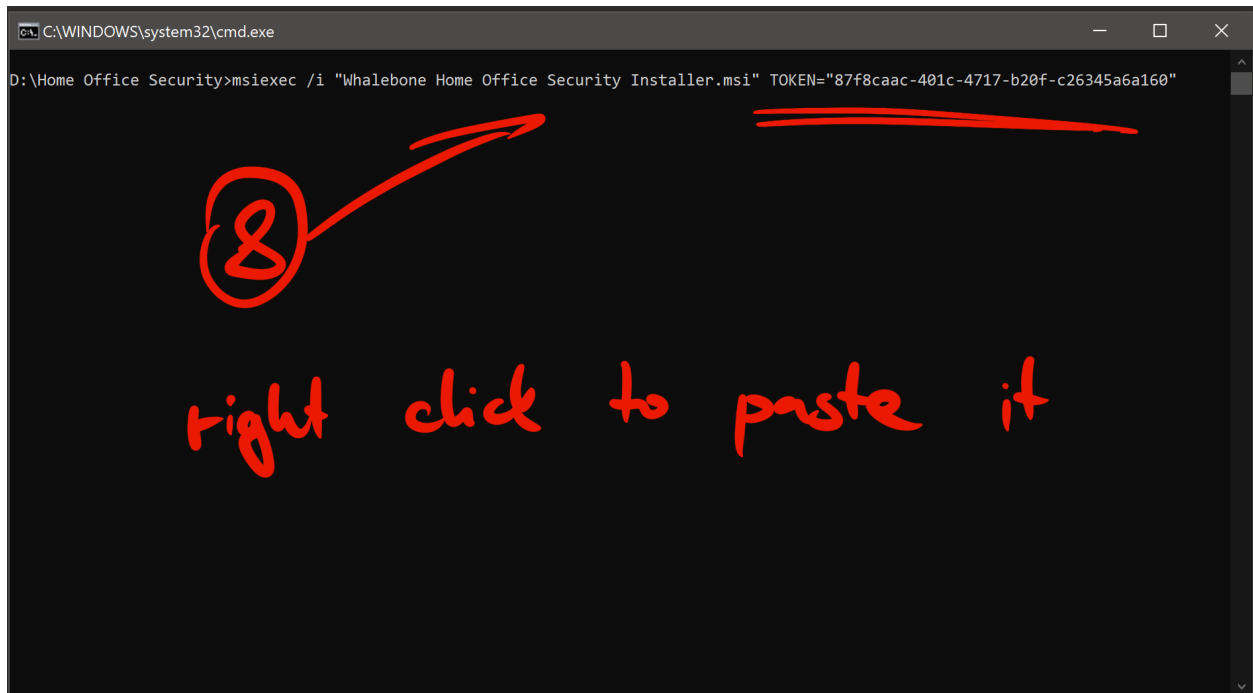
```
msiexec /x "Whalebone.Home.Office.Security.Installer.msi"
```



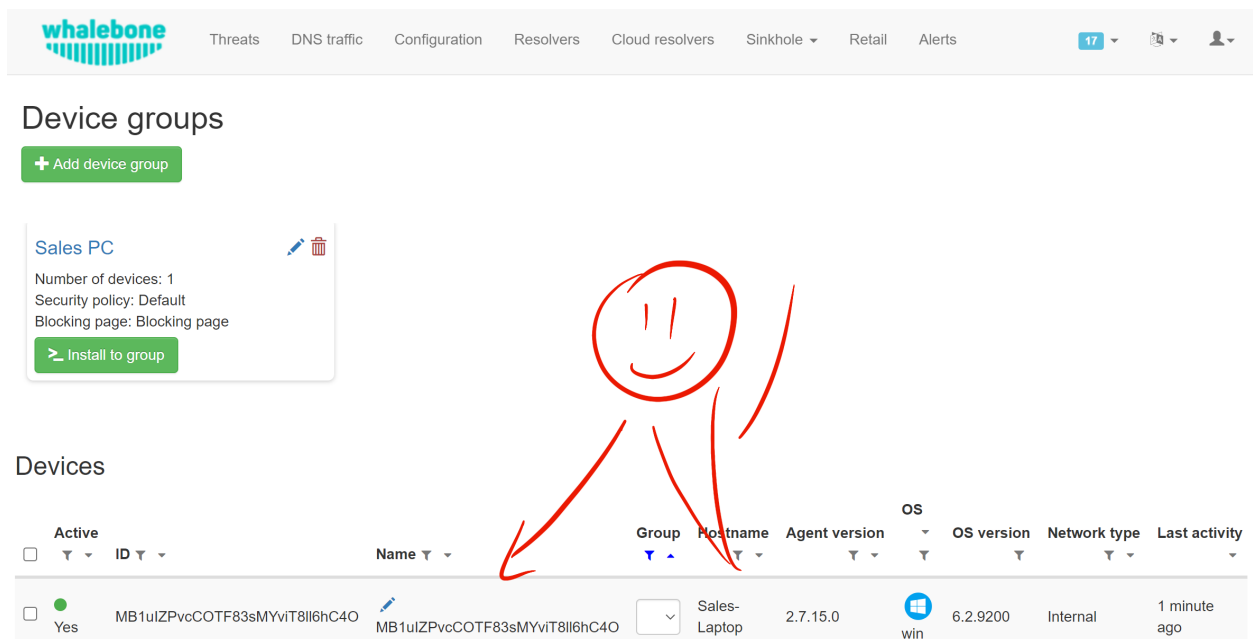
Find the folder where the installer is located. It should be file named **Whalebone.Home.Office.Security.Installer.msi**.

Open up a command prompt, change directory to the folder where is the installer and paste (8) the command with your mouse (right click). Execute the command. This requires admin privileges.

Installer will end prematurely with error when executed without token argument.



Tip: The installer has very minimal UI. If there was no error message, consider the installation successful.



Device is now visible in the Whalebone Portal web page.

24.1 Devices

Your organization may divide devices into single or multiple groups. Every device may belong exactly to a single group only. Each must be a member of **Device group** before they get monitored. Each group provides a security **Policy** which is later conditionally applied to them. Whether the device is present on the **internal** or **external** network makes it **active** or **inactive**.

It separates the network location into **internal** or **external** and the biggest role here has the **Internal domain** setting which must be defined in the **Device group**. If HOS detects the **Internal domain** the network location is decided as **internal**. Detection is performed by running DNS query for the configured internal domain and receiving the configured answer.

24.2 States

HOS is constantly monitoring changes on the network interfaces and based on the conditions it changes its states.

Active

All DNS traffic is diverted to DoH server. HOS becomes **Active** when it is connected to the public network, but the **Internal domain** is unreachable. This state is used for the danger zones such as public wifi.

Inactive

DNS traffic is left intact. This state is used when device can't connect to the Internet or when it is connected through internal network.

24.3 Security

In the background HOS uses **DNS-over-HTTPs** or **DoH**. The **Hostname** of the **Resolver** is never diverted and is cached. The identification and authenticity is left to the TLS protocol. When device belongs to any **Domain**, then all domain names and their subdomains are allowed to reach the DNS servers they route to. HOS uses Win32_NetworkAdapterConfiguration WMI table to get the information.

24.4 Service requirements

24.4.1 Windows

Because HOS must intercept network traffic it requires to run as **SYSTEM** account. You can query the service by name **hos** to see if it started properly. When none or invalid installation token is supplied the service it will stop.

```
C:\Users\admin>sc query "Whalebone Home Office Security"

SERVICE_NAME: HOS
        TYPE               : 10  WIN32_OWN_PROCESS
        STATE                : 4   RUNNING
                                (STOPPABLE, PAUSABLE, ACCEPTS_SHUTDOWN)
        WIN32_EXIT_CODE       : 0   (0x0)
        SERVICE_EXIT_CODE   : 0   (0x0)
        CHECKPOINT           : 0x0
        WAIT_HINT            : 0x0
```

On first run HOS also installs windivert system driver.

```
C:\Users\admin>sc query windivert type=kernel

SERVICE_NAME: windivert
        TYPE               : 1   KERNEL_DRIVER
        STATE                : 4   RUNNING
                                (STOPPABLE, NOT_PAUSABLE, IGNORES_SHUTDOWN)
        WIN32_EXIT_CODE       : 0   (0x0)
        SERVICE_EXIT_CODE   : 0   (0x0)
        CHECKPOINT           : 0x0
        WAIT_HINT            : 0x0
```

Service is configured to recover after crash three times and then stay stopped.

24.4.2 Android

The Android app has access to:

- Location
 - precise location (GPS and network-based)
- Camera
 - take pictures and videos (to scan QR code of the Device group from the portal)
- Wi-Fi connection information
 - view Wi-Fi connections
- Other
 - view network connections
 - connect and disconnect from Wi-Fi
 - full network access (to create a VPN tunnel to Whalebone Cloud resolvers)
 - run at startup

24.5 Application Firewall Settings

Enable TCP port 443 for the **Whalebone Home Office Security.exe** in the application firewall. To enable it for all network profiles in Windows, adjust following command to let HOS connect to your DoH server (e.g. 185.150.10.71):

If HOS service does not work please ensure that HOS service can connect to **hos.whalebone.io** and **mobileapi.whalebone.io**.

```
netsh advfirewall firewall add rule name="Whalebone Home Office Security" dir=out_
↪action=allow program="C:\Program Files (x86)\Whalebone\Home Office Security\Whalebone_
↪Home Office Security.exe" enable=yes remoteip=185.150.10.71,LocalSubnet
```

It is not necessary for the service to listen on port 53, thus there is no requirement for the application firewall to follow.

Additionally, service is listening on **TCP endpoint localhost:9000** to provide data endpoint for UI app, and UI app server **whosui.exe** listens on *TCP endpoint localhost:55221* to render graphical components. Even though these ports are not critical for HOS operation they are relevant for UI app **AdminUI.exe**. Please ensure that services are allowed to listen on those local ports as this allows user to have insight into app operation.

24.6 Application Logs

Service logs can be found at `c:\ProgramData\Whalebone\Home Office Security\Logs\`, which contain detailed information about application states and operation. In case you encounter unexpected service behaviour please include this Log folder and/or Config folder along inside your support ticket. Application provides additional information for operation trace, in AdminUI.exe app, Events tab may give you better insight in HOS operation.

24.7 Uninstalling the app

To completely remove the app, uninstall the service and delete all contents from `c:\ProgramData\Whalebone\Home Office Security\`

DEPLOYMENT

25.1 On-premise resolver deployment

Unlike other similar services, Whalebone can be deployed as a full-fledged local DNS resolver. This is the type of deployment we encourage. The installation is fairly simple. All you need is access to Whalebone Portal and a virtual or physical server, which is rather undemanding in terms of hardware. First, let's take a look at system requirements. Whalebone supports the latest versions of the most popular Linux distributions Debian, Ubuntu, CentOS, and Red hat Enterprise Linux. The minimum hardware sizing is 2 CPU cores, 4 GB RAM, and a 40 GB HDD. Such a machine can handle up to 20,000 users.

Before setting up your server, make sure you don't fall short of the network requirements and prevent the machine from being reachable from outside your network. Once the server is ready, go to the Whalebone Portal and create a new resolver. Come up with a fitting name, which can be changed later on. Once you initiate the addition of the new resolver, you'll see the installation script one-liner command. Copy it to the clipboard. At this point, access the terminal of the server created for this resolver. All that's left to do is to run the installation script previously copied to the clipboard. The installation shouldn't take more than a couple of minutes. The script will inform you about the progress. If the installation wasn't successful, send us the installation log and we'll look into it. Before long, the status of the resolver changes. As soon as the resolver becomes Active, you can route the traffic to it and start protecting your network.

25.2 Cloud resolvers

Whalebone also offers cloud resolvers with malware protection and content blocking. Their addresses are to be found in the Whalebone Portal in the Cloud resolver tab. You can use them directly as primary or secondary resolvers or as a backup to your existing local resolver. It's not rocket science to use them.

First of all, type in your public IP ranges you want to direct to the cloud resolvers. Afterward, all you need to do is set the Whalebone cloud resolver address as the DNS server address in your network. As with the local resolvers, you can create different policies and assign them to individual IP addresses or ranges. This allows you to offer Whalebone to institutions such as schools, which don't necessarily get their connectivity from you, but you administer their network. After having saved and directed the traffic, you're good to go. Just wait for the changes to be propagated to your clients.

CONFIGURATION

26.1 Basic configuration

Every network has its own specific needs. Whalebone can and will adapt to every single one of them. One of the key components that need to be configured when implementing Whalebone is setting up your “Security Policies.” This part of the configuration allows you to adjust the default settings. You can for example lower the blocking threshold or deactivate blocking entirely which leaves you with the audit mode. In this mode, Whalebone monitors the incidents without preventing them from happening. The core of the configuration of audit and blocking is a so-called “score”, which is assigned to individual domains by our algorithm. The higher the score, the more dangerous the domain. It’s up to you to choose from the preset levels of sensitivity or decide to adjust the threshold manually. We advise ISP networks to “block carefully”. The lower the threshold, the more sensitive the blocking. Keep in mind, though, that setting a low threshold increases the risk of false positives.

You can also choose different types of threats to be blocked. If needed, you can easily create your own blocking lists or define domains that should always be accessible. Our customers love that Whalebone can meet the legal blocking requirements of their government for them. If you don’t find your country our list, let us know and we’ll make sure it gets there. If you activated the content filtering add-on, you can configure it here as well. Create as many unique security policies as you want. Afterward, you can go into the configuration of a given resolver and assign these policies to different IP addresses or ranges. All you need to do is to go to the “Policy Assignment” section in the resolver details and assign a policy to a particular IP address or range. Make sure to save the settings.

26.2 Security policies

One of the key components that need to be configured when implementing Whalebone is setting up your “Security Policies. This part of the configuration allows you to adjust the default settings. You can for example lower the blocking threshold or deactivate blocking entirely which leaves you with the audit mode. In this mode, Whalebone monitors the incidents without preventing them from happening. The core of the configuration of audit and blocking is a so-called “score” which is assigned to individual domains by our algorithm. The higher the score, the more dangerous the domain. It’s up to you to choose from the preset levels of sensitivity or decide to adjust the threshold manually.

We advise ISP networks to **block carefully**. The lower the threshold, the more sensitive the blocking. Keep in mind, though, that setting a low threshold increases the risk of false positives. You can also choose different types of threats to be blocked.

If needed, you can easily create your own blocking list or define domains that should always be accessible. Our customers love that Whalebone can meet the legal blocking requirement of their government for them. If you don't find your country our list, let us know and we'll make sure it gets there.

If you activated the content filtering add-on, you can configure it here as well. Create as many unique security policies as you want. Afterward, you can go into the configuration of a given resolver and assign these policies to different IP addresses or ranges. All you need to do is to go to the **Policy Assignment** section in the resolver detail and assign a policy to a particular IP address or range. Make sure to save the settings.

26.3 Blocking page configuration

With Whalebone, you can fully customize blocking pages, which appear in case someone attempts to access a dangerous website in their browser. This tool needs a local resolver, where you can switch the blocking page from cloud to on-premise. In order to configure blocking pages, go to **Configuration** and then **Blocking pages**. You can adjust the existing ones or create a brand-new one. When creating a new blocking page, you can define its name, the domain, and the language of the page. Afterward, fill in all the necessary data including the name of the company, its logo and contact information. Naturally, you can change the information later on. If you want to do so, use the magic stick or edit directly in the HTML code. You can modify the design as well as the content of the blocking page as you choose. All you need to do is to preserve the necessary variables shown over the blocking field.

Once you have saved the modified blocking page, go to **Resolvers** and select the resolver to which you would like to apply the blocking page. Go to "Policy assignment" and apply the blocking page to a given resolver. Alternatively, you can assign it to a specific IP address or range. While you're at it, you can also activate a **bypass**, which will allow the user to access the blocked domain nonetheless.

26.4 Alerts

Set up Whalebone alerts and get live updates about what's going on with your resolvers, how secure your network is, and how well your DNS resolution works. The basic setup is simple: just choose what type of information you want to get and how often you want to be alerted. You can get alerts via E-mail or Slack. You can also integrate Whalebone alerts into your systems through webhooks or syslog. For the status of the resolver, resolution, and server it runs on. We would argue that everyone should at least create alerts.

Make sure to start by setting up alerts for resolution failures. Afterward, set up alerts for hardware resources failure, such as insufficiencies concerning the HDD, RAM, or CPU capacity. You can also monitor failures in communication between the resolver and the Whalebone cloud when the resolution works just fine, but the resolver isn't in sync with Whalebone data centers.

You can even create advanced alerts for DNS traffic and security incidents. We will gladly give you a hand with setting advanced alerts, no matter if it's during the introductory technical consultation, at the end of the trial or any time you decide to contact Whalebone support.

ANALYSIS

27.1 Domain analysis

There are two ways to manually perform an analysis of a domain against the Whalebone database. One way to open the **Domain Analysis** tool is from the user's menu. The other option is to check a specific domain from the context menu in **Threats** or **DNS traffic** overviews directly. Afterward, you will see all the information that Whalebone has collected about the domain. We used **kidos-bank.ru** as an example. We can see that there are different types of threats associated with the domain. Its score is 95-100 and it was labeled as dangerous in November 2019. In the following graphs, you can see the development of the detections, or rather the DNS resolution requests of the domain in your network. The outcome of the analysis also shows that the domain is not assigned a content category and its blocking wasn't ordered by law. You can inquire into any domain like that. Just enter it into the **top field**. We can see that **facebook.com** is not considered a security threat, there's quite some traffic going on and Whalebone categorizes it as a **social network**. If we type in **porn.com**, we can see that the category has changed into **Sexual content**.

27.2 DNS traffic

You can see the timeline of the DNS requests and answers of the last 1,7 or 14 days in the "DNS traffic" log. The log shows the first resolution of the domain by a given IP address in the last 24 hours, the type of query, the outcome of the resolution, the source and destination IP address. It also enables you to do a full-text filtration using wild card operators.

The summarizing logarithmic graphs under the main timeline display an overview of the most common answers, second-level domains, and IP addresses with the heaviest traffic. All the data is accessible in a table format, too, and you can even export them to a CSV file with a maximum of 1,000,000 lines. The DNS traffic logs are temporarily stored on the resolver's server. You can access them from there for your own processing. One of the biggest advantages of the DNS traffic log is the possibility of filtering errors in responses such as NXDOMAIN and SERVFAIL. This allows you to see the malicious traffic on devices connected to the network. This video shows a hashed IP address with almost 240,000 resolutions of different domains leading to NXDOMAIN and SERVFAIL errors. Here, you can see both public and private IP addresses.

This display is particularly useful especially if you add other queries to the filter, such as MX. Such as setting of the filter shows you IP addresses in your network, which send spam and are therefore in danger of being blacklisted and consequently endangering other customers as well, in case they're behind NAT. Similarly, you can choose for example A queries. We specialize in the detection of DGA malware communication. Clients, who are infected in this way, connect to quasi-randomly generated domains that try to communicate with the command center of the malware.

27.3 Threats

Whalebone is all about protecting your network. That's why you can access a complete overview of incidents that have happened in the last three months. Not only does the overview offer information, but it also provides you with the possibility of filtration and data analysis. The results are divided into three categories; events that have been blocked, audited, and allowed. The audited domains represent domains, which are somewhat suspicious. Their score is high enough to be listed in the log but lower than the blocking threshold. When it comes to blocked domains, the resolver returns a fully-customized blocking page with an optional bypass button.

You can also filter the data by the type of incident. Let's take a look at the example of communication with the command center of the malware. We can see specific blocked domains as well as local or public IPs that tried to access them. This is an example of active intensive traffic from a specific IP address and communication with malware called Necurs. Such an infected client would affect the quality of other client's connections as well. For every single record, you can choose different types of domain checks in the context menu. It's very practical to start the analysis by googling the domain. More often than not, though, the results will only tell you that the domain is dangerous.

Another way of checking the domain is by using various security sources. An example of such a service is a very useful website Virustotal. If you aren't convinced that there was a good reason for the blocking even after the analysis, feel free to report such a domain to us. We will examine the case and get back to you. In case it truly turns out to be a false positive blocking, we will globally allow access to the domain for all Whalebone customers.

27.4 Data Analysis

The Whalebone Portal allows detail full-text filtration and associated data analysis. The thorough manual is to be found in the technical documentation available at docs.whalebone.io. You will find a list of different operators, examples of their usage, and references to the potential difference between the DNS traffic and threats overview. You can use wildcard or logical operators. When using full-text filtration, all the parameters are to be type directly into the URL address. This way, you can easily create filters for future use.

27.5 API

With Whalebone API, you can integrate Whalebone into your own systems. This allows you to make use of all the advantages of Whalebone. First of all, you need to create a new key. Go to the API keys configuration from the context menu. After a new API key is created, you will see all the necessary details. The secret for the API key will never be displayed again, so make sure you really copied it. You can always invalidate the API key. Just click the corresponding icon. We have a detailed interactive documentation for Whalebone API. Just click the icon in the API keys overview or go directly to apidocs.whalebone.io/public. The documentation will take you through different categories of information and settings with specific examples. The “Event” section contains all the information about threats such as types of threats and domains. You can even model API calls directly in the documentation and use them right away. On top of that, the API contains certain information that isn’t available in the Whalebone Portal yet, such as the DNSSEC validation details. Naturally, you can access information about resolvers, such as latency, the health of the resolvers, or the usage of system resources. Before you start modeling API calls in the documentation, we recommend authorizing it with your API keys. This will allow you to directly work with your account in the documentation.

27.6 Domain resolution troubleshooting

When internet users can’t access a domain, they often think it’s the ISP’s fault. More often than not, you’re not the one to blame, it’s the domain itself. No matter what, you still have to answer the customer and explain the situation. Let’s take a look at how Whalebone improve this process.

First of all, examine the potential domain blocking by searching the domain in “Threats”. We recommend using search operators and querying for subdomains. It turns out that the domain “sufr.cz” has not been blocked as a threat. The second step is to go to “DNS traffic” and check if the domain was even accessed by anyone. If so, take a look at how Whalebone deal with the resolution. It turns out there have been attempts to access the domain. In that case, we have to check the results. We can see that the response for this domain was SERVFAIL. To further the troubleshooting process, we can analyze the domain through the context menu.

We recommend using the DNS Viz tool. DNS Viz is designed to fully inspect the DNS resolution behavior. A direct click-through leads to the DNSSEC validation results. It turns out that the problem of this particular domain is that it has issues with expired cryptographic signatures. If you feel like you still don’t really know what’s going on with the domain, feel free to contact us via E-mail at support@whalebone.io. We will gladly look into your issue.

27.7 Domain Tracing

A well-working DNS resolution is essential for a functional internet connection. That's why you can make sure that the individual resolvers are functioning all right in the administration portal. All you need to do is choose the corresponding local resolver, open the context menu and click "Trace domain". At this point, type in the domain you want to examine. Let's say it's whalebone.io.

Choose one of the query types, for example, "A" and trace the domain. You can see the outcome of the resolution here. The upper part shows the result of the query. The green color tells you there's nothing wrong with the DNS resolution. If there's an issue, there will be some information about the particular problem in orange or red. For example, if the domain doesn't exist, the result will be NXDOMAIN. In case there's an issue with the resolution, you will see the "SERVFAIL" response. If you encounter any issues, send the log to support@whalebone.io and we'll look into it.

LICENSE DISCLAIMERS

The Whalebone resolver utilizes the following technology in its solution:

28.1 the CRC64 variant with Jones coefficient

Copyright (c) 2012, Salvatore Sanfilippo <antirez at gmail dot com>
All rights reserved.

Redistribution **and** use **in** source **and** binary forms, **with or** without
modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice,
this **list** of conditions **and** the following disclaimer.
- * Redistributions **in** binary form must reproduce the above copyright
notice, this **list** of conditions **and** the following disclaimer **in** the
documentation **and/or** other materials provided **with** the distribution.
- * Neither the name of Redis nor the names of its contributors may be used
to endorse **or** promote products derived **from** **this** software without
specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE
POSSIBILITY OF SUCH DAMAGE.

28.2 the Lightning.NET Library

The OpenLDAP Public License
Version 2.8, 17 August 2003

Redistribution and use of this software and associated documentation ("Software"), with or without modification, are permitted provided that the following conditions are met:

1. Redistributions in source form must retain copyright statements and notices,
2. Redistributions in binary form must reproduce applicable copyright statements and notices, this list of conditions, and the following disclaimer in the documentation and/or other materials provided with the distribution, and
3. Redistributions must contain a verbatim copy of this document.

The OpenLDAP Foundation may revise this license from time to time. Each revision is distinguished by a version number. You may use this Software under terms of this license revision or under the terms of any subsequent revision of the license.

THIS SOFTWARE IS PROVIDED BY THE OPENLDAP FOUNDATION AND ITS CONTRIBUTORS ``AS IS'' AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE OPENLDAP FOUNDATION, ITS CONTRIBUTORS, OR THE AUTHOR(S) OR OWNER(S) OF THE SOFTWARE BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The names of the authors and copyright holders must not be used in advertising or otherwise to promote the sale, use or other dealing in this Software without specific, written prior permission. Title to copyright in this Software shall at all times remain with copyright holders.

OpenLDAP is a registered trademark of the OpenLDAP Foundation.

Copyright 1999-2003 The OpenLDAP Foundation, Redwood City, California, USA. All Rights Reserved. Permission to copy and distribute verbatim copies of this document is granted.