# Best practices for secure Zabbix setup

## Overview

This section contains best practices that should be observed in order to set up Zabbix in a secure way.

The practices contained here are not required for the functioning of Zabbix. They are recommended for better security of the system.

## Principle of least privilege

The principle of least privilege should be used at all times for Zabbix. This principle means that user accounts (in Zabbix frontend) or process user (for Zabbix server/proxy or agent) have only those privileges that are essential to perform intended functions. In other words, user accounts at all times should run with as few privileges as possible.

Giving extra permissions to 'zabbix' user will allow it to access configuration files and execute operations that can compromise the overall security of infrastructure.

When implementing the least privilege principle for user accounts, Zabbix frontend user types should be taken into account. It is important to understand that while a "Zabbix Admin" user type has less privileges than "Zabbix Super Admin" user type, it has administrative permissions that allow managing configuration and execute custom scripts.

Some information is available even for non-privileged users. For example, while *Administration* → *Scripts* is not available for non-Super Admins, scripts themselves are available for retrieval by using Zabbix API. Limiting script permissions and not adding sensitive information (like access credentials, etc) should be used to avoid exposure of sensitive information available in global scripts.

## Secure user for Zabbix agent

In the default configuration, Zabbix server and Zabbix agent processes share one 'zabbix' user. If you wish to make sure that the agent cannot access sensitive details in server configuration (e.g. database login information), the agent should be run as a different user:

1. Create a secure user
2. Specify this user in the agent configuration file ('User' parameter)
3. Restart the agent with administrator privileges. Privileges will be dropped to the specified user.

## UTF-8 encoding

UTF-8 is the only encoding supported by Zabbix. It is known to work without any security flaws. Users should be aware that there are known security issues if using some of the other encodings.

### Setting up SSL for Zabbix frontend

On RHEL/Centos, install mod_ssl package:

```
yum install mod_ssl
```

Create directory for SSL keys:

```
mkdir -p /etc/httpd/ssl/private
chmod 700 /etc/httpd/ssl/private
```

Create SSL certificate:

```
openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout
/etc/httpd/ssl/private/apache-selfsigned.key -out /etc/httpd/ssl/apache-
selfsigned.crt
```

Fill out the prompts appropriately. The most important line is the one that requests the Common Name. You need to enter the domain name that you want to be associated with your server. You can enter the public IP address instead if you do not have a domain name. We will use *example.com* in this article.

```
Country Name (2 letter code) [XX]:
State or Province Name (full name) []:
Locality Name (eg, city) [Default City]:
Organization Name (eg, company) [Default Company Ltd]:
Organizational Unit Name (eg, section) []:
Common Name (eg, your name or your server's hostname) []:example.com
Email Address []:
```

Edit Apache SSL configuration:

```
/etc/httpd/conf.d/ssl.conf
```

```
DocumentRoot "/usr/share/zabbix"
ServerName example.com:443
SSLCertificateFile /etc/httpd/ssl/apache-selfsigned.crt
SSLCertificateKeyFile /etc/httpd/ssl/private/apache-selfsigned.key
```

Restart the Apache service to apply the changes:

```
systemctl restart httpd.service
```

### Enabling Zabbix on root directory of URL

Add a virtual host to Apache configuration and set permanent redirect for document root to Zabbix

SSL URL. Do not forget to replace *example.com* with the actual name of the server.

```
/etc/httpd/conf/httpd.conf
```

```
#Add lines

<VirtualHost *:*>
    ServerName example.com
    Redirect permanent / http://example.com
</VirtualHost>
```

Restart the Apache service to apply the changes:

```
systemctl restart httpd.service
```

#### Enabling HTTP Strict Transport Security (HSTS) on web server

HSTS is enforced by Zabbix frontend in versions 4.0.0 - 4.0.2.

Starting with **4.0.3** to protect Zabbix frontend against protocol downgrade attacks, we recommend to enable HSTS policy on webserver.

For example, to enable HSTS policy for your Zabbix frontend in Apache configuration:

```
/etc/httpd/conf/httpd.conf
```

add the following directive to your virtual host's configuration:

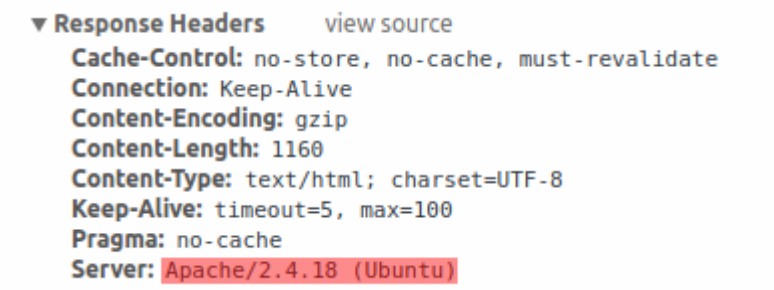
```
<VirtualHost *:443>
   Header set Strict-Transport-Security "max-age=31536000"
</VirtualHost>
```

Restart the Apache service to apply the changes:

```
systemctl restart httpd.service
```

#### Disabling web server information exposure

It is recommended to disable all web server signatures as part of the web server hardening process. The web server is exposing software signature by default:




The signature can be disabled by adding two lines to the Apache (used as an example) configuration file:

```
ServerSignature Off
ServerTokens Prod
```

PHP signature (X-Powered-By HTTP header) can be disabled by changing the php.ini configuration file (signature is disabled by default):

```
expose_php = Off
```

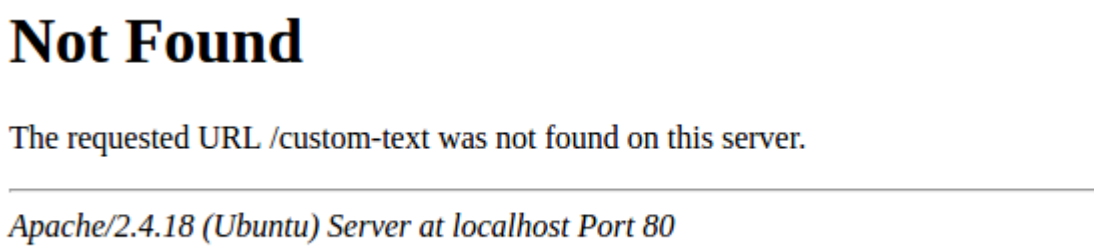
Web server restart is required for configuration file changes to be applied.

Additional security level can be achieved by using the mod_security (package libapache2-mod-security2) with Apache. mod_security allows to remove server signature instead of only removing version from server signature. Signature can be altered to any value by changing "SecServerSignature" to any desired value after installing mod_security.

Please refer to documentation of your web server to find help on how to remove/change software signatures.

#### Disabling default web server error pages

It is recommended to disable default error pages to avoid information exposure. Web server is using built-in error pages by default:



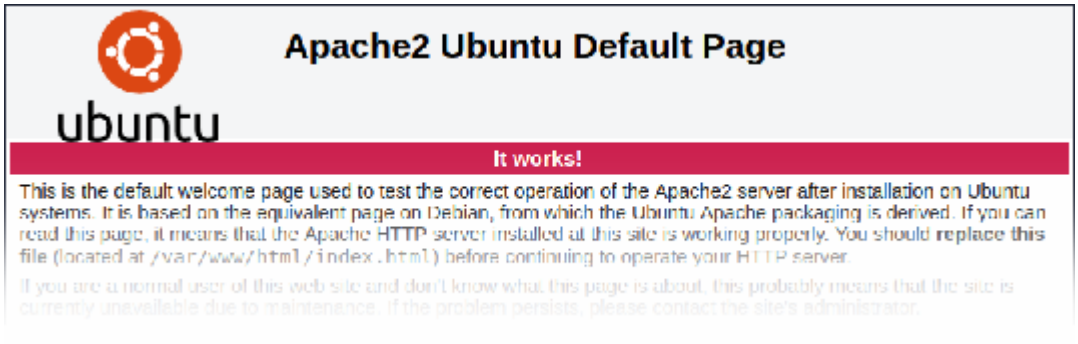

Default error pages should be replaced/removed as part of the web server hardening process. The "ErrorDocument" directive can be used to define a custom error page/text for Apache web server (used as an example).

Please refer to documentation of your web server to find help on how to replace/remove default error

pages.

### Removing web server test page

It is recommended to remove the web server test page to avoid information exposure. By default, web server webroot contains a test page called index.html (Apache2 on Ubuntu is used as an example):




The test page should be removed or should be made unavailable as part of the web server hardening process.

### Displaying URL content in the sandbox

Since version 4.0.22, some Zabbix frontend elements (for example, the URL widget) are preconfigured to sandbox content retrieved from the URL. It is recommended to keep all sandboxing restrictions enabled to ensure protection against XSS attacks.