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.

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.

Setting up SSL for Zabbix frontend

On RHEL/Centos, install mod_ssl package:

```
yum install mod_ssl
```

Create directory for SSL keys:

```
mkdir /etc/httpd/ssl
```

Add settings for SSL setup:

```
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) []:localhost
Email Address []:
```

Edit Apache SSL configuration:

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

```
DocumentRoot "/usr/share/zabbix"
ServerName localhost:443
SSLCertificateFile /etc/httpd/ssl/apache.crt
```

```
SSLCertificateKeyFile /etc/httpd/ssl/apache.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. Replace *localhost* with the actual name of the server.

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

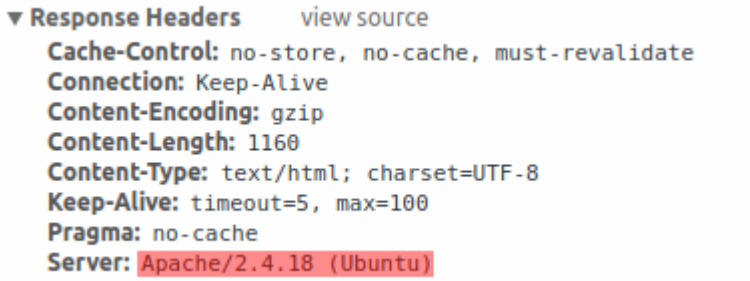
```
#Add lines

<VirtualHost *:*>
    ServerName localhost
    Redirect permanent / http://localhost
</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
```

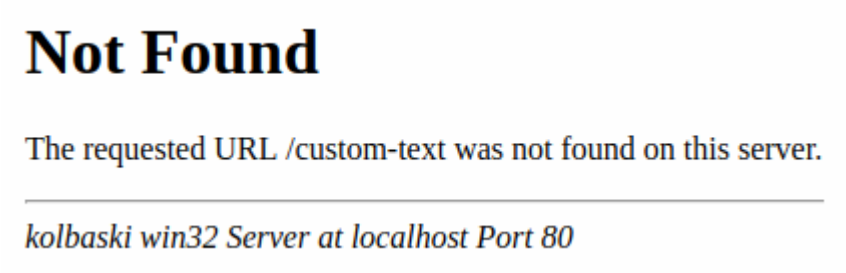
Web server restart is required for configuration file changes to be applied.

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:

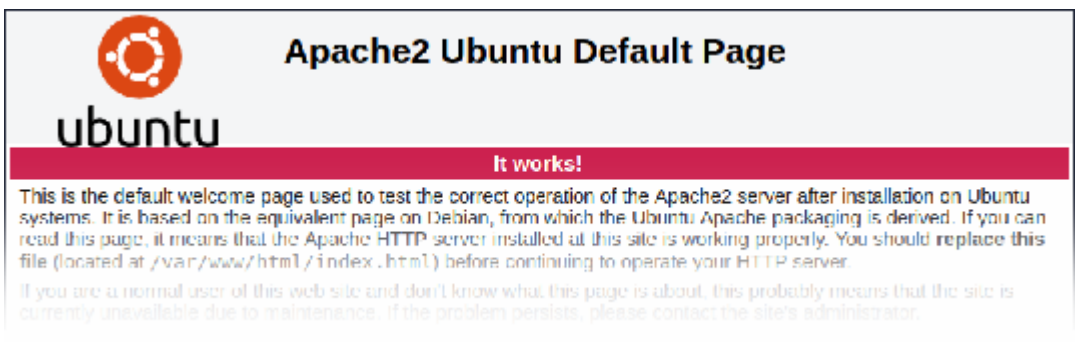


Default error pages should be replaced/removed as part of the web server hardening process. The "ErrorDocument" directive can be used to define a custom error page/text for Apache web server (used as an example).

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.