

Building Windows agent binaries with/without TLS

Overview

This section demonstrates how to build Windows agent binaries from sources with or without TLS.

Compiling OpenSSL

The following steps will help you to compile OpenSSL from sources on MS Windows 10 (64-bit).

1. For compiling OpenSSL you will need on Windows machine:
 1. C compiler (e.g. VS 2017 RC),
 2. NASM (<https://www.nasm.us/>),
 3. Perl (e.g. Strawberry Perl from <http://strawberryperl.com/>),
 4. Perl module Text::Template (cpan Text::Template).
2. Get OpenSSL sources from <https://www.openssl.org/>. OpenSSL 1.1.1 is used here.
3. Unpack OpenSSL sources, for example, in E:\openssl-1.1.1.
4. Open a commandline window e.g. the x64 Native Tools Command Prompt for VS 2017 RC.
5. Go to the OpenSSL source directory, e.g. E:\openssl-1.1.1.
 1. Verify that NASM can be found:

```
e:\openssl-1.1.1> nasm --version
NASM version 2.13.01 compiled on May  1 2017
```

6. Configure OpenSSL, for example:

```
e:\openssl-1.1.1> perl E:\openssl-1.1.1\Configure VC-WIN64A no-shared
no-capieng no-srp no-gost no-dgram no-dtls1-method no-dtls1_2-method -
-api=1.1.0 --prefix=C:\OpenSSL-Win64-111-static --
openssldir=C:\OpenSSL-Win64-111-static
```

- Note the option 'no-shared': if 'no-shared' is used then the OpenSSL static libraries libcrypto.lib and libssl.lib will be 'self-sufficient' and resulting Zabbix binaries will include OpenSSL in themselves, no need for external OpenSSL DLLs. Advantage: Zabbix binaries can be copied to other Windows machines without OpenSSL libraries. Disadvantage: when a new OpenSSL bugfix version is released, Zabbix agent needs to be recompiled and reinstalled.
- If 'no-shared' is not used, then the static libraries libcrypto.lib and libssl.lib will be using OpenSSL DLLs at runtime. Advantage: when a new OpenSSL bugfix version is released, probably you can upgrade only OpenSSL DLLs, without recompiling Zabbix agent. Disadvantage: copying Zabbix agent to another machine requires copying OpenSSL DLLs, too.

7. Compile OpenSSL, run tests, install:

```
e:\openssl-1.1.1> nmake
e:\openssl-1.1.1> nmake test
...
All tests successful.
Files=152, Tests=1152, 501 wallclock secs ( 0.67 usr + 0.61 sys =
```

```
1.28 CPU)
Result: PASS
e:\openssl-1.1.1> nmake install_sw
```

'install_sw' installs only software components (i.e. libraries, header files, but no documentation). If you want everything, use "nmake install".

Compiling Zabbix with OpenSSL

The following steps will help you to compile Zabbix from sources on MS Windows 10 (64-bit). When compiling Zabbix with/without TLS support the only significant difference is in step 4.

Support for OpenSSL 1.1.1 was added in 3.0.23, some additional fixes - in 3.0.24rc1.

1. On a Linux machine check out the source from SVN:

```
$ svn co svn://svn.zabbix.com/tags/3.0.24
$ cd 3.0.24/
$ ./bootstrap.sh
$ ./configure --enable-agent --enable-ipv6 --prefix=`pwd`
$ make dbschema
$ make dist
```

2. Copy and unpack the archive, e.g. zabbix-3.0.24.tar.gz, on a Windows machine.
3. Let's assume that sources are in e:\zabbix-3.0.24. Open a commandline window e.g. the x64 Native Tools Command Prompt for VS 2017 RC. Go to e:\3.0.24\build\win32\project.
4. Compile zabbix_get, zabbix_sender and zabbix_agent.
 - without TLS:

```
E:\zabbix-3.0.24\build\win32\project> nmake /K
```

- with TLS:

```
E:\zabbix-3.0.24\build\win32\project> nmake /K -f Makefile_get
TLS=openssl TLSINCDIR="C:\OpenSSL-Win64-111-static\include"
TSLIBDIR="C:\OpenSSL-Win64-111-static\lib"
E:\zabbix-3.0.24\build\win32\project> nmake /K -f Makefile_sender
TLS=openssl TLSINCDIR="C:\OpenSSL-Win64-111-static\include"
TSLIBDIR="C:\OpenSSL-Win64-111-static\lib"
E:\zabbix-3.0.24\build\win32\project> nmake /K -f Makefile_agent
TLS=openssl TLSINCDIR="C:\OpenSSL-Win64-111-static\include"
TSLIBDIR="C:\OpenSSL-Win64-111-static\lib"
```

5. New binaries are located in e:\zabbix-3.0.24\bin\win64. Since OpenSSL was compiled with 'no-shared' option, Zabbix binaries contain OpenSSL within themselves and can be copied to other machines that do not have OpenSSL.

Compiling Zabbix with LibreSSL

The process is similar to compiling with OpenSSL, but you need to make small changes in files located in the `build\win32\project` directory:

- In `Makefile_tls` delete `/DHAVE_OPENSSL_WITH_PSK`. i.e. find

```
CFLAGS = $(CFLAGS) /DHAVE_OPENSSL /DHAVE_OPENSSL_WITH_PSK
```

and replace it with

```
CFLAGS = $(CFLAGS) /DHAVE_OPENSSL
```

- In `Makefile_common.inc` add `/NODEFAULTLIB:LIBCMT` i.e. find

```
/MANIFESTUAC:"level='asInvoker' uiAccess='false'" /DYNAMICBASE:NO  
/PDB:$(TARGETDIR)\$(TARGETNAME).pdb
```

and replace it with

```
/MANIFESTUAC:"level='asInvoker' uiAccess='false'" /DYNAMICBASE:NO  
/PDB:$(TARGETDIR)\$(TARGETNAME).pdb /NODEFAULTLIB:LIBCMT
```

From:

<https://www.zabbix.com/documentation/3.0/> - **Zabbix Documentation 3.0**

Permanent link:

https://www.zabbix.com/documentation/3.0/manual/installation/install/win_agent

Last update: **2019/04/08 08:50**

