Setting up a build/testing/release environment

The instructions on building a release can be found in the build/README file in svn.

This page details how to set up an environment in which to build and test a release.

Because an environment that can build SpamAssassin and run all tests will contain everything that is needed to simply install and run SpamAssassin, this page may be useful to someone who wants to ensure that their system is ready for installing SpamAssassin. However, the setup described here does include packages that are optional in an installation of SpamAssassin as well as some that are required for building a release package.

An early version of this documentation described setting up a CentOS 7 build environment.
That information is included here for completeness, however it is recommended to not use CentOS 7 for future release builds. It is a very old system nearing the end of its very extended support life, that has purposefully not upgraded the versions of various packages it includes. Also, the information copied here does not include all the optional modules that SpamAssassin can use, which a build intended for official release should install and test.

The environment used for the most recent release builds has been Ubuntu 20.04. That is the setup that is described on this page.

Whatever OS you use, do not a system that is already running SpamAssassin. If building and testing a release package, there should be no system-wide SpamAssassin installed at all. Best practice is to use a virtual machine that is dedicated for the purpose of building and testing SpamAssassin. Do not use a public DNS resolver like Google's 8.8.8.8. They often have various blocks and filters that are intended to protect users, but may cause spurious failures on tests like dnsbl.t. Use the DNS server normally provided for the Internet connection on the network the computer is connected to.

CentOS 7 (deprecated for release building)

Installing these packages on a fresh CentOS 7 install will allow SpamAssassin to build and pass tests other than a few that require certain optional packages:

Packages that can be installed using sudo yum, including adding the EPEL repo:
  sudo yum install gpg zip perl*
  sudo yum remove perl-homedir
  sudo yum install epel-release
  sudo yum update
  sudo yum install perl-BerkeleyDB

Packages that are installed from source on CPAN using cpanm -S which installs as root
  cpanm -S Razor2::Client::Agent
  cpanm -S Net::Patricia
  cpanm -S Geo::IP
  cpanm -S Devel::SawAmpersand
  custom compiled re2c
    - obtain the latest release from github (linked from https://re2c.org) because epel rpm is too old
    - sudo yum install gcc-c++
    - build with ./configure && make && sudo make install

Ubuntu 20.04

Starting with a fresh installation Ubuntu 20.04 that was installed with the "minimal install" option:

Install the following packages using `sudo apt install`

```
build-essential
curl
cpanminus
git
pyzor
razor
subversion
libdb-dev
libdbi-dev
libidn11-dev
libidn2-dev
libmaxminddb-dev
libssl-dev
zlib1g-dev
poppler-utils
tesseract-ocr

libarchive-zip-perl
libberkeleydb-perl
libbsd-resource-perl
libdigest-sha-perl
libencode-detect-perl
libgeo-ip-perl
libgeoip2-perl
libio-compress-perl
libmail-dkim-perl
libmail-spf-perl
libnet-patricia-perl
libfile-sharedir-install-perl
libtest-exception-perl
libregexp-common-perl
libxml-libxml-perl
libtest-pod-coverage-perl
libdbd-sqlite2-perl
libdbd-sqlite3-perl
libdevel-cycle-perl
libgeography-countries-perl
libtest-perl-critic-perl
libdbix-simple-perl
libemail-mime-perl
libemail-sender-perl
libnet-idn-encode-perl
libtest-file-sharedir-perl
libtest-output-perl
libnet-imap-simple-perl
libnet-smtps-perl
libtext-diff-perl
```

Install the following packages from CPAN which do not have equivalent apt packages on Ubuntu already.
The -S option causes the installation step to be run as root using sudo.

```
cpanm -S Digest::SHA1 IP::Country::DB_File IP::Country Mail::DMARC::PurePerl Net::LibIDN2 \
 Perl::Critic::Policy::Bangs::ProhibitBitwiseOperators Perl::Critic::Policy::Perlsecret \
 Perl::Critic::Policy::Compatibility::ProhibitThreeArgumentOpen \
 Perl::Critic::Policy::Lax::ProhibitStringyEval::ExceptForRequire \
 Perl::Critic::Policy::ValuesAndExpressions::PreventSQLInjection \
 Perl::Critic::Policy::ControlStructures::ProhibitReturnInDoBlock
```

Install dcc from source: Get the source from the link at https://www.dcc-servers.net/dcc/ and

```
wget https://www.dcc-servers.net/dcc/source/dcc.tar.Z
tar xf dcc.tar.Z
cd dcc-*
./configure --disable-server --disable-dccm --disable-dccifd
make
sudo make install
```

custom compile re2c because the version installed by Ubuntu is too old
- obtain the latest release from github (linked from https://re2c.org)
- build with

```
./configure
make
sudo make install
```

Initialize the razor installation by running once (not as root) after razor is installed

```
razor-admin --create
razor-admin --register
```

Now that everything is installed, verify that you have a working build and test environment by downloading the trunk svn working directory, build it, and run tests.

```
mkdir ~/sabuild
cd ~/sabuild
svn co https://svn.apache.org/repos/asf/spamassassin/trunk
cd trunk
perl Makefile.PL < /dev/null
make ENABLE_SSL=yes
make test
```

That will verify that most things are set up ok, but to run the full set of tests that this setup should support try
`xt/run_release_test_suite.sh`

If you are creating an official SpamAssassin release, you should have access to the project's signing keys for the release package and for rule updates. The keys are given to you in a directory tree that you should put on the build machine, naming it ~/sabuildtools The build process looks for files in that specific named directory. You will need the appropriate password to use it at build time.

Other perl versions and sawampersand test

The test xt/20_sawampersand.t only applies to perl versions older than 5.17, which is a quite older than the one that is distributed with Ubuntu 20.04.

To be able to test with old versions of perl , install a package called plenv. If you are familiar with perlbrew and want to use it instead, both work equally well. I'm only going to bother documenting how to use plenv.

The following commands in a terminal shell will download and install plenv into your home directory, install perl 5.16.3, cpanm, and all the CPAN modules required for SpamAssassin building and testing.
After running this, log out of Ubuntu and log back in again to get the new environment with the perl 5.16.3, cd to the working directory and rebuild SpamAssassin starting with the perl Makefile.PL command.

With this old version of perl , the the sawampersand test can run. Read documentation about plenv to learn how to switch between versions of perl.

```
git clone https://github.com/tokuhirom/plenv.git ~/.plenv
git clone https://github.com/tokuhirom/Perl-Build.git ~/.plenv/plugins/perl-build/
echo 'export PATH="$HOME/.plenv/bin:$PATH"' >> ~/.profile
echo 'eval "$(plenv init -)"' >> ~/.profile
export PATH="$HOME/.plenv/bin:$PATH"
eval "$(plenv init -)"
plenv install 5.16.3
plenv rehash
plenv global 5.16.3
perl -v
plenv install-cpanm

cpanm Archive::Zip BSD::Resource BerkeleyDB Compress::Zlib DBI DB_File Devel::Cycle \
Digest::SHA Digest::SHA1 Email::Address::XS Encode::Detect Encode::Detect::Detector \
Geo::IP GeoIP2 GeoIP2::Database::Reader Geography::Countries HTML::Parser HTTP::Cookies \
HTTP::Daemon HTTP::Date HTTP::Negotiate IO::Socket::INET6 IO::Socket::SSL IO::String \
IP::Country IP::Country::DB_File LWP::Protocol::https LWP::UserAgent Mail::DKIM \
Mail::DMARC::PurePerl Math::Int128 MaxMind::DB::Reader::XS Net::CIDR::Lite Net::DNS \
Net::DNS::Nameserver Net::LibIDN Net::LibIDN2 Net::Patricia Net::Works::Network NetAddr::IP \
Params::Validate Razor2::Client::Agent Sys::Hostname::Long Test::Perl::Critic Test::Pod \
Test::Pod::Coverage WWW::RobotRules Text::Diff \
Perl::Critic::Policy::Bangs::ProhibitBitwiseOperators Perl::Critic::Policy::Perlsecret \
Perl::Critic::Policy::Compatibility::ProhibitThreeArgumentOpen \
Perl::Critic::Policy::Lax::ProhibitStringyEval::ExceptForRequire \
Perl::Critic::Policy::ValuesAndExpressions::PreventSQLInjection \
Perl::Critic::Policy::ControlStructures::ProhibitReturnInDoBlock

cpanm Mail::SPF -n --install-args="--install_path sbin=$HOME/bin"
```