SWAMP-in-a-Box Administrator Manual

Version 1.34, 2019-02-14 11:58:37 -0600

Table of Contents

1. Introduction

1.1. What Is SWAMP

The Software Assurance Marketplace (SWAMP) is a platform for running software assurance tools on your code. It is a joint effort of four research institutions — the Morgridge Institute for Research, Indiana University, the University of Illinois at Urbana-Champaign, and the University of Wisconsin-Madison — to advance the capabilities and increase the adoption of software assurance technologies through an open continuous assurance facility. The SWAMP originally went live in February 2014 as a web application at https://www.mir-swamp.org, where it provides continuous software assurance capabilities to developers and researchers.

The SWAMP project is funded by the Department of Homeland Security (DHS) Science and Technology Directorate, Homeland Security Advanced Research Projects Agency, Cyber Security Division (DHS S&T/HSARPA/CSD); BAA 11-02; and the Air Force Research Laboratory, Information Directorate under agreement number FA8750-12-2-0289.

1.2. What Is SWAMP-in-a-Box

For users that need or prefer to run software assurance tools on their own computing infrastructure, the SWAMP project offers a standalone software application called SWAMP-in-a-Box (SiB). It is, in essence, a local instance of the SWAMP that can be deployed on your own servers if you have higher security or compliance requirements for your software, or, being open-source, when you want to customize the software.

1.3. Obtaining SWAMP-in-a-Box

SWAMP-in-a-Box is currently available as an open beta. Visit https://github.com/mirswamp/deployment for instructions on how to download SWAMP-in-a-Box as a pre-packaged installer or as source code.

1.4. Documentation for SWAMP-in-a-Box

Copies of this SWAMP-in-a-Box Administrator Manual and the SWAMP-in-a-Box Reference Manual can be found in `/opt/swamp/doc` on the SWAMP-in-a-Box host after SWAMP-in-a-Box has been installed. Each manual is available as a single page HTML document and as a PDF.

Documentation for users of the SWAMP is available on the Help page of the SWAMP web application after SWAMP-in-a-Box has been installed.

2. System Requirements

SWAMP-in-a-Box is designed to be installed on a dedicated host, one that is not providing other

services — including Apache HTTP Server, MySQL/MariaDB, and HTCondor.

2.1. Hardware Requirements

Minimum:

- Cores: 4
- Memory: 16G
- Disk: 256G
- Support for KVM virtualization

Recommended:

- Cores: 8
- Memory: 64G
- Disk: 1T
- Support for KVM virtualization

The hardware requirements for SWAMP-in-a-Box are driven primarily by the number of simultaneous assessments and instances of the optional Code Dx results viewer you wish to be able to run at any given time. Each assessment and viewer instance is run in a virtual machine that requires:

- 2 cores
- 6G of memory.

The minimum requirements allow the host to run two virtual machines simultaneously while leaving resources available to run the web server and database that together provide the SWAMP web application to users.

Because SWAMP-in-a-Box uses virtual machines to run assessments, the host must support KVM virtualization. Modern x86-family processors provide support for KVM via Intel's VT-x or AMD's AMD-V extensions. On some systems, it might be necessary to enable Intel Virtual Technology extensions in the BIOS.

If you are installing SWAMP-in-a-Box in a virtual machine, the hypervisor must support and be configured for nested virtualization. For example, when using a VMware product as the hypervisor, enable settings such as "Expose hardware-assisted virtualization to the guest operating system" and "Virtualize Intel VT-x/EPT or AMD-V/RVI."

2.2. Supported Operating Systems

CentOS 6 and 7 are both supported. Other similar Linux distributions, such as Red Hat Enterprise Linux, might work but are untested.

2.3. Supported Disk Partitioning Schemes

As much space as possible should be allocated to the `/` partition without deleting or shrinking required system partitions, e.g., `/boot` and `swap`. For example, if there is a separate partition for `/home`, delete it, and allocate the space to the `/` partition.

2.4. Create a User Account with Full `sudo` Privileges

We recommend creating a normal user account with full `sudo` privileges so that the SWAMP-in-a-Box host can be administered without being logged in as `root` all the time. To create such an account:

1. Log in as `root`.
2. Create the new user account (replace `<username>` with the name of the new account):

   ```
   useradd <username>
   ```

3. Set the new account's password:

   ```
   passwd <username>
   ```

4. Run `visudo`, which will let you edit the `sudoers` file in the `vi` text editor. Find the line similar to

   ```
   root ALL=(ALL) ALL
   ```

   Add below it

   ```
   <username> ALL=(ALL) ALL
   ```

Whenever a task requires `root` access to the SWAMP-in-a-Box host, it can be run while logged in as the user created above by prefixing the relevant commands with `sudo`. For example, to use the `vi` text editor to edit `/opt/swamp/etc/swamp.conf` as `root`, run the following command:

```
sudo vi /opt/swamp/etc/swamp.conf
```

2.5. Disable SELinux

SWAMP-in-a-Box will not install or function correctly when SELinux is in `enforcing` mode, in part because the various software packages that SWAMP-in-a-Box relies on do not all support SELinux.

To disable SELinux, as `root` (or using `sudo`), edit `/etc/selinux/config` by changing the line `SELINUX=enforcing` to `SELINUX=disabled`. Then reboot the host.

2.6. Configure Firewalls

With regards to network traffic, the SWAMP-in-a-Box host is expected to:

- Respond to incoming HTTPS (port 443) network traffic, because it is required to access the SWAMP web application and for the web application to function correctly.
- Potentially generate outgoing traffic while performing an assessment of a package, typically using HTTP, HTTPS, FTP, FTPS, SSH, and rsync. Traffic can include:
  - Updating of the assessment platform's currently installed set of packages
  - Downloading of user-specified dependencies for the package being assessed
  - Contacting license servers
  - Other traffic generated by the package's build system.

  Each assessment is run in a virtual machine that is managed by `libvirtd`. By default, SWAMP-in-a-Box configures `libvirtd` to assign each virtual machine an IP address in the range 192.168.123.2 through 192.168.123.254, and to use network address translation (NAT) to contact external hosts.

Any firewalls protecting the SWAMP-in-a-Box host must be configured to allow the above network traffic. The SWAMP-in-a-Box installer will not modify the host's firewall configuration.

Restart the `libvirtd` service on the host whenever the host's firewall configuration is modified. To do so, as `root` (or using `sudo`), run the following command:

```
service libvirtd restart
```

This is necessary because the `libvirtd` service modifies the host's firewall configuration in order to allow the virtual machines started by it to access the host's network, but it does not make its configuration changes permanent.

Example 1. Allowing Incoming HTTPS and SSH Traffic with `iptables`

For systems that use `iptables`, such as CentOS 6 by default, a sample configuration file can be found in the `config_templates` sub-directory of the SWAMP-in-a-Box installer (referred to below as `<installer-dir>`). Copy the `iptables` file from that directory to `/etc/sysconfig`. Then restart the `iptables` and `libvirtd` services. For example, as `root` (or using `sudo`), run the following commands:

```
cp <installer-dir>/config_templates/iptables /etc/sysconfig
service iptables restart
service libvirtd restart
```

Example 2. Allowing Incoming HTTPS and SSH Traffic with `firewalld`

For systems that use `firewalld`, such as CentOS 7 by default, use `firewall-cmd` to permanently allow HTTPS and SSH traffic. Then restart the `firewalld` and `libvirtd` services. For example, as `root` (or using `sudo`), run the following commands:

```
firewall-cmd --zone=public --permanent --add-service=https
firewall-cmd --zone=public --permanent --add-service=ssh
systemctl restart firewalld
systemctl restart libvirtd
```

2.7. Other Considerations

The SWAMP-in-a-Box install and upgrade process configures only those aspects of the host that are directly involved in ensuring that the SWAMP functions correctly. Other aspects are the responsibility of the host's system administrator. For example, consider looking at the Applied Crypto Hardening guide at https://bettercrypto.org for suggestions on how to configure the SSH server and other cryptographic tools on the host.

3. Installing and Upgrading SWAMP-in-a-Box

3.1. Before You Begin

3.1.1. Before Installing a New SWAMP-in-a-Box

- You will need `root` access to the SWAMP-in-a-Box host.
- The install script will prompt for the DNS hostname to use for the host. It must match the hostname that users will use to access the SWAMP web application and the hostname on the SSL certificate

for the host's web server.

- The install script will prompt for the initial values to use for the following passwords, which can then be used to access the SWAMP web application and its SQL database:
  - Database `root` password: This is the password for the SWAMP SQL database's `root` user. It may be different from the host operating system's `root` user's password, because the database maintains its own, separate collection of user accounts for accessing it.

    

    Do not forget this password. It is required to upgrade SWAMP-in-a-Box and reset the passwords below.

  - Database web password: This is the password used by the SWAMP web application's backend to connect to the SQL database.
  - Database SWAMP services password: This is the password used by SWAMP-in-a-Box's system daemons and backend processes to connect to the SQL database.
  - SWAMP administrator account password: This is the password for the SWAMP web application's `admin-s` account, which is created during the install process and can be used to administer the SWAMP.

3.1.2. Before Upgrading an Existing SWAMP-in-a-Box

- You will need `root` access to the SWAMP-in-a-Box host.
- You will need `root` access to the SWAMP-in-a-Box database.
- The SWAMP-in-a-Box host must currently have version 1.29 or later of SWAMP-in-a-Box installed. Upgrades from earlier versions are not supported and will likely result in a non-working system. Older systems should first be upgraded to 1.29 or 1.30, before upgrading them to a more recent version.

Back up any customizations. The SWAMP-in-a-Box upgrade process will overwrite existing SWAMP files. Add on Tools, Platforms, and Viewers will be retained, as will customizations made to SWAMP-in-a-Box via configuration files (.env, config.json, swamp.conf, and services.conf). Any other customizations you have made by modifying SWAMP files will need to be reimplemented after an upgrade.

3.2. Run `yum update`

We recommend running `yum update` (as `root` or using `sudo`) to ensure that any software currently installed on the SWAMP-in-a-Box host is up-to-date. This is especially important when there has been a new release of the host's operating system since the host was initially set up. In this case, the steps below will likely cause a partial update to the new release, which might leave the host in an inconsistent and non-working state.

3.3. Obtain the Installer

Visit https://github.com/mirswamp/deployment for instructions on how to download SWAMP-in-a-Box as a pre-packaged installer, which is what the instructions below assume you are working with.

3.4. Extract the Installer

On the SWAMP-in-a-Box host, move or copy the following files into the same directory (for example, a user's home directory):

- `extract-installer.bash`
- `swampinabox-<version>-installer.tar.gz`
- `swampinabox-<version>-platforms.tar.gz`
- `swampinabox-<version>-tools.tar.gz`

From that directory, run `extract-installer.bash`:

```
bash extract-installer.bash
```

When the script completes successfully, it will display the location of the SWAMP-in-a-Box installer. The instructions below use `<installer-dir>` to refer to that directory.

3.5. Install/Upgrade Dependencies

The directory `<installer-dir>/repos` contains set up scripts that will

- configure package repositories,
- install dependencies,
- enable required services, and
- create required user accounts.

Even if you have gone through this step on the SWAMP-in-a-Box host for a previous release of SWAMP-in-a-Box, it is important to run the scripts for the current release as they will ensure that the correct versions of SWAMP-in-a-Box's dependencies are installed.

If your host has unrestricted access to the internet, as `root` (or using `sudo`), run the `install-all.bash` script:

```
<installer-dir>/repos/install-all.bash
```

If your host has restricted access to the internet, or if you run into issues with running `install-all.bash`, refer to the appendix on installing SWAMP-in-a-Box's dependencies. This appendix lists the

dependencies in detail, so that you can determine how best to install them, and provides other troubleshooting guidance. Continue with the steps below after the dependencies have been installed.

3.6. Run the Main Install/Upgrade Script

As `root` (or using `sudo`):

- If you are installing a new SWAMP-in-a-Box, run the following script:

```
<installer-dir>/bin/install_swampinabox.bash
```

- If you are upgrading an existing SWAMP-in-a-Box, run the following script:

```
<installer-dir>/bin/upgrade_swampinabox.bash
```

You will be prompted for the passwords and other information listed above. Output will be saved to a log file in `<installer-dir>/log`, a copy of which can be found in `/opt/swamp/log`. If the install or upgrade process is unsuccessful, the log file will be helpful in determining the cause.

At the end of the install or upgrade process, the script will check for and warn about many common problems (refer to the section on checking the host's health for details). When installing a new SWAMP-in-a-Box, you might see a warning that the host does not appear to have a valid SSL certificate because you have not yet configured an SSL certificate. If this is the only warning, then the SWAMP should function correctly, though users might have to click through a warning in their browser stating that the web site is insecure.

When upgrading an existing SWAMP-in-a-Box, the upgrade script will use `mysqldump` to create a backup of the SWAMP's databases before making any modifications to them. The database dumps will be stored in the following files, which can be found in the directory from which you run the upgrade:

- `bkup_all_databases.<YYYY_MM_DD>.sql`
- `bkup_information_schema.<YYYY_MM_DD>.sql`

3.7. Verify that the Install/Upgrade Was Successful

1. In a web browser, navigate to https://<SWAMP-in-a-Box-hostname>/.
2. Sign in to the SWAMP with the administrator account (username: `admin-s`).
3. Upload a package, create and run a new assessment of it, and view the results. Several small, sample packages known to work with the SWAMP can be found in `<installer-dir>/sample_packages`. The `README.txt` file in that directory provides basic information about the samples.

3.8. Check for Updates

After upgrading an existing SWAMP-in-a-Box, refer to the section on checking for updates to determine whether there are components that still need to be upgraded. The SWAMP-in-a-Box upgrade script does not necessarily upgrade all components of SWAMP-in-a-Box for which a newer version might be available.

4. Installing Additional Components

4.1. Installing Additional Platforms

The SWAMP-in-a-Box installer includes and installs only the Ubuntu 16.04 platform for performing assessments. For most package types, this is the only supported platform. For C/C++ packages, additional platforms are available, including releases of CentOS, Debian, Fedora, Scientific Linux, and older releases of Ubuntu.

4.1.1. Before You Begin

- You will need `root` access to the SWAMP-in-a-Box host.
- You will need `root` access to the SWAMP-in-a-Box database.

4.1.2. Procedure

1. Visit https://platform.swampinabox.org/platform-images/.
2. Download and copy to the SWAMP-in-a-Box host the `.qcow2.gz` files corresponding to the additional platforms you wish to perform assessments on. The naming scheme for these files is as follows:

   ```
   condor-<Linux distribution>-<version>-<32 or 64 bit>-master-<YYYYMMDD>.qcow2.gz
   ```

   When downloading the files, name the copies exactly as shown on platform.swampinabox.org. Otherwise, they will not be recognized as supported platforms in the next step.

3. On the SWAMP-in-a-Box host, for each file, as `root` (or using `sudo`), run the `install_platform` script, providing the path to the `.qcow2.gz` file:

   ```
   /opt/swamp/bin/install_platform <path to .qcow2.gz file>
   ```

   When prompted, provide the password for the database's `root` user, which is needed to add the platform to the database and make it available in the SWAMP. Note that `install_platform` will likely take several minutes to complete due to the size of the file.

4.2. Enabling Android Assessments

You can enable SWAMP-in-a-Box to assess Android Java Source and Android .APK packages.

4.2.1. Before You Begin

- You will need `root` access to the SWAMP-in-a-Box host.
- You will need `root` access to the SWAMP-in-a-Box database.
- You will need to make sure you have enough available hard drive space on your SWAMP-in-a-Box host to download and decompress the Android Ubuntu platform image.

The Android Ubuntu platform includes the OS dependencies needed to build and assess Android Packages. As such it is quite large. The downloadable .qcow2.gz file needed to add the platform is over 28 GB. The decompressed, installed .qcow2 file takes up about 175 GB in the /swamp/platforms/images directory.

4.2.2. Procedure

1. Download and install the Android Ubuntu 12.04 64-bit platform following the procedure for Installing Additional Platforms.
2. Optionally delete the Android Ubuntu platform .qcow2.gz file if you wish to conserve disk space.

When the Android Ubuntu platform is added, the Android Java Source and Android .APK package types are enabled. Android-specific tools are installed with SWAMP-in-a-Box, but they cannot be used for assessments until the Android Ubuntu platform is added.

4.3. Installing Additional Tools

The SWAMP-in-a-Box installer includes and installs a variety of tools for assessing packages.

For C/C++ packages, three additional tools can be installed:

- CodeSonar, and
- Parasoft C/C++test.

For Java packages, two additional tools can be installed:

- OWASP Dependency Check and
- Parasoft Jtest.

For Web Scripting packages, one additional tool can be installed:

- A version of Retire.js that does not require internet access (the version included with SWAMP-in-a-Box will not function correctly without internet access)

The process for obtaining tool installers/archives from their vendors and packaging them into the format that the SWAMP requires differs *significantly* between the tools. However, once that is done, the process for installing and configuring the tools for use in the SWAMP is largely similar.

4.3.1. Before You Begin

- You will need `root` access to the SWAMP-in-a-Box host.
- You will need `root` access to the SWAMP-in-a-Box database.

4.3.2. Obtain the Tool Installer/Archive from Its Vendor

CodeSonar

Parasoft C/C++test

Parasoft Jtest

Refer to the appendix on obtaining add ons for information on how to contact each tool's vendor. Obtain one or both of the 32-bit and 64-bit Linux archives for the tool you wish to install. For Parasoft C/C++test and Parasoft Jtest, only versions 10.3.0 and later are supported.

In addition, follow the vendor's instructions on setting up a license server. Ensure that the SWAMP-in-a-Box host is able to contact the license server on the required ports.

The archives for CodeSonar should be named:

```
codesonar-<version>.<YYYYMMDD>-i686-pc-linux.tar.gz    (32-bit)
codesonar-<version>.<YYYYMMDD>-x86_64-pc-linux.tar.gz  (64-bit)
```

The archives for Parasoft C/C++test should be named:

```
parasoft_cpptest_engine_<version>_linux.tar.gz         (32-bit)
parasoft_cpptest_engine_<version>_linux_x86_64.tar.gz  (64-bit)
```

The archives for Parasoft Jtest should be named:

```
parasoft_jtest_<version>_linux_x86.tar.gz     (32-bit)
parasoft_jtest_<version>_linux_x86_64.tar.gz  (64-bit)
```

OWASP Dependency Check

Visit https://platform.swampinabox.org/tool-archives/ and download the `dependency-check-<version>.tar.gz` file corresponding to the version of OWASP Dependency Check that you would like to install. The archive from platform.swampinabox.org includes scripts and documentation for integrating the tool into the SWAMP.

Retire.js

On the SWAMP-in-a-Box host, copy the `/swamp/store/SCATools/retire-js-<version>.tar.gz` file corresponding to the version of Retire.js that you would like to install to some directory that you have write access to (for example, your home directory).

4.3.3. Create the SWAMP Tool Archive

CodeSonar

Parasoft C/C++test

Parasoft Jtest

Run the `make_swamp_tool` script to package the vendor's installers into the archive format that the SWAMP uses.

```
/opt/swamp/bin/make_swamp_tool \
    --tool-name <gt-csonar or ps-ctest or ps-jtest> \
    --tool-version <version> \
    --installer-linux32 <path to 32-bit archive> \
    --installer-linux64 <path to 64-bit archive>
```

Specify `gt-csonar`, `ps-ctest`, or `ps-jtest` for the `--tool-name` option, depending on the tool being packaged. Omit the `--installer-linux32` option if you have only the 64-bit installer, and similarly for the `--installer-linux64` option.

When `make_swamp_tool` completes, the output should include the path to the SWAMP tool archive file that was created. Note that `make_swamp_tool` will likely take several minutes to complete due to the size of the installers.

OWASP Dependency Check

Retire.js

Expand the archive file that you downloaded or copied:

```
tar zxvf <tool-name-and-version>.tar.gz
```

This should create a directory `<tool-name-and-version>`. Inside the directory will be `README` files in various formats. Follow the directions in the `README` for creating the SWAMP tool archive. Make note of whether you will need to add additional entries to `services.conf` or configure assessments to run without internet access (i.e., "internet-inaccessible" assessments).

4.3.4. Install the Tool

The `install_tool` script is used to install and manage any tools that are added onto a SWAMP-in-a-Box installation. The script must always be invoked with the following command line arguments:

- `--tool <id>`: This specifies the tool being managed. Recognized values for `<id>` include:
  - `dependency-check`: OWASP Dependency Check
  - `gt-csonar`: GrammaTech CodeSonar
  - `ps-ctest`: Parasoft C/C++test
  - `ps-jtest`: Parasoft Jtest
  - `retire-js`: Retire.js
- `--add`, `--configure`, `--remove`, or `--replace`: This determines the "mode" that the script will run in, i.e., whether to add a new version of the tool, remove an existing version of the tool, replace an existing version of the tool, or configure the tool. Depending on the mode selected, other command line arguments will be required.

To add a new version of a tool to the swamp, as `root` (or using `sudo`), run `install_tool` as follows:

```
/opt/swamp/bin/install_tool \
    --tool <id> \
    --add \
    --tool-version <version> \
    --tool-archive <path to the SWAMP tool archive file>
```

For some tools, notably Parasoft C/C++test and Parasoft Jtest, you might need to pass a different version string to `install_tool` than the one for `make_swamp_tool` in order to match the filename produced by `make_swamp_tool`. For example, compared to the version string for `make_swamp_tool`, you might need to append `-2` or `-12`.

Note that `install_tool` will likely take several minutes to complete due to the size of the tool archive.

4.3.5. Configure the Tool

As `root` (or using `sudo`), run the `install_tool` script with the `--configure` option, providing additional options as needed.

CodeSonar

Parasoft C/C++test

Parasoft Jtest

Use the `--license-server-host` and `--license-server-port` options to specify the hostname of the license server to use and the port on which to contact the license server:

```
/opt/swamp/bin/install_tool \
    --tool <id> \
    --configure \
    --license-server-host <hostname of the license server> \
    --license-server-port <port number>
```

For Parasoft C/C++test and Parasoft Jtest, the script will prompt for the username and password to use for contacting the license server.

In addition, use the `--limit` option to specify how many simultaneous instances of the tool the SWAMP may run:

```
/opt/swamp/bin/install_tool \
    --tool <id> \
    --configure \
    --limit <max number of simultaneous instances>
```

OWASP Dependency Check

Use the `--tool-conf` option to specify the path to a file containing additional entries that should be added to `services.conf`:

```
/opt/swamp/bin/install_tool \
    --tool <id> \
    --configure \
    --tool-conf <path to file containing entries for services.conf>
```

4.3.6. Manage the Installed Versions of the Tool

- Additional versions of each tool can be installed using the directions above. In the SWAMP web application, the "latest" version of the tool will be whichever version was **installed** most recently. We recommend installing multiple versions in order (e.g., 1.0, 2.0, 2.1, 3.0, etc.) so that the "latest" version matches users' expectations.
- If you wish to remove a version of a tool from the SWAMP, as `root` (or using `sudo`), run the `install_tool` script with the `--remove` option, providing the version to remove:

```
/opt/swamp/bin/install_tool \
    --tool <id> \
    --remove \
    --tool-version <version>
```

- If you have previously installed, say, only the 32-bit version of a tool and now wish to make both

the 32-bit and 64-bit versions available, first create the SWAMP tool archive, as above, to package the 32-bit and 64-bit archives together for the SWAMP. Then as `root` (or using `sudo`), run the `install_tool` script with the `--replace` option:

```
/opt/swamp/bin/install_tool \
    --tool <id> \
    --replace \
    --tool-version <version> \
    --tool-archive <path to SWAMP tool archive file>
```

4.4. Installing Additional Viewers

The SWAMP-in-a-Box installer includes and installs only a "native" viewer for viewing the results of an assessment. The SWAMP web application also provides a link to download the raw results of an assessment as a SCARF XML file.

For users with needs that are not met by either of these options, it is possible to install a SWAMP-specific version of Code Dx for viewing results. Refer to the appendix on Code Dx for information about obtaining this SWAMP-specific version of Code Dx from Code Dx, Inc. (SWAMP-in-a-Box currently does **not** support integrating with an existing, standalone Code Dx installation.)

4.4.1. Before You Begin

- You will need `root` access to the SWAMP-in-a-Box host.
- You will need `root` access to the SWAMP-in-a-Box database.
- You will need to obtain the `.war` file for the SWAMP-specific version of Code Dx from Code Dx, Inc. If you are provided with a `.zip` file or some other archive format, first expand the archive and locate the `.war` file within the expanded contents.

4.4.2. Procedure

1. Copy the Code Dx `.war` file obtained from Code Dx, Inc. to the SWAMP-in-a-Box host.
2. On the SWAMP-in-a-Box host, as `root` (or using `sudo`), run the `install_codedx` script, providing the path to the `.war` file:

   ```
   /opt/swamp/bin/install_codedx <path to Code Dx .war file>
   ```

   When prompted, provide the password for the database's `root` user, which is needed to add the Code Dx viewer to the database and make it available in the SWAMP.

5. Configuring SWAMP-in-a-Box

For additional information on the configuration options discussed below, see the SWAMP-in-a-Box Reference Manual.

5.1. Configuring Assessments to Run Without Internet Access

By default, when an assessment is performed, the platform will first attempt to update its collection of installed packages. This step will fail when the SWAMP-in-a-Box host's access to the internet is limited, which will in turn cause the assessment as a whole to fail. For such hosts, it is possible to configure SWAMP-in-a-Box such that platforms skip this step.

This configuration will **not** make a difference if the package being assessed specifies additional dependencies or if it uses a build system or script that requires access to the internet. If the assessment framework cannot download and install the additional dependencies, or if the build fails due to not being able to access resources on the internet, the assessment will still fail.

5.1.1. Before You Begin

- You will need `root` access to the SWAMP-in-a-Box host.

5.1.2. Procedure

Modify `/opt/swamp/etc/swamp.conf` such that the line

```
SWAMP-in-a-Box.internet-inaccessible = false
```

reads instead as

```
SWAMP-in-a-Box.internet-inaccessible = true
```

Any assessments submitted after making this change should no longer fail due to not having access to the internet, subject to the caveats noted above.

5.2. Configuring an SSL Certificate for SWAMP-in-a-Box

A self-signed certification is included by default when `httpd` and `mod_ssl` are installed for SWAMP-in-a-Box. Most web browsers will flag your SWAMP-in-a-Box website as insecure when using the self-signed certification. This section provides instructions for configuring SWAMP-in-a-Box to use an SSL

certificate signed by a trusted certificate authority.

Below, the fully qualified domain name (FQDN) needs to correspond to the main URL for your SWAMP-in-a-Box website, for example `https://sib.example.org`.

5.2.1. Acquire the SSL Certificate

The first step is to acquire an SSL certificate matching your SWAMP-in-a-Box domain name from a trusted certificate authority (CA). For the example above, the SSL certificate would match `sib.example.org`.

1. Generate a private key without a passphrase. For the example domain name used above, the command would be:

```
openssl genres -des3 -out sib.example.org.private.key
```

2. Create your CSR. For the example domain name used above, the command would be:

```
openssl req -new -key sib.example.org.private.key -out sib.example.org.csr
```

3. Purchase the SSL certificate by submitting your CSR. The vendor will send you the signed SSL certificate and any required intermediate certificates.

5.2.2. Install the SSL Certificate

The second step is to install the certificate on your SWAMP-in-a-Box and configure it for use with Apache (`httpd`).

1. Copy the certificates, along with the private key, to the SWAMP-in-a-Box host, typically in `/etc/pki/tls/certs` and `/etc/pki/tls/private`.
2. Make the private key readable only by `root`.
3. Make the certificates readable by the web server (i.e., world readable).
4. Modify `/etc/httpd/conf.d/ssl.conf`.

   Set the path to your certificate and private key (based on the example domain used above):

```
SSLCertificateFile /etc/pki/tls/certs/sib.example.org.cert
SSLCertificateKeyFile /etc/pki/tls/private/sib.example.org.private.key
```

   Depending on the specific SSL certificate, you may also need to set the path to the following files:

```
SSLCertificateChainFile
SSLCACertificateFile
```

Set the following parameters as shown:

```
SSLProtocol all -SSLv2 -SSLv3
SSLCipherSuite
EDH+CAMELLIA:EDH+aRSA:EECDH+aRSA+AESGCM:EECDH+aRSA+SHA256:EECDH:+CAMELLIA128:+AES128:+
SSLv3:!aNULL:!eNULL:!LOW:!3DES:!MD5:!EXP:!PSK:!DSS:!RC4:!SEED:!IDEA:!ECDSA:kEDH:CAMELL
IA128-SHA:AES128-SHA
SSLHonorCipherOrder On
```

5.2.3. Restart the `httpd` Service

The third step is to restart Apache (`httpd`).

Run the following command to verify that there are no syntax errors in Apache's configuration files.

```
apachectl -t
```

Fix any issues that are reported.

Then as `root` (or using `sudo`), restart the `httpd` service:

```
service httpd restart
```

5.3. Configuring Outgoing Email for SWAMP-in-a-Box

Enabling outgoing email allows the SWAMP to send email notifications to users. The following functionality is enabled when outgoing email is enabled:

- New user accounts are "pending" until email address is verified.
- Users can edit email addresses. Changes take place once verified.
- User email addresses are displayed throughout the user interface.
- Users can request a password reset through an email link.
- Users can request an email indicating the username associated with an email address.
- Permission requests, project invitations, and SWAMP admin invitations are handled through email notifications in addition to the notification system in the SWAMP UI.
- Users can opt to receive an email on completion of an assessment.

- SWAMP Administrators can configure Restricted Domains for email addresses.
- SWAMP Administrators can send system emails to one or more SWAMP users.
- SWAMP Administrators can flag users to force a password reset the next time they sign in.
- SWAMP Administrators can flag inactive users as hibernated. This forces a password reset the next time the user signs in.
- Contact Us and Report Security Incident pages (if enabled) may include a means to submit a message directly through the web interface.
- Emails are sent to notify users of events such as removal from project membership and disabling and re-enabling of projects and user accounts.

5.3.1. Before You Begin

- You need `root` access to the SWAMP-in-a-Box host.
- You need an SMTP server that you are authorized to relay mail through.

5.3.2. Modify `/etc/postfix/main.cf` to Use Your SMTP Server

- Set the `relayhost` attribute to your SMTP server.
- Restart the postfix service by running the following command as `root` (or using `sudo`):

```
service postfix restart
```

5.3.3. Modify `/var/www/swamp-web-server/.env` to Enable Outgoing SWAMP Email

Any values that include spaces must be enclosed in double-quotes. Any values that do *not* include spaces must *not* be enclosed in double-quotes.

- Set `MAIL_ENABLED` to "true".
- Set `MAIL_DRIVER` to "sendmail".
- Set the `MAIL_FROM_ADDRESS` and `MAIL_FROM_NAME` to the email address and name you want to use as the sender of outgoing SWAMP emails.

5.3.4. Enable "Contact Us" for SWAMP-in-a-Box

Enabling "Contact Us" creates a Contact link in the SWAMP menu bar. This link provides access to the "Contact Us" page, which displays general contact information. If email is enabled, this page can also be configured to provide a form for users to submit a contact/support message.

Step 1: Modify `/var/www/html/config/config.json` to enable the "Contact Us" page and set display parameters.

- Add a `contact` array containing a `support` array.
- Add `email`, `phoneNumber`, `description`, and `message`, values to the `support` array.

Note:

- The `config.json` file defines parameters within JSON arrays. Therefore, it is important to maintain the array format when editing, adding, or removing parameters in this file.

Sample:

```
"contact": {
  "support": {
    "description": "Support staff",
    "email": "<Support email address (optional)>",
    "message": "Feel free to contact us with questions.",
    "phoneNumber": "<Support phone number (optional)>"
  }
},
```

Step 2: Optionally modify `/var/www/swamp-web-server/.env` to enable a web form and to configure contact message recipients. This is only applicable when outgoing email is enabled.

Any values that include spaces must be enclosed in double-quotes. Any values that do *not* include spaces must *not* be enclosed in double-quotes.

- Set `APP_CONTACT_FORM` to "true"
- Set `MAIL_CONTACT_ADDRESS` to the email address of the recipient of "Contact Us" messages.
- Set `MAIL_CONTACT_NAME` to the name of the recipient of "Contact Us" messages.

5.3.5. Enable "Report Security Incident" for SWAMP-in-a-Box

Enabling "Report Security Incident" creates a Security link on the SWAMP Contact Us page. This link provides access to the "Report Security Incident" page, which displays information about reporting a security incident. If email is enabled, this page can also be configured to provide a form for users to submit a security incident report.

You must have already enabled the "Contact Us" page (see above).

Step 1: Modify `/var/www/html/config/config.json` to enable the "Report Security Incident" page and set display parameters.

- Add a `security` array to the `contact` array (see sample).
- Add `email`, `phoneNumber`, `description`, and `message`, values to the "security" array (see sample).

Note:

- The `config.json` file defines parameters within JSON arrays. Therefore, it is important to maintain the array format when editing, adding, or removing parameters in this file.

Sample:

```
"contact": {
  "support": {
    "description": "Support staff",
    "email": "<Support email address (optional)>",
    "message": "Feel free to contact us with questions.",
    "phoneNumber": "<Support phone number (optional)>"
  },
  "security": {
    "description": "Security team",
    "email": "<Security email address (optional)>",
    "message": "<Security message here (optional)>",
    "phoneNumber": "<Security phone number (optional)>"
  }
},
```

Step 2: Optionally modify `/var/www/swamp-web-server/.env` to enable a web form and to configure security incident message recipients. This is only applicable when outgoing email is enabled.

Any values that include spaces must be enclosed in double-quotes. Any values that do *not* include spaces must *not* be enclosed in double-quotes.

- Set `APP_CONTACT_FORM` to "true"
- Set `MAIL_SECURITY_ADDRESS` to the email address of the recipient of "Report Security Incident" messages.
- Set `MAIL_SECURITY_NAME` to the name of the recipient of "Report Security Incident" messages.

5.4. Configuring LDAP for User Authentication and Attributes

In a basic installation of SWAMP-in-a-Box, user information for the SWAMP is stored in the `project` database in the `user` table with the following information (attributes):

- SWAMP user UID
- username
- password (encrypted using BCRYPT)

- first name
- last name
- full name
- email
- affiliation

It is possible to configure the SWAMP to use a local LDAP or Active Directory (AD) server — assuming Active Directory has been configured to act as an LDAP server, as is the default — to store user records and their attributes.

You can configure your SWAMP-in-a-Box to access user accounts in the LDAP/AD server in one of two ways: with read-only access or with the ability to create and edit records in the LDAP/AD server.

You would configure your SWAMP-in-a-Box with read-only access to an LDAP/AD server when the LDAP/AD server is managed by processes external to the SWAMP. Your SWAMP-in-a-Box may then be one of multiple clients of the LDAP/AD server. In this case, the SWAMP does not provide workflows to create user records or edit any of the user attributes described above.

If you configure SWAMP-in-a-Box with the ability to create and edit user records in the LDAP/AD server, it is assumed that the SWAMP is the primary, if not only, client of the LDAP/AD server. In this case, the SWAMP provides the same workflows for creating and editing user records that it does when it is not configured with an LDAP/AD server. The only difference is that the user records and attributes described above are stored in the LDAP/AD server instead of in the `user` table in the SWAMP's local database.

5.4.1. Configuring LDAP Options in the Web Backend Configuration File

Before You Begin

- You will need `root` access to the SWAMP-in-a-Box host.
- Consult the SWAMP-in-a-Box Reference Manual for detailed descriptions of the parameters discussed below.

Procedure

1. As `root` (or using `sudo`), edit the web backend configuration file:

   ```
   vi /var/www/swamp-web-server/.env
   ```

2. Set the following parameters to enable LDAP and configure whether LDAP is read-only:

```
LDAP_ENABLED
LDAP_READ_ONLY
```

3. Set the following parameters to determine how user passwords are validated:

```
APP_PASSWORD_ENCRYPTION_METHOD
LDAP_PASSWORD_VALIDATION
```

4. Set the following parameters to identify your LDAP/AD server and provide SWAMP-in-a-Box access to it:

```
LDAP_HOST
LDAP_PORT
LDAP_WEB_USER
LDAP_WEB_USER_PASSWORD
LDAP_PASSWORD_SET_USER (only if LDAP_READ_ONLY=false)
LDAP_PASSWORD_SET_USER_PASSWORD (only if LDAP_READ_ONLY=false)
```

5. Set the following parameters to identify where in the LDAP/AD structure user records are stored:

```
LDAP_BASE_DN
LDAP_USER_RDN_ATTR
```

6. Set the following parameters to map SWAMP user attributes to the corresponding attributes in your LDAP/AD server:

```
LDAP_SWAMP_UID_ATTR
LDAP_FIRSTNAME_ATTR
LDAP_LASTNAME_ATTR
LDAP_FULLNAME_ATTR
LDAP_PASSWORD_ATTR
LDAP_USERNAME_ATTR
LDAP_EMAIL_ATTR
LDAP_ORG_ATTR
```

7. Set the following parameter with a comma-separated list of the `objectClass` attributes required for new user records in your LDAP/AD server. This is applicable only if `LDAP_READ_ONLY=false`.

```
LDAP_OBJECTCLASS
```

8. Save your changes to the `.env` file.

5.4.2. Designating an Initial SWAMP Administrator

When SWAMP-in-a-Box is installed, a default SWAMP administrator user is set up. The user record for this SWAMP administrator, the "admin-s" user, is stored in the SWAMP's local database. You can sign in as this user and invite other SWAMP users to become SWAMP administrators, as needed.

However, SWAMP-in-a-Box is designed to access only one source of user records. Therefore, when you configure SWAMP-in-a-Box to use an LDAP/AD server for user records, you can no longer sign in to your SWAMP with users whose records are stored in the local database. This means that initially, on configuring SWAMP-in-a-Box to use a local LDAP/AD server, your SWAMP will have no administrator users.

You can use the following procedure to promote a user to a SWAMP administrator.

Before You Begin

- You will need access to the SWAMP-in-a-Box host.
- You should have configured the SWAMP-in-a-Box to use an LDAP/AD server.
- You should have signed up or signed in to your SWAMP with the user to be promoted.
- You will need the `SWAMP_UID` value for the user to be promoted. This is the value which corresponds to the `LDAP_SWAMP_UID_ATTR` attribute for the user.
- You will need the password for the `web` database user for the SWAMP's SQL database. This can be found in `/var/www/swamp-web-server/.env` on the SWAMP-in-a-Box host. Note that `root` access is required to view this file.

Procedure

1. Enter the following on the command line for your SWAMP-in-a-Box host:

```
export PROJECT_DB_HOST=localhost
export PROJECT_DB_PORT=3306
export PROJECT_DB_DATABASE=project
export PROJECT_DB_USERNAME=web
export SWAMP_UID=<unique SWAMP_UID of new admin user>
mysql -h $PROJECT_DB_HOST -P $PROJECT_DB_PORT -u $PROJECT_DB_USERNAME -p \
      -e "USE $PROJECT_DB_DATABASE; UPDATE user_account SET admin_flag=1 \
      WHERE user_uid='$SWAMP_UID';"
```

2. When prompted, enter the password for the `web` database user.
3. Sign in to the SWAMP with the user and confirm that the user now has SWAMP administrator access.

5.4.3. Examples

Below are some examples showing the output of a command line `ldapsearch` query and the corresponding `.env` configuration.

Example 3. Secure LDAP Server with Anonymous Read Access

ldapsearch command

```
ldapsearch -LLL -x -H ldaps://ldap.ncsa.illinois.edu \
           -b "dc=ncsa,dc=illinois,dc=edu" "(sn=*smith*)"
```

ldapsearch output

```
dn: uid=jsmith,ou=People,dc=ncsa,dc=illinois,dc=edu
cn: John Smith
givenName: John
sn: Smith
uid: jsmith
mail: jsmith@illinois.edu
employeeType: all_ncsa_employe
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
objectClass: inetUser
objectClass: posixAccount
uidNumber: 28064
gidNumber: 202
homeDirectory: /afs/ncsa/.u7/jsmith
loginShell: /bin/csh
memberOf: cn=jira-users,ou=Groups,dc=ncsa,dc=illinois,dc=edu
memberOf: cn=grp_bw_ncsa_staf,ou=Groups,dc=ncsa,dc=illinois,dc=edu
memberOf: cn=org_all_groups,ou=Groups,dc=ncsa,dc=illinois,dc=edu
memberOf: cn=org_do,ou=Groups,dc=ncsa,dc=illinois,dc=edu
memberOf: cn=all_ncsa_employe,ou=Groups,dc=ncsa,dc=illinois,dc=edu
memberOf: cn=grp_jira_users,ou=Groups,dc=ncsa,dc=illinois,dc=edu
memberOf: cn=all_users,ou=Groups,dc=ncsa,dc=illinois,dc=edu
memberOf: cn=grp_bldg_ncsa,ou=Groups,dc=ncsa,dc=illinois,dc=edu
memberOf: cn=grp_bldg_both,ou=Groups,dc=ncsa,dc=illinois,dc=edu
memberOf: cn=org_cisr,ou=Groups,dc=ncsa,dc=illinois,dc=edu
memberOf: cn=org_ici,ou=Groups,dc=ncsa,dc=illinois,dc=edu
memberOf: cn=org_csd,ou=Groups,dc=ncsa,dc=illinois,dc=edu
memberOf: cn=prj_cerb_users,ou=Groups,dc=ncsa,dc=illinois,dc=edu
memberOf: cn=iam_sec_testing,ou=Groups,dc=ncsa,dc=illinois,dc=edu
memberOf: cn=lsst_users,ou=Groups,dc=ncsa,dc=illinois,dc=edu
memberOf: cn=lsst_security,ou=Groups,dc=ncsa,dc=illinois,dc=edu
memberOf: cn=ncsa-ca,ou=Groups,dc=ncsa,dc=illinois,dc=edu
```

Corresponding `.env` entry

```
LDAP_ENABLED=true
LDAP_PASSWORD_VALIDATION=true
LDAP_READ_ONLY=true
LDAP_HOST=ldaps://ldap.ncsa.illinois.edu
LDAP_PORT=636
LDAP_BASE_DN=ou=People,dc=ncsa,dc=illinois,dc=edu
LDAP_USER_RDN_ATTR=uid
LDAP_SWAMP_UID_ATTR=uid
LDAP_FIRSTNAME_ATTR=givenName
LDAP_LASTNAME_ATTR=sn
LDAP_FULLNAME_ATTR=cn
LDAP_PASSWORD_ATTR=userPassword
LDAP_USERNAME_ATTR=uid
LDAP_EMAIL_ATTR=mail
LDAP_ORG_ATTR=ignore
LDAP_OBJECTCLASS="<not applicable, ldap is read-only>"
LDAP_WEB_USER=<user here>
LDAP_WEB_USER_PASSWORD=<password here>
LDAP_PASSWORD_SET_USER="<not applicable, ldap is read-only>"
LDAP_PASSWORD_SET_USER_PASSWORD="<not applicable, ldap is read-only>"
```

Notes

In the query response, you should see:

```
dn: uid=jsmith,ou=People,dc=ncsa,dc=illinois,dc=edu
```

In this case, the `LDAP_USER_RDN_ATTR` is the key for the `uid=jsmith` portion of the `dn`, and the `LDAP_BASE_DN` is the rest of the `dn` string.

Since the `uid` field is globally unique in the LDAP directory, we set that for `LDAP_SWAMP_UID_ATTR`.

We also want the user to enter "jsmith" for username/password, so we use the default value for `LDAP_USERNAME_ATTR=uid`.

Finally, we use the default value of `LDAP_PORT=636` because we are connecting with `ldaps://`.

Example 4. Insecure Active Directory Server with Credentialed User

ldapsearch command

```
ldapsearch -LLL -x -H ldap://128.104.100.232 \
           -b "dc=swamp,dc=ad" \
           -D "ldapquery@swamp.ad" \
           -W "(sAMAccountName=*jsmith*)"
Enter LDAP Password: <password entered>
```

ldapsearch output

```
dn: CN=John Smith,CN=Users,DC=swamp,DC=ad
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: user
cn: John Smith
sn: Smith
telephoneNumber: +1 555 5551234
givenName: John
distinguishedName: CN=John Smith,CN=Users,DC=swamp,DC=ad
instanceType: 4
whenCreated: 20161102135807.0Z
whenChanged: 20161103141526.0Z
displayName: John Smith
uSNCreated: 65515
memberOf: CN=Domain Admins,CN=Users,DC=swamp,DC=ad
uSNChanged: 66272
streetAddress:: MTIwNSBXLiBDbGFyayBTdC4NClVyYmFuYSwgSUwgNjE4MjE=
name: John Smith
objectGUID:: 4YwXKKIRxEOMD9BK4WaXGQ==
userAccountControl: 66048
badPwdCount: 0
codePage: 0
countryCode: 0
badPasswordTime: 131231177822233920
lastLogoff: 0
lastLogon: 131231177936769682
pwdLastSet: 131225686874147433
primaryGroupID: 513
objectSid:: AQUAAAAAAAUVAAAA7H5IDl2Zlbb2qCf1UgQAAA==
adminCount: 1
accountExpires: 9223372036854775807
logonCount: 1
sAMAccountName: jsmith
sAMAccountType: 805306368
userPrincipalName: jsmith@swamp.ad
objectCategory: CN=Person,CN=Schema,CN=Configuration,DC=swamp,DC=ad
dSCorePropagationData: 20161102144813.0Z
dSCorePropagationData: 16010101000000.0Z
lastLogonTimestamp: 131226561268705498
mail: jsmith@illinois.edu
```

Corresponding `.env` entry

```
LDAP_ENABLED=true
LDAP_PASSWORD_VALIDATION=true
LDAP_READ_ONLY=true
LDAP_HOST=ldap://128.104.100.232
LDAP_PORT=389
LDAP_BASE_DN=cn=Users,dc=swamp,dc=ad
LDAP_USER_RDN_ATTR=cn
LDAP_SWAMP_UID_ATTR=userPrincipalName
LDAP_FIRSTNAME_ATTR=givenName
LDAP_LASTNAME_ATTR=sn
LDAP_FULLNAME_ATTR=cn
LDAP_PASSWORD_ATTR=userPassword
LDAP_USERNAME_ATTR=sAMAccountName
LDAP_EMAIL_ATTR=mail
LDAP_ORG_ATTR=ignore
LDAP_OBJECTCLASS="<not applicable, ldap is read-only>"
LDAP_WEB_USER=ldapquery@swamp.ad
LDAP_WEB_USER_PASSWORD=<password here>
LDAP_PASSWORD_SET_USER="<not applicable, ldap is read-only>"
LDAP_PASSWORD_SET_USER_PASSWORD="<not applicable, ldap is read-only>"
```

Notes

In the query response, you should see:

```
dn: CN=John Smith,CN=Users,DC=swamp,DC=ad
```

In this case, the `LDAP_USER_RDN_ATTR` is the key for the `cn=John Smith` portion of the `dn`, and the `LDAP_BASE_DN` is the rest of the `dn` string.

The user "ldapquery@swamp.ad" was configured in the AD server to have read access for the other users in the server. This was an out-of-band step.

We need a unique AD identifier to store in the local SWAMP database. In this case, we configure `LDAP_SWAMP_UID_ATTR=userPrincipalName`, but any other unique identifier could be used.

We want the user to enter "jsmith" for username/password, so we use `LDAP_USERNAME_ATTR=sAMAccountName`.

Since `LDAP_HOST` is using `ldap://`, we configure `LDAP_PORT=389` (insecure). Note that it is a bad idea to use an insecure LDAP protocol since user passwords would be transmitted in the clear.

5.4.4. Other Considerations

Required Attributes

When the SWAMP creates new users, it will always populate the following:

- SWAMP user UID
- username
- password
- first name
- last name
- full name
- email (unless email is disabled)

If you want to configure SWAMP-in-a-Box to be able to add and edit user records in an LDAP/AD server, you must have a unique corresponding attribute mapped for each of these values.

Similarly, if you have required attributes for user records in your LDAP/AD server, each must be mappable to one of the above values. Otherwise, the SWAMP will not be able to set them, and any attempt by the SWAMP to create new user records will fail.

If email is a required attribute for your LDAP/AD server but you want to configure SWAMP-in-a-Box with email disabled, you can map your email attribute, and the SWAMP will populate that attribute with a space (" ") when it creates new user records.

If you have more required attributes for your LDAP/AD server than the SWAMP can accommodate, you should configure SWAMP-in-a-Box to access your LDAP/AD server as read only.

Password Validation

If SWAMP-in-a-Box is configured for read-only access to the LDAP/AD server, then the LDAP/AD server should validate passwords.

- Set `LDAP_PASSWORD_VALIDATION=true`

If SWAMP-in-a-Box is configured to be able to add and edit records in the LDAP/AD server, and your LDAP/AD server is configured to encrypt user passwords itself, then the SWAMP should not encrypt passwords, and the LDAP/AD server should validate passwords.

- Set `APP_PASSWORD_ENCRYPTION_METHOD=NONE`
- Set `LDAP_PASSWORD_VALIDATION=true`

If SWAMP-in-a-Box is configured to be able to add and edit records in the LDAP/AD server, and your LDAP/AD server is not configured to encrypt user passwords, then the SWAMP should handle password encryption and validation.

- Set `APP_PASSWORD_ENCRYPTION_METHOD=BCRYPT`
- Set `LDAP_PASSWORD_VALIDATION=false`

LDAP Size

If your LDAP/AD server has several thousand users, your SWAMP administrator user many not be able to manage users. This is dependent on how the limits on the LDAP/AD server are configured. If the server limits the number of records that can be returned on a search, the SWAMP may receive only a subset of users when asking for all users. This in turn affects the "Review Accounts" page by causing it to show only a subset of the SWAMP's users.

5.5. Configuring Third-party Sign-in via OAuth2 Providers

The SWAMP can be configured to use external OAuth2 identity providers. Currently, the following identity providers are supported:

- GitHub
- Google
- CILogon

5.5.1. Creating a GitHub OAuth Application for Your SWAMP-in-a-Box

Before You Begin

- You will need a GitHub account or organization for which to register an OAuth application.

Procedure

1. Sign in to your GitHub account, or sign in and access your organization.
2. Navigate to the user's or organization's "Settings" page.
3. Under "Developer Settings", navigate to the "OAuth Applications" page:
    - User: https://github.com/settings/developers
    - Organization: https://github.com/organizations/<organization_name>/settings/applications
4. Click the "Register a new application" button.
5. Enter the following information:
    - Application name: "SWAMP-in-a-Box" or the name of your SWAMP-in-a-Box
    - Homepage URL: The URL to your SWAMP-in-a-Box's or organization's home page
    - Application description: Optional, you can leave this blank
    - Authorized callback URL: "https://<hostname>/oauth2", using your SWAMP-in-a-Box's hostname

6. Click the "Register application" button.
7. (Optional) Add an application logo on the summary screen. Click the "Update application" button when finished.
8. Copy down the "Client ID" and "Client Secret".

5.5.2. Enabling GitHub as an OAuth2 Provider

Before You Begin

- You will need `root` access to the SWAMP-in-a-Box host.
- You will need the "Client ID" and "Client Secret" for your SWAMP-in-a-Box's GitHub OAuth application.

Procedure

1. As `root` (or using `sudo`), edit the web backend configuration file:

```
vi /var/www/swamp-web-server/.env
```

2. Set the following parameters:

```
GITHUB_ENABLED=true
GITHUB_CLIENT_ID=<Your Client ID>
GITHUB_CLIENT_SECRET=<Your Client Secret>
```

3. Save your changes.

5.5.3. Creating Google OAuth Credentials for Your SWAMP-in-a-Box

Before You Begin

- You will need a Google account for which to enable the Google+ API and create OAuth credentials.

Procedure

1. Sign in to your Google account.
2. Navigate to the Google API Manager: https://console.developers.google.com/.
3. Select or create a Project for your SWAMP-in-a-Box OAuth credentials.
4. Enable the Google+ API for your project:
    - On the left, under API Manager, select "Library".
    - On the right, under Social APIs, select the link for "Google+ API".

- Click the "Enable" button.

5. Configure the OAuth consent screen:
   - On the left, under API Manager, select "Credentials".
   - On the right, under Credentials, select "OAuth consent screen".
   - Enter the following information:
     - Email address: Your email
     - Product name shown to users: "SWAMP-in-a-Box" or the name of your SWAMP-in-a-Box
     - Homepage URL: The URL to your SWAMP-in-a-Box's or organization's home page
     - Product logo URL: The URL to a logo for your SWAMP-in-a-Box. For example, mir-swamp.org uses: https://www.mir-swamp.org/images/logos/swamp-icon-small.png.
     - Privacy policy URL: The URL to your privacy policy. For example, mir-swamp.org uses: https://www.swampinabox.org/doc/SWAMP-Privacy-Policy.pdf.
     - Terms of service URL: The URL to your terms of service. For example, mir-swamp.org uses: https://www.mir-swamp.org/#policies/acceptable-use-policy.
   - Click "Save".
6. Configure OAuth Client ID Credentials:
   - On the left, under API Manager, select "Credentials".
   - On the right, under Credentials, select "OAuth client ID" from the "Create credentials" menu.
   - Under "Application type", select "Web application".
   - Enter the following information:
     - Name: "SWAMP-in-a-Box" or the name of your SWAMP-in-a-Box
     - Authorized JavaScript origins: "https://<hostname>", using your SWAMP-in-a-Box's hostname
     - Authorized redirect URIs: "https://<hostname>/oauth2", using your SWAMP-in-a-Box's hostname
   - Click "Create".
7. Copy down the "Client ID" and "Client Secret".

5.5.4. Enabling Google as an OAuth2 Provider

Before You Begin

- You will need `root` access to the SWAMP-in-a-Box host.
- You will need the "Client ID" and "Client Secret" for your SWAMP-in-a-Box's Google OAuth credentials.

Procedure

1. As `root` (or using `sudo`), edit the web backend configuration file:

```
vi /var/www/swamp-web-server/.env
```

2. Set the following parameters:

```
GOOGLE_ENABLED=true
GOOGLE_CLIENT_ID=<Your Client ID>
GOOGLE_CLIENT_SECRET=<Your Client Secret>
```

3. Save your changes.

5.5.5. Registering for CILogon OAuth2 Credentials

Procedure

1. Go to https://cilogon.org/oauth2/register.
2. Enter the following information:
   - Client Name: "SWAMP-in-a-Box" or the name of your SWAMP-in-a-Box
   - Contact email: Your email address
   - Home URL: The URL to your SWAMP-in-a-Box's or organization's home page
   - Uncheck "Use Limited Proxy Certificates"
   - Callback URLs: "https://<hostname>/oauth2", using your SWAMP-in-a-Box's hostname
3. Click the "Submit" button.
4. Copy down the client identifier and client secret.
5. Wait for email approval from CILogon Administrator.

5.5.6. Enabling CILogon as an OAuth2 Provider

Before You Begin

- You will need `root` access to the SWAMP-in-a-Box host.
- You will need the client identifier and client secret for your SWAMP-in-a-Box's CILogon OAuth2 credentials.

Procedure

1. As `root` (or using `sudo`), edit the web backend configuration file:

```
vi /var/www/swamp-web-server/.env
```

2. Set the following parameters:

```
CILOGON_ENABLED=true
CILOGON_CLIENT_ID=<Your Client ID>
CILOGON_CLIENT_SECRET=<Your Client Secret>
```

3. Save your changes.

5.6. Configuring a Welcome Message for SWAMP-in-a-Box

You can configure SWAMP-in-a-Box to display a welcome message as a pop up whenever a user accesses the SWAMP-in-a-Box home page (not signed in).

You can use this to provide a welcome message, convey information about your SWAMP-in-a-Box, or provide status information.

5.6.1. Before You Begin

- You need `root` access to the SWAMP-in-a-Box host.

5.6.2. Modify `/var/www/html/config/config.json` with your message

- Add a `notifications` array containing a `welcome` array.
- Add `title` and `message` values to the `welcome` array.

Note:

- The `config.json` file defines parameters within JSON arrays. Therefore, it is important to maintain the array format when editing, adding, or removing parameters in this file.

Sample:

```
"notifications": {
  "welcome": {
    "message": "Your message here!",
    "title": "Welcome to SWAMP-in-a-Box for <organization>"
  }
},
```

6. Maintaining SWAMP-in-a-Box

6.1. Checking for Updates

The SWAMP-in-a-Box upgrade script does not necessarily update all components of SWAMP-in-a-Box for which a newer version might be available. Components that require additional steps to upgrade are listed below, along with instructions on how to upgrade them.

Assessment Platforms

If you have previously installed additional assessment platforms, run the following command to determine whether there are updated versions available (the command does not require `root` access):

```
/opt/swamp/bin/swamp_check_platform_images
```

For any out-of-date platform images, follow the instructions on installing additional assessment platforms for downloading and installing the updated images. Once that is done, the out-of-date images may be deleted.

6.2. Updating the Host's Hostname

During the SWAMP-in-a-Box install process, the hostname specified for the host is set in the SWAMP's various configuration files and its database. When that hostname changes, those configuration locations must be updated in order for the system to continue functioning correctly. To set the new hostname, as `root` (or using `sudo`), run the following command:

```
/opt/swamp/bin/swamp_set_web_host --hostname="<new hostname>"
```

6.3. Backing Up and Restoring the SQL Database

SWAMP-in-a-Box uses MariaDB as its SQL database implementation. While it is technically possible to use MariaDB's `mysqldump` command and `mysql` shell to back up and restore the SQL database, that process will not account for changes to the database schemas and records between SWAMP-in-a-Box releases.

The SWAMP-in-a-Box upgrade script creates a backup of the SWAMP's SQL database prior to upgrading it. A backup can also be created at any other time by running the following command as `root` (or using `sudo`):

```
/opt/swamp/bin/swamp_backup_db --output-dir=<DIR>
```

Replace `<DIR>` with the directory where the backup should be saved, and provide the database's `root` user's password when asked. The command's output will indicate where the backup has been saved.

To restore the backup, run the following command as `root` (or using `sudo`):

```
/opt/swamp/bin/swamp_restore_db --sql-db-file=<FILE>
```

Replace `<FILE>` with the path to the file created by `swamp_backup_db`, and provide the database's `root` user's password when asked.

6.4. Managing Disk Space

SWAMP-in-a-Box will use increasing amounts of disk space as SWAMP users upload packages and assess them. However, disk space from temporary files and (older) log files can be reclaimed by removing those files.

`/var/log/httpd`
: This directory contains the web server's log files. By default, when the web server (i.e., `httpd` package) is installed, the `logrotate` utility is configured, via `/etc/logrotate.d/httpd`, to rotate these log files and ensure that log entries do not persist indefinitely.

`/var/www/swamp-web-server/storage/logs`
: This directory contains the SWAMP web application's backend's log files, one per day, when it is *not* configured to make log entries in the system log.

`/var/www/swamp-web-server/storage/framework/sessions`
: This directory contains data about SWAMP users' login sessions when the SWAMP web application's backend is configured to store session data on the file system (as opposed to in a cookie in the user's web browser).

`/swamp/outgoing`
: This directory contains temporary copies of SWAMP artifacts, such as the SCARF XML for the results from an assessment, so that SWAMP users can download them. Older files may safely be deleted from this directory. The following `find` command, when run as `root` (or using `sudo`), will remove any files older than 4 hours (240 minutes):

  ```
  find /swamp/outgoing -mindepth 1 -mmin +240 -delete
  ```

`/swamp/working/results`
: This directory contains temporary copies of the artifacts produced by assessments, before they have been fully processed and saved by the SWAMP's backend. Each assessment gets its own

subdirectory, named after its execution record UUID. Older subdirectories may safely be deleted from this directory, at the expense of possibly losing some debugging information for assessments that failed. The following `find` command, when run as `root` (or using `sudo`), will remove any subdirectories that have not changed in 14 days:

```
find /swamp/working/results -mindepth 1 -maxdepth 1 -ctime +14 -delete
```

6.5. Managing Firewalls

Refer back to the section on configuring firewalls for information on the network traffic that the SWAMP-in-a-Box host is expected to respond to and generate. The SWAMP-in-a-Box install/upgrade process and supporting utility scripts do not modify the host's firewall configuration.

6.6. Managing HTCondor

SWAMP-in-a-Box uses an HTCondor pool to run assessments and the optional Code Dx viewer. Each SWAMP assessment and viewer instance is submitted to the HTCondor pool as a single job. The commands listed below can be used to examine and modify HTCondor's queue of jobs and the resources available to it for running those jobs.

Much of the functionality provided by the commands listed below is also available through the SWAMP web application. As a SWAMP administrator, go to the System Overview page, and from there, to the Review Status page.

SWAMP-in-a-Box installs and uses a "personal" HTCondor pool. As a result, HTCondor's commands are not in the shell's default search path, and each command must be run in an environment where `CONDOR_CONFIG` is set to the path to the SWAMP's HTCondor configuration file.

Shells started after SWAMP-in-a-Box has been installed should automatically load `/etc/profile.d/swamp.sh`, which modifies the shell's environment so that the commands listed below may be run exactly as shown.

When using `sudo` to run the commands, first run `sudo -i`, which starts a new shell that runs as `root`. Then run the commands as shown. Alternatively, explicitly specify the value for `CONDOR_CONFIG` and the full path to the command as part of the `sudo` command:

```
sudo CONDOR_CONFIG=/opt/swamp/htcondor/etc/condor_config
/opt/swamp/htcondor/bin/<command> ...
```

`condor_q`
: Lists the jobs currently in the queue. HTCondor's ID for each job is shown in the "ID" column. The status of each job is shown in the "ST" column: "I" for idle; "R" for running; and "H" for on-hold, indicating that the job encountered an error.

`condor_q -better-analyze <job ID>`
: Displays detailed information about why an idle job is not currently running. On a normally functioning system, it is normal for a job to be idle because there are not enough CPU or memory resources available (they should become available as other running assessments finish).

`condor_q -hold <job ID>`
: Displays detailed information about why a job is on-hold. On a normally functioning system, no job should be on-hold.

`condor_status`
: Lists all of the available resources that HTCondor can use to run jobs. On a normally functioning system, there should be at least one "machine" in this list.

`condor_status -vm`
: Lists all of the available resources that HTCondor can use to run jobs that require a virtual machine. All SWAMP jobs require a virtual machine. On a normally functioning system, there should be at least one "machine" in this list.

`condor_release <job ID>`
: Releases a job that is on-hold, which allows HTCondor to try scheduling and running the job again. This command must be run as `root` (or using `sudo`).

`condor_rm <job ID>`
: Removes a job from the queue. This command must be run as `root` (or using `sudo`). It can be used to remove a job from the queue, though this should not be necessary on a normally functioning system.

6.7. Managing SWAMP Daemons

SWAMP-in-a-Box includes a collection of daemons that run on the host, all managed by the `swamp` system service. These daemons must be running in order to submit and perform assessments. The `swamp` service and the daemons managed by it can be stopped and started using the standard commands for interacting with system services (the commands must be run as `root` or using `sudo`). For example:

```
service swamp start
service swamp stop
service swamp restart
```

6.8. Other Considerations

SWAMP-in-a-Box uses Apache HTTP Server, HTCondor, and MariaDB. For instructions on how to interact with or administer Apache HTTP Server, HTCondor, and MariaDB, refer to the documentation associated with each product. Be aware that the install and upgrade process for SWAMP-in-a-Box makes changes to their default configurations; see the SWAMP-in-a-Box Reference Manual for further details.

7. Troubleshooting SWAMP-in-a-Box

7.1. Checking the Host's Health

SWAMP-in-a-Box includes a script, `swamp_check_install`, for checking that its components are functioning as expected. Run this script as the first step in troubleshooting issues with the SWAMP web application or assessments, because it checks for and warns about many common problems.

To run the script, run the following command as `root` (or using `sudo`), replacing `<hostname>` with the hostname for your SWAMP-in-a-Box's web application (for example, `swamp.example.com`):

```
/opt/swamp/bin/swamp_check_install --hostname "<hostname>"
```

The script will display information about the checks it is performing and summarize its findings. The meanings of any warnings and errors, and potential remedies for them, are described below.

If you are unable to resolve your issue, contact SWAMP staff. Include the full output from the script and SWAMP-in-a-Box's log files. Refer to the section below on collecting log files for instructions on how to bundle SWAMP-in-a-Box's logs into a single archive.

7.1.1. Messages About the SQL Database

The mysql system service is not running

Indicates that the database server is not running, which will prevent the web application and assessments from functioning correctly. Start the service by running the following command as `root` (or using `sudo`):

```
service mysql restart
```

Failed to connect to the SQL database

Indicates that the libraries used by the SWAMP's backend are unable to establish a connection to the database. If you have recently changed the password for the database's `java_agent` user, update the

`dbPerlPass` setting in `/opt/swamp/etc/swamp.conf` with the new password (you will need `root` access to edit this file).

7.1.2. Messages About HTCondor

Any HTCondor-related warnings and errors indicate an issue that is likely preventing the SWAMP from performing assessments and running the optional Code Dx viewer.

The condor system service is not running

Indicates that HTCondor is not running. This is likely the cause of any other HTCondor-related warnings and errors found by `swamp_check_install`. Start the service by running the following command as `root` (or using `sudo`):

```
service condor restart
```

Note that it normally takes a few minutes for HTCondor's daemons to start up.

'condor_q ...' exited with ...

'condor_status ...' exited with ...

Indicates that HTCondor is misconfigured. Contact SWAMP staff.

The HTCondor pool has no resources for running jobs

Indicates that HTCondor is misconfigured. Contact SWAMP staff.

The HTCondor pool has no resources for running VM jobs

Indicates that HTCondor is unable to run jobs which require a virtual machine, which will prevent the SWAMP from performing assessments and running the optional Code Dx viewer.

If `swamp_check_install` also indicated issues with `libvirtd` (see the list of messages about libvirtd below), resolve those first.

Otherwise, run the following script to determine whether the SWAMP-in-a-Box host supports KVM virtualization, as described in the hardware requirements for SWAMP-in-a-Box:

```
/opt/swamp/bin/swamp_check_virtualization_support
```

If the script does *not* find the necessary support for KVM virtualization, it will display an error message and a suggestion on how to resolve the issue.

If the script *does* find the necessary support for KVM virtualization, then what likely happened is that some job failed to start its virtual machine successfully. The immediate cause of the failure might be listed in `/var/log/condor/VMGahpLog` (look around the times an assessment was submitted or failed).

In any event, restart the `condor` service by running the following command as `root` (or using `sudo`):

```
service condor restart
```

Note that it normally takes a few minutes for HTCondor's daemons to start up.

If this error from `swamp_check_install` persists, then there is likely a systemic issue that requires further investigation. Contact SWAMP staff.

The HTCondor queue has ... held jobs

Indicates that one or more HTCondor jobs encountered an unexpected error. Use the `condor_q` command, as described in the section on managing HTCondor, to determine why HTCondor put the job on-hold. Then contact SWAMP staff.

7.1.3. Messages About libvirtd

The libvirtd system service is not running

Indicates that the `libvirtd` service is not running, which will prevent the SWAMP from performing assessments and running the optional Code Dx viewer. Start the service by running the following command as `root` (or using `sudo`):

```
service libvirtd restart
```

If `swamp_check_install` also indicated issues with HTCondor (see the list of messages about HTCondor above), also restart the `condor` system service by running the following command as `root` (or using `sudo`):

```
service condor restart
```

Note that it normally takes a few minutes for HTCondor's daemons to start up.

Failed to find SWAMP's libvirt iptables rules

Indicates that the firewall rules that allow assessment virtual machines to access the host's network are absent, which will prevent assessments from doing anything that requires network access, such as contacting license servers. Restart the `libvirtd` service by running the following command as `root` (or using `sudo`):

```
service libvirtd restart
```

7.1.4. Messages About SWAMP Daemons

The swamp system service is not running

Indicates that the `swamp` system service is not running, which will prevent the SWAMP from performing assessments and running the optional Code Dx viewer. Start the service by running the following command as `root` (or using `sudo`):

```
service swamp restart
```

RPC to AgentMonitor failed

RPC to LaunchPad failed

RPC to AgentMonitor returned something unexpected

RPC to LaunchPad returned something unexpected

Indicates that one or both of the SWAMP daemons necessary for running assessments is not running correctly. Contact SWAMP staff.

7.1.5. Messages About the Web Application

The httpd system service is not running

Indicates that the web server is not running, which will prevent everyone from accessing the SWAMP web application. Start the web server by running the following command as `root` (or using `sudo`):

```
service httpd restart
```

Failed to connect to 'http(s)://<hostname>'

Indicates that the web server for the SWAMP web application is not reachable. If the web server is not running on the host (see the message directly above), resolve that issue first. Otherwise, refer to the section on configuring the host's firewall.

'<hostname>' does not appear to support https

Indicates that the web server does not support encrypted connections, which means that any information, including usernames and passwords, sent between the SWAMP web application and the web server will **not** be encrypted.

By default, SWAMP-in-a-Box configures the web server to support only encrypted connections (HTTPS). If you intentionally disabled this support, then you can ignore this message. Otherwise, contact SWAMP staff.

'<hostname>' might not have a valid SSL certificate

Indicates that the web server does not have a properly signed SSL certificate that matches `<hostname>`. Refer to the section on configuring an SSL certificate for SWAMP-in-a-Box.

'http(s)://<hostname>/config/config.json' is not valid JSON

Indicates that `/var/www/html/config/config.json` on the SWAMP-in-a-Box host does not contain valid JSON, which will cause the SWAMP web application to appear stuck on a "loading" screen. Check this file for typos.

'http(s)://<hostname>/config/config.json' does not define 'servers.web'

Indicates that `/var/www/html/config/config.json` on the SWAMP-in-a-Box host does not contain the configuration key that specifies the location of the SWAMP API (i.e., SWAMP backend). Follow the directions in the section on updating the host's hostname, which will, as a side effect, set the required configuration key.

Failed to fetch '<api-location>/environment'

Failed to fetch '<api-location>/platforms/public'

'<api-location>/platforms/public' is not valid JSON

Indicates that SWAMP API is not functioning correctly. First, resolve the other errors reported by `swamp_check_install`, if any. If these errors persist, then contact SWAMP staff.

7.1.6. Other Messages

Failed to determine PHP version

Indicates that the script was unable to determine the version of the PHP command line interpreter that is installed on the host.

If the SWAMP web application is *not* functioning as expected, it is likely that the wrong version of PHP is installed. Refer to the appendix on installing PHP for instructions on installing the required version of PHP.

If the SWAMP web application *is* functioning as expected, then this message can be ignored. However, feel free to contact SWAMP staff and provide the full output from `swamp_check_install` so that they can work on removing this false positive from future versions of the script.

Found PHP ..., not 7.*

Indicates that the script found an unsupported version of PHP installed on the host. Refer to the appendix on installing PHP for instructions on installing the required version of PHP.

7.2. Collecting Log Files

When investigating an issue with SWAMP-in-a-Box, SWAMP staff often find it helpful to review:

- The SWAMP's log files (located in `/opt/swamp/log`)
- HTCondor's log files (located in `/var/log/condor`)
- HTCondor's configuration files (located in `/etc/condor`)

To bundle all of these files into a single archive, run the following command:

```
tar -cvz -f swampinabox-logs.tar.gz --exclude="*.old" \
    /opt/swamp/log/*.log \
    /var/log/condor/*Log \
    /var/log/condor/*Log.slot* \
    /etc/condor
```

This will create a file `swampinabox-logs.tar.gz` in the current working directory. Errors from `tar` about `No such file or directory` may safely be ignored (some of the files that the command looks for might not exist on all systems).

7.3. Debugging Failed Assessments

If an assessment reaches a status of "Finished with Errors," the SWAMP is functioning normally, but the assessment failed to yield any results. Click the "Error" button to view a detailed report about the failure. The following sections in the report summarize the major steps in performing the assessment and the output they produced:

- Contents of status.out
- Contents of stderr
- Contents of stdout

Details about the messages in `status.out` can be found in the document Status.out and Debugging SWAMP Failures, a link to which can be found near the beginning of the error report.

If an assessment reaches some other error state, the assessment's log file might indicate why:

1. On the Assessment Results page of the SWAMP web application, click on the assessment's status. Locate the execution record UUID.
2. The assessment's log file will be located at `/opt/swamp/log/<execution-record-UUID>.log` on the SWAMP-in-a-Box host.

7.4. Debugging Stuck Assessments

If an assessment appears stuck at a status of "Waiting in HTCondor Queue" or is unable to be submitted to HTCondor, refer back to the section on checking the host's health.

If an assessment appears stuck at a status of "Shutting down the VM," it is likely that the assessment's virtual machine has encountered an issue. At this point, the assessment has completed its work; all that remains is to extract its output from the virtual machine, which requires that the virtual machine be shut down.

To force the virtual machine to shut down:

1. On the Assessment Results page of the SWAMP web application, click on the assessment's status.

Locate the execution record UUID.

2. As a SWAMP administrator, go to the System Overview page, and from there, to the Review Status page.
3. Under the Condor Queue tab, locate the row for the assessment's execution run UUID. Note the virtual machine's name in the VM column.
4. On the SWAMP-in-a-Box host, as `root` (or using `sudo`), run `virsh`. This will start a shell that can be used to interact with the virtual machines currently running on the host. Available commands include:

   `list`
   : Displays a list of all the virtual machines currently being managed by `libvirt`.

   `console <id>`
   `console <virtual-machine-name>`
   : Connects to a virtual machine's console. Type control-] to detach from the console.

   `destroy <id>`
   `destroy <virtual-machine-name>`
   : Immediately shuts down and stops a virtual machine.

   `exit`
   : Exits the `virsh` shell.

5. Use the `destroy` command, as described above, to shut down the virtual machine. The SWAMP should finish processing the assessment and make its results available in the SWAMP web application.

7.5. Using Java CLI and Related Plugins with SWAMP-in-a-Box

SWAMP-in-a-Box supports the following minimum versions of Java CLI and related plug-ins. If you are using an earlier version, please upgrade.

- Java CLI version 1.5.2
- SWAMP Eclipse Plugin version 1.1.2
- SWAMP Jenkins Plugin version 1.2.1

8. Support and Contact Information

We welcome your feedback, contributions, and questions at:

- Email: support@continuousassurance.org

To receive updates on SWAMP-in-a-Box and be part of the user community, please join our mailing list:

- Email: swampinabox@lists.discovery.wisc.edu
- Sign up: https://lists.cosalab.org/mailman/listinfo/swampinabox

To report a security incident or concern with SWAMP-in-a-Box, please contact us at:

- Email: security@continuousassurance.org

You may encrypt your email for privacy using GPG (key id#739202FA, fingerprint 2793 A0A7 4340 7587 FC2A 160F FE83 C695 7392 02FA).

Appendix A: Installing Dependencies

The software packages that SWAMP-in-a-Box depends on include:

- MariaDB 5.5,
- PHP 7.0, and
- other assorted utilities.

All of these dependencies must be installed in order for SWAMP-in-a-Box to function correctly. If you run into issues installing these dependencies, refer to the troubleshooting section at the end of this appendix for possible solutions.

In the sections below, `<installer-dir>` refers to the directory containing the SWAMP-in-a-Box installer. Refer back to the section on installing and upgrading SWAMP-in-a-Box for instructions on obtaining and extracting the installer.

A.1. MariaDB 5.5

For CentOS 6, the set-up scripts for SWAMP-in-a-Box configure and download MariaDB from the repository hosted by the MariaDB Foundation, using the configuration file produced by the "repository configuration" tool at https://downloads.mariadb.org/mariadb/repositories/. The specific packages installed are `MariaDB-client`, `MariaDB-server`, `MariaDB-shared`, and their dependencies.

For CentOS 7, the set-up scripts for SWAMP-in-a-Box download MariaDB from CentOS's default repositories. The specific packages installed are `mariadb`, `mariadb-server`, `mariadb-libs`, and their dependencies.

The following script will install MariaDB using the process described above:

```
<installer-dir>/repos/install-mariadb.bash
```

A.2. PHP 7.0

The set-up scripts for SWAMP-in-a-Box configure and download PHP from Remi's RPM Repository, using the instructions produced by the "configuration wizard" at http://rpms.famillecollet.com/. The specific packages installed are:

- `php`,
- `php-ldap`,
- `php-mbstring`,
- `php-mcrypt`,
- `php-mysqlnd`,
- `php-pecl-zip`,
- `php-xml`,

and their dependencies.

The following script will install PHP using the process described above:

```
<installer-dir>/repos/install-php.bash
```

A.3. Other Assorted Utilities

In addition to MariaDB and PHP, the set-up scripts for SWAMP-in-a-Box download assorted software packages from CentOS's default repositories. The specific packages installed are:

- `ant`,
- `bind-utils`,
- `curl`,
- `git`,
- `httpd`,
- `libguestfs`,
- `libguestfs-tools`,
- `libguestfs-tools-c`,
- `libvirt`,
- `mod_ssl`,
- `ncompress`,
- `patch`,

- `perl`,
- `perl-parent`,
- `python34`,
- `rubygems`,
- `unzip`,
- `zip`,

and their dependencies.

The following script will install these packages:

```
<installer-dir>/repos/install-other-deps.bash
```

A.4. Troubleshooting Issues with Installing Dependencies

The SWAMP-in-a-Box setup and install process requires downloading and installing packages from multiple package repositories. On systems configured to check for GPG signatures on the repositories' metadata, this process might fail because not all of the repositories provide GPG signatures for their metadata. This is indicated by HTTP 404 errors when attempting to download `repomd.xml.asc` from the repository:

```
http://example.com/.../repomd.xml.asc: [Errno 14] HTTP Error 404 - Not Found
```

These GPG signature checks can be disabled by changing `repo_gpgcheck=1` to `repo_gpgcheck=0` in the configuration files used by `yum` (you will need `root` access to modify these files). To locate the configuration files that contain `repo_gpgcheck=1`, run the following command:

```
grep -lr "repo_gpgcheck=1" /etc/yum.conf /etc/yum.repos.d/
```

Appendix B: Obtaining Additional Tools and Viewers

B.1. Code Dx

Through SWAMP's partnership with Code Dx, Inc., a SWAMP-specific version of Code Dx software has been created to be solely used with SWAMP software. Code Dx software shall not be redistributed with

SWAMP software without written consent of Code Dx, Inc.

To obtain the SWAMP version of Code Dx, contact Code Dx, Inc. at:

- sales@codedx.com,
- +1-631-759-3993, or
- https://codedx.com/support/?v=7516fd43adaa.

After contacting Code Dx, Inc., you will be asked to agree to an End User's License Agreement (EULA) with Code Dx, Inc. Once you have agreed to the EULA, you will receive a download kit from Code Dx, Inc.

Code Dx is third-party software created and maintained by Code Dx, Inc. Copyright 2010-2018 Code Dx, Inc. All rights reserved.

B.2. CodeSonar

SWAMP-in-a-Box can be used with CodeSonar, a deep-path static analysis tool provided by GrammaTech, Inc. CodeSonar finds cases of undefined behavior (such as buffer overruns, null pointer dereferences, ...), API Misuse (use after free, socket API, ...), as well as suspicious behavior (dead code, unused variables, concurrency violations, taint, ...), and works on source code and binaries.

Contact information for obtaining CodeSonar and licensing information for CodeSonar can be found at:

- sales@grammatech.com,
- +1-888-695-2668, or
- https://www.grammatech.com/products/codesonar.

CodeSonar is third-party software created and maintained by GrammaTech, Inc. Copyright 2018 GrammaTech, Inc. CodeSonar is a registered trademark of GrammaTech, Inc. All rights reserved.

B.3. Parasoft C/C++test and Parasoft Jtest

SWAMP-in-a-Box can be used with C/C++test and Jtest, static analysis and unit testing tools for C/C++ and Java development, provided by Parasoft. Part of Parasoft's suite of automated software testing tools, these solutions facilitate software development best practices, rigorous bug detection, and security vulnerability remediation. Parasoft C/C++test and Jtest's static analysis and unit testing technologies bring efficiency to quality and compliance initiatives. The latest releases improve developer workflows, with a focus on enhanced environment and embedded support, and provide enriched dashboards and tracking, to aid users in addressing vulnerabilities in standards like OWASP, CWE, or achieving MISRA compliance.

Contact information for obtaining C/C++test or Jtest and licensing information for C/C++test or Jtest can

be found at:

- swamp@parasoft.com, and
- +1-719-424-7907.

Appendix C: License and Notices

The Software Assurance Marketplace (SWAMP) is released under the open source Apache License, Version 2.0, reproduced below.

Additional notices for SWAMP can be found following the license.

C.1. Apache License, Version 2.0

```
                                 Apache License
                           Version 2.0, January 2004
                        http://www.apache.org/licenses/

   TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

   1. Definitions.

      "License" shall mean the terms and conditions for use, reproduction,
      and distribution as defined by Sections 1 through 9 of this document.

      "Licensor" shall mean the copyright owner or entity authorized by
      the copyright owner that is granting the License.

      "Legal Entity" shall mean the union of the acting entity and all
      other entities that control, are controlled by, or are under common
      control with that entity. For the purposes of this definition,
      "control" means (i) the power, direct or indirect, to cause the
      direction or management of such entity, whether by contract or
      otherwise, or (ii) ownership of fifty percent (50%) or more of the
      outstanding shares, or (iii) beneficial ownership of such entity.

      "You" (or "Your") shall mean an individual or Legal Entity
      exercising permissions granted by this License.

      "Source" form shall mean the preferred form for making modifications,
      including but not limited to software source code, documentation
      source, and configuration files.

      "Object" form shall mean any form resulting from mechanical
      transformation or translation of a Source form, including but
```

```
      not limited to compiled object code, generated documentation,
      and conversions to other media types.

      "Work" shall mean the work of authorship, whether in Source or
      Object form, made available under the License, as indicated by a
      copyright notice that is included in or attached to the work
      (an example is provided in the Appendix below).

      "Derivative Works" shall mean any work, whether in Source or Object
      form, that is based on (or derived from) the Work and for which the
      editorial revisions, annotations, elaborations, or other modifications
      represent, as a whole, an original work of authorship. For the purposes
      of this License, Derivative Works shall not include works that remain
      separable from, or merely link (or bind by name) to the interfaces of,
      the Work and Derivative Works thereof.

      "Contribution" shall mean any work of authorship, including
      the original version of the Work and any modifications or additions
      to that Work or Derivative Works thereof, that is intentionally
      submitted to Licensor for inclusion in the Work by the copyright owner
      or by an individual or Legal Entity authorized to submit on behalf of
      the copyright owner. For the purposes of this definition, "submitted"
      means any form of electronic, verbal, or written communication sent
      to the Licensor or its representatives, including but not limited to
      communication on electronic mailing lists, source code control systems,
      and issue tracking systems that are managed by, or on behalf of, the
      Licensor for the purpose of discussing and improving the Work, but
      excluding communication that is conspicuously marked or otherwise
      designated in writing by the copyright owner as "Not a Contribution."

      "Contributor" shall mean Licensor and any individual or Legal Entity
      on behalf of whom a Contribution has been received by Licensor and
      subsequently incorporated within the Work.

   2. Grant of Copyright License. Subject to the terms and conditions of
      this License, each Contributor hereby grants to You a perpetual,
      worldwide, non-exclusive, no-charge, royalty-free, irrevocable
      copyright license to reproduce, prepare Derivative Works of,
      publicly display, publicly perform, sublicense, and distribute the
      Work and such Derivative Works in Source or Object form.

   3. Grant of Patent License. Subject to the terms and conditions of
      this License, each Contributor hereby grants to You a perpetual,
      worldwide, non-exclusive, no-charge, royalty-free, irrevocable
      (except as stated in this section) patent license to make, have made,
      use, offer to sell, sell, import, and otherwise transfer the Work,
      where such license applies only to those patent claims licensable
      by such Contributor that are necessarily infringed by their
```

Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
   Work or Derivative Works thereof in any medium, with or without
   modifications, and in Source or Object form, provided that You
   meet the following conditions:

   (a) You must give any other recipients of the Work or
       Derivative Works a copy of this License; and

   (b) You must cause any modified files to carry prominent notices
       stating that You changed the files; and

   (c) You must retain, in the Source form of any Derivative Works
       that You distribute, all copyright, patent, trademark, and
       attribution notices from the Source form of the Work,
       excluding those notices that do not pertain to any part of
       the Derivative Works; and

   (d) If the Work includes a "NOTICE" text file as part of its
       distribution, then any Derivative Works that You distribute must
       include a readable copy of the attribution notices contained
       within such NOTICE file, excluding those notices that do not
       pertain to any part of the Derivative Works, in at least one
       of the following places: within a NOTICE text file distributed
       as part of the Derivative Works; within the Source form or
       documentation, if provided along with the Derivative Works; or,
       within a display generated by the Derivative Works, if and
       wherever such third-party notices normally appear. The contents
       of the NOTICE file are for informational purposes only and
       do not modify the License. You may add Your own attribution
       notices within Derivative Works that You distribute, alongside
       or as an addendum to the NOTICE text from the Work, provided
       that such additional attribution notices cannot be construed
       as modifying the License.

   You may add Your own copyright statement to Your modifications and
   may provide additional or different license terms and conditions
   for use, reproduction, or distribution of Your modifications, or
   for any such Derivative Works as a whole, provided Your use,
   reproduction, and distribution of the Work otherwise complies with

the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
   any Contribution intentionally submitted for inclusion in the Work
   by You to the Licensor shall be under the terms and conditions of
   this License, without any additional terms or conditions.
   Notwithstanding the above, nothing herein shall supersede or modify
   the terms of any separate license agreement you may have executed
   with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
   names, trademarks, service marks, or product names of the Licensor,
   except as required for reasonable and customary use in describing the
   origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
   agreed to in writing, Licensor provides the Work (and each
   Contributor provides its Contributions) on an "AS IS" BASIS,
   WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
   implied, including, without limitation, any warranties or conditions
   of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
   PARTICULAR PURPOSE. You are solely responsible for determining the
   appropriateness of using or redistributing the Work and assume any
   risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
   whether in tort (including negligence), contract, or otherwise,
   unless required by applicable law (such as deliberate and grossly
   negligent acts) or agreed to in writing, shall any Contributor be
   liable to You for damages, including any direct, indirect, special,
   incidental, or consequential damages of any character arising as a
   result of this License or out of the use or inability to use the
   Work (including but not limited to damages for loss of goodwill,
   work stoppage, computer failure or malfunction, or any and all
   other commercial damages or losses), even if such Contributor
   has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
   the Work or Derivative Works thereof, You may choose to offer,
   and charge a fee for, acceptance of support, warranty, indemnity,
   or other liability obligations and/or rights consistent with this
   License. However, in accepting such obligations, You may act only
   on Your own behalf and on Your sole responsibility, not on behalf
   of any other Contributor, and only if You agree to indemnify,
   defend, and hold each Contributor harmless for any liability
   incurred by, or claims asserted against, such Contributor by reason
   of your accepting any such warranty or additional liability.

```
   END OF TERMS AND CONDITIONS

   Copyright 2012-2019 Software Assurance Marketplace
```

C.2. Notices

- This product includes HTCondor (https://research.cs.wisc.edu/htcondor/index.html) software developed by the Center for High Throughput Computing at the University of Wisconsin-Madison. All rights reserved. Details about the HTCondor license can be found at https://research.cs.wisc.edu/htcondor/license.html.
- This product contains Laravel (https://laravel.com/), an open source PHP framework licensed under the MIT License (https://opensource.org/licenses/MIT). Copyright Taylor Otwell.
- This product contains Code Dx, a commercial product created by Code Dx, Inc. Copyright 2010-2019 Code Dx, Inc. All rights reserved. SWAMP has a partnership with Code Dx, Inc. and offers a SWAMP-specific version of Code Dx software to be used solely with SWAMP software. Code Dx software shall not be redistributed with SWAMP software without written consent of SWAMP or Code Dx, Inc. Contact for licensing information and support for Code Dx can be found at sales@codedx.com, +1-631-759-3993, or https://codedx.com/support/?v=7516fd43adaa.
- This product includes a compiled, unmodified version of lib_mysqludf_sys, an open source library developed by Roland Bouman and Bernardo Damele A. G. and licensed under the GNU Lesser General Public License v2.1 (https://www.gnu.org/licenses/old-licenses/lgpl-2.1.en.html). Copyright 2007 Roland Bouman, 2008-2009 Roland Bouman and Bernardo Damele A. G. All rights reserved.