SWAMP-in-a-Box Administrator Manual

v1.31

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 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 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.

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, MySQL/MariaDB, and HTCondor).

2.1. Hardware Requirements

Minimum:

- Memory: 16G
- Disk: 256G
- Cores: 4

Recommended:

- Memory: 64G
- Disk: 1T
- Cores: 8

SWAMP-in-a-Box uses virtual machines managed by an HTCondor pool to perform assessments of packages and to run the optional Code Dx results viewer from Code Dx, Inc. Each virtual machine is provisioned with 6G of RAM and 1 core. The minimum requirements above are intended to allow the host machine 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.

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.

If you are installing SWAMP-in-a-Box in a virtual machine, the hypervisor must support and be configured for nested virtualization, because SWAMP-in-a-Box itself uses virtual machines to perform assessments of packages and to run the optional Code Dx results viewer from Code Dx, Inc.

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. Disabling 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, add or update `/etc/selinux/config` on the host to include the following line (you will need `root` access to edit this file):

```
SELINUX=disabled
```

Then reboot the host.

2.5. Creating 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`:

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

2.6. Configuring Firewalls

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

- Respond to incoming HTTPS (port 443) network traffic, as 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 platform's currently installed set of packages (this can be disabled, if desired) and downloading of user-specified dependencies for the package being assessed. The package's build system might also require access to the internet.

Any firewall(s) 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
```

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 the 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 directory of the SWAMP-in-a-Box installer. Copy the iptables file from that directory to /etc/sysconfig. Then restart the iptables service. 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 service. 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
```

3. Installing SWAMP-in-a-Box

3.1. Before You Begin

- 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 certificates 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 database that are installed as part of

SWAMP-in-a-Box:

  - Database root password: SWAMP-in-a-Box uses MariaDB as its database backend. This is the password for the database's root user. It may be different from the host operating system's root user's password (the database maintains a 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 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 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.

- We strongly recommend running yum update (as root or using sudo) to ensure that any software installed on the SWAMP-in-a-Box host is up to date. This is especially important if there has been a new release of CentOS (or whatever similar Linux distribution is installed on the host) since the host was initially set up. The steps below will likely cause a partial update to the new release, which might leave the host in an inconsistent state.

3.2. Obtain the SWAMP-in-a-Box 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.

3.3. Extract the Installer

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

- 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 will use <installer-dir> to refer to that directory.

3.4. Install SWAMP-in-a-Box's 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 corresponding to your host's distribution:

```
<installer-dir>/repos/CentOS-6/install-all.bash
<installer-dir>/repos/CentOS-7/install-all.bash
```

If your host has restricted access to the internet, see Installing Dependencies for a list of SWAMP-in-a-Box's dependencies so that you can determine how best to install them on the host. Continue with the steps below after you have installed the dependencies.

3.5. Run the Main SWAMP-in-a-Box Install Script

As `root` (or using `sudo`), run `install_swampinabox.bash`:

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

The script will prompt you for the hostname and passwords listed above.

The script's output will be saved to a file, the exact location of which will be listed at the end of the install. If the install is unsuccessful, the log will be helpful in determining the cause.

3.6. Verify that the Install Was Successful

1. In a web browser, navigate to https://<SWAMP-in-a-Box-hostname>/.
2. Sign in with the administrator account/user:
   - Username: admin-s
   - Password: <the one entered during the install process>
3. Upload a package, create and run a new assessment of it, and view the results. Sample packages can be found in `<installer-dir>/sample_packages`; see the `README.txt` file in that directory for more information.

4. Upgrading SWAMP-in-a-Box

4.1. Significant Changes and Other Considerations

Changes that users of the SWAMP will see in each release of SWAMP-in-a-Box, such as new features and significant bug fixes, can be found in `/opt/swamp/doc/CHANGELOG.txt` on the SWAMP-in-a-Box host after the upgrade has completed. Changes and "gotchas" that are more relevant to the administrators of the SWAMP-in-a-Box host are listed below.

Changes Introduced in 1.31:

- The upgrade script no longer supports upgrading releases of SWAMP-in-a-Box prior to 1.29. Older systems can be upgraded to 1.29 or 1.30 first, and then upgraded to 1.31.

4.2. 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.
- 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 can be upgraded to 1.29 or 1.30 first, and then upgraded to 1.31.
- We strongly recommend running `yum update` (as `root` or using `sudo`) to ensure that any software installed on the SWAMP-in-a-Box host is up-to-date. This is especially important if there has been a new release of CentOS (or whatever similar Linux distribution is installed on the host) since the host was initially set up. The steps below will likely cause a partial update to the new release, which might leave the host in an inconsistent state.

4.3. Obtain the SWAMP-in-a-Box 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.

4.4. Extract the Installer

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

- `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 will use `<installer-dir>` to refer to that directory.

4.5. Install SWAMP-in-a-Box's 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 corresponding to your host's distribution:

```
<installer-dir>/repos/CentOS-6/install-all.bash
<installer-dir>/repos/CentOS-7/install-all.bash
```

If your host has restricted access to the internet, see Installing Dependencies for a list of SWAMP-in-a-Box's dependencies so that you can determine how best to install them on the host. Continue with the steps below after you have installed the dependencies.

4.6. Run the Main SWAMP-in-a-Box Upgrade Script

As `root` (or using `sudo`), run `upgrade_swampinabox.bash`:

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

The script will prompt you for the database's `root` user's password and create a backup of the SWAMP's databases before making any modifications to them. Specifically, the following files will be created in the directory from which you run the upgrade:

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

The script's output will be saved to a file, the exact location of which will be listed at the end of the upgrade. If the upgrade is unsuccessful, the log will be helpful in determining the cause.

4.7. Verify that the Upgrade Was Successful

1. In a web browser, navigate to https://<SWAMP-in-a-Box-hostname>/.

2. Sign in with the administrator account/user:
    - Username: admin-s
    - Password: <the one previously set for the `admin-s` user>
3. Upload a package, create and run a new assessment of it, and view the results. Sample packages can be found in `<installer-dir>/sample_packages`; see the `README.txt` file in that directory for more information.

4.8. Check 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. For example, if you have previously added on an additional assessment platform and there is an updated version of that platform available, you will have to download the new version separately and install it.

There is a utility script that you can run that that will identify such components for you. See the section on checking for updates for further information.

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.

> ℹ 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://mysib.example.org`.

5.2.1. Acquire the SSL Certificate

The first step is to acquire a 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 `mysib.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 mysib.example.org.private.key
   ```

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

   ```
   openssl req -new -key mysib.example.org.private.key -out mysib.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/mysib.example.org.cert
   SSLCertificateKeyFile /etc/pki/tls/private/mysib.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:+AES12
   8:+SSLv3:!aNULL:!eNULL:!LOW:!3DES:!MD5:!EXP:!PSK:!DSS:!RC4:!SEED:!IDEA:!ECDSA:kEDH:
   CAMELLIA128-SHA:AES128-SHA
   SSLHonorCipherOrder On
   ```

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) 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

- 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.
- Set the `MAIL_CONTACT_ADDRESS` to the email address you want to receive contact email from users. This is displayed in the content of some SWAMP emails.
- Set the `MAIL_SECURITY_ADDRESS` to the email address you want to receive security reports from users. This is displayed in the content of some SWAMP emails.

5.3.4. Modify `/opt/swamp/etc/swamp.conf` to Configure Outgoing Email

These settings enable email notifications that assessments have completed.

> ❗ There is a known bug that prevents these emails from being sent.

- Set `email.host` to your SMTP server.
- Set `email.arun.subject` to the subject line to be used for assessment completion notification emails.
- Set `email.from` to the name and email address you want to use as the sender of assessment completion notification emails.

- Restart the swamp service so your changes to swamp.conf are in effect by running the following command as root (or using sudo):

```
service swamp restart
```

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

Step 2: Modify /var/www/swamp-web-server/.env to configure contact message recipients. This is only necessary if outgoing email is enabled.

- 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.

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

Step 2: Modify `/var/www/swamp-web-server/.env` to configure security incident message recipients. This is only necessary if outgoing email is enabled.

- 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 3rd-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.

6. Installing Additional Components

6.1. Installing Additional Assessment Platforms

The SWAMP-in-a-Box installer includes only the Ubuntu 16.04 platform for performing assessments. For C/C++ packages, additional platforms are available, including releases of CentOS, Debian, Fedora, Scientific Linux, and older releases of Ubuntu. (Packages for other languages will always be assessed on Ubuntu 16.04.)

6.1.1. 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.

6.1.2. 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 https://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.

6.2. Installing Additional Assessment Tools

The SWAMP-in-a-Box installer includes variety of tools for assessing packages. For C/C++ packages, two additional tools can be installed: CodeSonar and Coverity.

6.2.1. About 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 2017 GrammaTech, Inc. CodeSonar is a registered trademark of GrammaTech, Inc. All rights reserved.

6.2.2. About Coverity

SWAMP-in-a-Box can be used with Synopsys Static Analysis (Coverity). Synopsys Static Analysis is an accurate and comprehensive static analysis solution for finding critical quality defects and security violations. Its high-fidelity analysis delivers business relevant findings for developers and security audit teams alike. Synopsys' SAST solutions are uniquely designed to scale from safety-critical IoT software to global enterprise systems.

Contact information for obtaining Synopsys Static Analysis (Coverity) and licensing information for Synopsys Static Analysis can be found at:

- software-integrity-sales@synopsys.com,
- U.S. Sales +1-800-873-8193,
- International Sales +1-415-321-5237, or
- https://www.synopsys.com/software-integrity/security-testing/static-analysis-sast.html.

Synopsys Static Analysis (Coverity) is third party software maintained by Synopsys, Inc. Copyright 2017 Synopsys, Inc. Synopsys Static Analysis (Coverity) is a registered trademark of Synopsys, Inc. All rights reserved worldwide.

6.2.3. 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 one or both of the 32-bit and 64-bit Linux archives for the tool you wish to install from its vendor.

  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 Coverity should be named (the .sh "installers" will not work):

```
cov-analysis-linux-<version>.tar.gz    (32-bit)
cov-analysis-linux64-<version>.tar.gz  (64-bit)
```

- You will need the hostname and port for the tool's license server. (Contact the tool's vendor if you need help setting up and configuring the license server.) The SWAMP-in-a-Box host will need to be able to contact the license server in order to successfully perform assessments using the tool; the scripts used below will not modify any firewall configurations.
- You will need to know or decide the maximum number of simultaneous instances of the tool the SWAMP may run.

6.2.4. Procedure

1. 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 coverity> \
       --tool-version <version> \
       --installer-linux32 <path to 32-bit archive> \
       --installer-linux64 <path to 64-bit archive>
   ```

   Specify gt-csonar or coverity 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 it will likely take several minutes to complete due to the size of the installers.

2. As root (or using sudo), run the install_tool script with the --add option, providing the version of tool being installed and the path to the SWAMP tool archive file produced in the previous step. (The first option should be --codesonar or --coverity, depending on the tool being installed.)

   ```
   /opt/swamp/bin/install_tool \
       [--codesonar or --coverity] \
       --add \
       --tool-version <version> \
       --tool-archive <path to SWAMP tool archive file>
   ```

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

3. As root (or using sudo), run the install_tool script a second time, this time with the --configure option, providing the hostname and port of the tool's license server, and the maximum number

of simultaneous instances of the tool that may run.

```
/opt/swamp/bin/install_tool \
    [--codesonar or --coverity] \
    --configure \
    --license-server-host <hostname of the license server> \
    --license-server-port <port number> \
    --limit <max number of simultaneous instances>
```

6.2.5. Managing the Installed Versions of CodeSonar and Coverity

- 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 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 run the `make_swamp_tool` script, 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
    [--codesonar or --coverity] \
    --replace
    --tool-version <version>
    --tool-archive <path to gt-csonar-<version>.tar.gz file>
```

- 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
    [--codesonar or --coverity] \
    --remove
    --tool-version <version>
```

6.3. Installing Additional Viewers

For viewing the results of an assessment, the SWAMP instance installed by SWAMP-in-a-Box includes a "native" viewer and additionally provides a link to download the "raw" results as a SCARF `.xml` file.

It is also possible to install a SWAMP-specific version of Code Dx for viewing results. (SWAMP-in-a-Box currently does **not** support integrating with an existing Code Dx installation.)

6.3.1. About 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 a 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.

6.3.2. 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.

  Note that the SWAMP officially supports version 1.8.3 of Code Dx. Later versions might work, but the experience for end users will be significantly different from 1.8.3.

6.3.3. 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 to the database and make it available in the SWAMP.

7. Administrative Commands

7.1. 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. For example, if you have previously added on an additional assessment platform and there is an updated version of that platform available, you will have to download the new version separately and install it.

To check whether updated assessment platforms are available, run the following command (it does not require `root` access):

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

Assessment platforms are currently the only component of SWAMP-in-a-Box that are not necessarily updated by the SWAMP-in-a-Box upgrade script.

7.2. Updating the Host's Hostname

During the SWAMP-in-a-Box install process, the hostname specified for the host (usually, the host's detected hostname) is set in various configuration locations. When the host's hostname changes, those configuration locations need to be updated in order for the system to continue functioning correctly. To do so, as `root` (or using `sudo`), run the following command:

```
/opt/swamp/bin/swamp_set_web_host <new hostname>
```

7.3. Managing the `swamp` System Service

SWAMP-in-a-Box installs a collection of daemons that run on the host, all managed by the `swamp` system service. These daemons must be running in order to perform assessments. The `swamp` service, and by extension all of the daemons, may 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
```

7.4. Other Components

SWAMP-in-a-Box also makes use of the Apache HTTP Server, HTCondor, and MariaDB. For instructions on how to interact with or administer Apache HTTP Server, HTCondor, and MariaDB, refer to the specific 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.

8. Documentation for Users of the SWAMP

Documentation for users of the SWAMP includes:

- SWAMP User Manual
- Status.out and Debugging SWAMP Failures

Links to these documents can be found on the Help page of the SWAMP web application.

9. Support and Contact Information

We welcome your feedback and contributions at:

- Email: sib@continuousassurance.org
- Phone: +1 (317) 274-3942

We also host a mailing list for the user community:

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

Appendix A: Installing Dependencies

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

- HTCondor 8.6,
- 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.

> In the sections below, `<installer-dir>` refers to the directory containing the SWAMP-in-a-Box installer. See Installing SWAMP-in-a-Box and Upgrading SWAMP-in-a-Box.

A.1. HTCondor 8.6

The set-up scripts configure and download HTCondor from the repository hosted by the University of Wisconsin-Madison, generally following the instructions provided on the project's home page (https://research.cs.wisc.edu/htcondor/index.html). The specific packages installed are `condor-all` and its dependencies.

The following distribution-dependent scripts will install HTCondor using the process described above:

```
<installer-dir>/repos/CentOS-6/install-htcondor.bash
<installer-dir>/repos/CentOS-7/install-htcondor.bash
```

A.2. MariaDB 5.5

For CentOS 6, the set-up scripts 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 download MariaDB from CentOS's default repositories. The specific packages installed are `mariadb`, `mariadb-server`, `mariadb-libs`, and their dependencies.

The following distribution-dependent scripts will install MariaDB using the process described above:

```
<installer-dir>/repos/CentOS-6/install-mariadb.bash
<installer-dir>/repos/CentOS-7/install-mariadb.bash
```

A.3. PHP 7.0

The set-up scripts 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 distribution-dependent scripts will install PHP using the process described above:

```
<installer-dir>/repos/CentOS-6/install-php.bash
<installer-dir>/repos/CentOS-7/install-php.bash
```

A.4. Other Assorted Utilities

In addition to HTCondor, MariaDB, and PHP, the set-up scripts download assorted software packages from CentOS's default repositories. The specific packages installed are

- `ant`,
- `bind-utils`,
- `git`,
- `httpd`,
- `libguestfs`,
- `libguestfs-tools`,

- `libguestfs-tools-c`,
- `libvirt`,
- `mod_ssl`,
- `ncompress`,
- `patch`,
- `perl`,
- `zip`,

and their dependencies.

The following script will install these packages and perform additional, necessary configuration of the SWAMP-in-a-Box host. It must be run after HTCondor, MariaDB, and PHP are installed, as described above.

```
<installer-dir>/repos/common/install-and-configure-deps.bash
```

Appendix B: License and Notices

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

Additional notices for the SWAMP can be found at the end of this section.

```
                                 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.
```

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. More details about 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 2010-2018 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_mysql_sys library, an open source library developed by Roland Bouman and Bernardo Damele A.G. and licensed under LGPL v3.0, the GNU Lesser General Public License v3.0 (https://www.gnu.org/licenses/lgpl-3.0.en.html). Copyright 2007, 2008-2009 Roland Bouman and Bernardo Damele A.G. All rights reserved.