Security

Reporting security issues

Please report any security issues you find in OpenDaylight to: security@lists.opendaylight.org

Anyone can post to this list. The subscribers are only trusted individuals who will handle the resolution of any reported security issues in confidence. In your report, please note how you would like to be credited for discovering the issue and the details of any embargo you would like to impose.

OpenDaylight - Vulnerability Management

Glossary

Term	Definition
Embargo	A time period where vendors have access to details concerning the security vulnerability, with an understanding not to publish these details or the fixes they have prepared. The embargo ends with a coordinated release date ("CRD"). (from source)
Subject matter expert	A developer or other specialist who can provide contextual information that helps to determine the validity and impact of a potential security vulnerability.
Peer reviewed	In the context of a patch, the term peer reviewed refers to the patch having been reviewed by the security response team and any other relevant key stakeholders. There is not yet a strict definition of the number of people who need to have reviewed the patch, or how they provide sign off.
Downstream stakeholder	An organization that builds products based on OpenDaylight. These products may be free, commercial, or for internal usage.

Security Response Procedure

Reference procedures

In an attempt to avoid re-inventing the wheel, the OpenDaylight vulnerability management process borrows unashamedly from the following procedures:

- The Linux Kernel process for reporting security issues
- The OpenStack vulnerability management process

- Recommendations for a minimal security response process

Security supported projects

As OpenDaylight is a hive of innovation and change it is necessary to outline the criteria for a project under the OpenDaylight umbrella to be security supported, and to be subject to the vulnerability management process.

The OpenDaylight project lifecycle & release process references six succinct stages that may categorise a project. These categories have been used to outline the recommended security support status for projects below.

Project Type	Status
Proposal	Not security supported.
Bootstrap	Not security supported.
Incubation	Not security supported, but may receive patches for high-impact vulnerabilities.
Mature	Subject to audit prior to receiving security supported status.
Core	Security supported
Top Level	Top Level projects are collections of sub-projects, and are therefore out of scope for support.
Archived	Not security supported.

Before an OpenDaylight project is considered security supported it needs to undergo a security audit and have any issues addressed. The OpenDaylight security response team will be responsible for performing audits in a timely fashion. Projects can prepare for an audit by using the following maven plugins, and resolving any issues they detect:

- Dependency-Check
- Victi.ms
- find-sec-bugs

Once receiving security supported status, all security issue will need to be addressed under embargo and have a security advisory issued.

Top level projects should try to provide at least one developer to work on the security team. Having connections within the development organization along with understanding of the implementation details is very useful when triaging security issues.

Security supported versions

A commitment needs to be made for all security supported projects within OpenDaylight to backport security fixes to previous releases of the products affected by the flaw. Backporting is only required for previous releases that are still maintained, according to the release engineering current maintenance strategy.

Note: The security team needs to provide accurate information about the version the flaw was first introduced so that vendors operating still maintaining older product lines can backport fixes outside of the upstream maintenance window.

Third party components

Third party components are included in many OpenDaylight projects as bundled dependencies. Vulnerabilities affecting third party components are in scope for this process if and only if they are notable. The OpenDaylight security response team determines at its own discretion which vulnerabilities are considered notable. As a rough guide, vulnerabilities that allow for remote code execution, authentication bypasses, or a complete denial-of-service, would usually be considered notable. Other vulnerabilities would not.

Vulnerability Management Workflow

Workflow for private security issues

See thumbnail at right.blocked URL

Reception

A public page must be made available detailing the OpenDaylight vulnerability management process, and providing a single point of contact for contacting the security team. Preferably this should be a private email list that only members of the OpenDaylight security team have access to. The public GPG keys associated with the email queue must also be published to allow reporters to GPG encrypt the email.

The OpenDaylight security team must also monitor development mailing lists and bug creation feeds to ensure that there are no issues that have been publicly reported which need to be treated as a security flaw. Should such a situation exist the *public security issue workflow* needs to be followed.

Upon receiving a privately reported security issue the OpenDaylight security team needs to complete the following tasks.

Extent of disclosure:

- Original Reporter
- OpenDaylight Security Team

Next Steps:

1. Send *reception confirmation email*
2. Create a JIRA issue with a Security Level of 'Security Issue', under the affected project.
3. Add reporter to private security bug

Triage

The bug must then be confirmed to be a security problem. This may require the inclusion of a subject matter expert to determine if the problem needs to be treated as a security flaw. If the bug is determined not be a security issue then a statement should be added indicating why, the bug should then be opened by setting the JIRA 'Security Level' field to 'Not a security bug' and fixed by following the standard OpenDaylight development process.

Should all parties agree that the issue is a security flaw then all parties need to work on determining the affected product, assessing the risk to OpenDaylight users, and proposing a fix to the flaw. All of this work **must** be done under embargo. Proposed fixes must not be committed to SCM, and the problem should not be discussed outside of those that have been added to the bug. If patch files need to be shared for review, it is recommended to use the JIRA issue with a Security Level of 'Security Issue'.

Extent of disclosure:

- Original Reporter
- OpenDaylight Security Team
- Subject matter expert (optional)

Next Steps (status: confirmed):

1. Post the *confirmed security issue* notification in the bug
2. Determine which versions of the product are affected by the flaw
3. Draft an impact description
4. Confirm that the original reporter wants to be credited for finding the flaw
5. Propose a fix / patch for the flaw
6. Get the patch peer reviewed

Next Steps (status: non-security):

1. Post a statement for non-security issues in the bug
2. Change the JIRA bug 'Security Level' to 'None'
3. Follow the normal OpenDaylight development process to get the issue fixed if necessary

Pre-disclosure

When a patch has been developed and peer reviewed it is then possible to start planning on how and when to announce the issue. This involves agreeing on a disclosure date and notifying any downstream stakeholders. Extent of Disclosure:

- Original Reporter
- OpenDaylight Security Team
- Subject Matter Expert (optional)
- Downstream stakeholders

Next Steps:

1. Send CVE *request email* to [kseifried@redhat.com kseifried@redhat.com]
2. Agree on disclosure date with original reporter. This will most likely need to fall on a Tuesday, Wednesday, or a Thursday. It should also be set sufficiently in the future to allow downstream stakeholders enough time to assess their risk. (3-5 business days). Also ensure a developer is available at that time to push up the fix.
3. Send *advanced notification* email to list of downstream stakeholders.
4. Re-test the patch. Ensure that it still applies to the various branches and that all unit tests pass.

Disclosure date

When the coordinated disclosure date has been reached the assigned member from the OpenDaylight security team must perform the following tasks.

Extent of Disclosure:

- Everybody. The issue will now be public.

Next steps:

1. Re-test the patch and make sure all unit tests pass.
2. Change the 'Security Level' of the JIRA bug to 'None'.
3. Coordinate the submission of the patch. The fix should be fast tracked as it has already been peer reviewed.
4. When the commit has been merged into the code an announcement must be sent individually to the following mailing lists: oss-security@lists.openwall.com, opendaylight-announce@lists.opendaylight.org, security-announce@lists.opendaylight.org

.

Post-disclosure

Post disclosure the standard development process applies. Some optional additional tasks that the security team could undertake would be:

- Convert the advisory publication to CVRF format and publish on a separate CVE stream
- Calculate the CVSS2 score for the flaw
- Determine the appropriate CWE for this flaw
- Write an automated reproducer of the flaw and add it to the regression tests
- Write an static analysis / lint rule to detect the pattern that lead to the flaw
- Ensure the correct CVE is listed in the release notes for the next version of the affected product.
- Add an entry to the Security Advisories page.

Handling public security issues

What is considered public?

- Any comment on a public forum, whether it be a mailing list, irc, twitter, or news group, that discloses the details of the flaw.
- Any commit or review comment that indicates that the change may be security related.

Public security issue workflow

There will be occasions where the vulnerability management workflow process is either not followed, or at some stage a party leaks the details of the flaw. In these cases the workflow in the thumbail to the right is applicable.

Communication

Message formatblocked URL

All messages communicated formally by the OpenDaylight security team should be in a well formed YAML format. This includes statements in bugs, as well as published advisories. This approach will provide a foothold for future reporting, automation and conversion to more standardised formats such as CVRF.

Reception confirmation email

Upon reception of a security report the OpenDaylight security team needs to clearly indicate the expectation of how the issue will be handled.

```
Thank-you for your security report.

The OpenDaylight security team has created a private security bug
to track this issue. Please provide us with your OpenDaylight
Linux Foundation ID so we can add you to the JIRA bug. All communications
and decisions about how this issue will be handled will be recorded
on the this bug to provide proper tracking.

{jira_issue_url}

Regards,

--
{opendaylight_security_team_member}
OpenDaylight Security Team
```

Confirmed private security issues

Clear instructions need to be provided to all parties involved with the fix as to how the issue needs to be fixed. When the flaw is confirmed, the following statement should be added to the bug by a member of the security team.

```
#security-status: confirmed

This issue has been confirmed as a security vulnerability in
{ product } and is to be fixed under the OpenDaylight embargoed
security vulnerability process. Please do not discuss or
disclose details about this flaw prior to the agreed disclosure
date (TBA). All decisions, discussions, and proposed patches
and reviews are to be done via this tracking bug only.
```

If are you unsure of this process please refer to {opendaylight_security_process_url} for more detailed instructions.

Confirmed public security issues

When an issue is leaked

```
#security-status: confirmed-leaked

This issue has been confirmed as a security vulnerability in
{ product }. Unfortunately the details of this flaw have been
made public { reference_to_leak }. Therefore it cannot be
fixed under the OpenDaylight embargoed security vulnerability
process. As this issue is now public it is important that the
flaw is addressed in a timely manner. The OpenDaylight security
team will ensure that a CVE is assigned for this issue.
```

When an issue was not reported privately

```
#security-status: confirmed-public

This issue has been confirmed as a security vulnerability in
{ product }. As this issue was originally a public report it
cannot be fixed under the OpenDaylight embargoed security
vulnerability process. As this issue is public it is important
that the flaw is addressed in a timely manner. The OpenDaylight
security team will ensure that a CVE is assigned for this issue.
```

Impact description

The impact description needs to provide an accurate description of the flaw, how it affects the product, and outline the version range that is affected. The impact description should be entered into the tracking bug and reviewed for correctness.

```
title: { impact description title }

reporters:
- name: { reporter_name }
company: { reporter_company }

affects:
- product: { product }
version: { version_range }

risk-assessment:
impact-rating: { impact_rating }

description:
{ reporter_name } from { reporter_company } reported a vulnerability
in { product } ....

Version
The version string must explicitly indicate the range of products
affected by the flaw. For example: "2.1.2=>3.1.0"
```

TBD: This approach may be more versatile.

Risk Assessment

The security team should provide a judgement call for the severity of the issue for the most common use case of the project. Suggested impact rating categories:

- **Critical**: This rating is given to flaws that could be easily exploited by a remote unauthenticated attacker and lead to system compromise (arbitrary code execution) without requiring user interaction. These are the types of vulnerabilities that can be exploited by worms. Flaws that require an authenticated remote user, a local user, or an unlikely configuration are not classed as Critical impact.
- **Important**: This rating is given to flaws that can easily compromise the confidentiality, integrity, or availability of resources. These are the types of vulnerabilities that allow local users to gain privileges, allow unauthenticated remote users to view resources that should otherwise be protected by authentication, allow authenticated remote users to execute arbitrary code, or allow local or remote users to cause a denial of service.
- **Moderate**: This rating is given to flaws that may be more difficult to exploit but could still lead to some compromise of the confidentiality, integrity, or availability of resources, under certain circumstances. These are the types of vulnerabilities that could have had a Critical impact or Important impact but are less easily exploited based on a technical evaluation of the flaw, or affect unlikely configurations.
- **Low**: This rating is given to all other issues that have a security impact. These are the types of vulnerabilities that are believed to require unlikely circumstances to be able to be exploited, or where a successful exploit would give minimal consequences.

Note: Formal methods such as CVSS may follow.

Description : The description must endeavour to accurately depict the nature of the flaw. Information that should be included must indicate the attack vector that is exposed by the flaw and the initial access level required by the attacker. Where applicable advice on how an operator may audit for abuse of the flaw within their environment.

CVE Request

To ensure proper traceability a CVE identifier needs to be requested from a CNA. An email requesting a CVE should be sent to either cve-assign@mitre.org or secalert@redhat.com. The email must be GPG-signed and GPG-encrypted.

```
A vulnerability was discovered in OpenDaylight (see below). In order to
ensure full traceability, we need a CVE number assigned that we can attach
to private and public notifications. Please treat the following information
as confidential until further public disclosure.

{ impact_description }

Thanks in advance,

--
{ opendaylight_security_team_member }
OpenDaylight security team
```

Advanced notification

In accordance with a coordinated disclosure model, large operators or vendors selling a derivative of OpenDaylight software should be given advanced notification of security issues to patch their products prior to making the details of the flaw public. A timeline of 3-5 working days should be allowed for stakeholders to assess the impact on their products and services. Advanced notification is provided via the security-prerelease@lists.opendaylight.org list. Membership of this list is restricted to members of the OpenDaylight foundation. Members are not added to this list automatically. To request to be added to the list, email security@lists.opendaylight.org. If any member of the security-prerelease list violates an embargo by releasing information or patches prior to the specified embargo date, they will be immediately removed from the list. They may appeal this decision if it can be shown that the embargo violation was accidental.

```
This is an advance warning of a vulnerability discovered in OpenDaylight,
to give you, as downstream   stakeholders, a chance to coordinate the
release of fixes and reduce the vulnerability window. Please treat the
following information as confidential until the proposed public
disclosure date.

{ impact_description }

Proposed patch: See attached patches. Unless a flaw is discovered in them,
these patches will be merged to { branches } on the public disclosure date.

CVE: { cve_id }

Proposed public disclosure date/time: { disclosure_date }. Please do not
make the issue public (or release public patches) before this coordinated
embargo date.

Regards,

--
{ opendaylight_security_team_member }
```

Advisory

The advisory notification builds on the existing impact description however it includes all the relative details of the fix. Each advisory should be given a unique identifier. This will be of the format <year>-<seq>. Where seq is a number that is incremented for each advisory issued for a given year.

```
opendaylight-advisory: {odsa_id } cve: { cve_id } advisory-date: { date_announced }

{ impact_description }

changes: # { release_name }
branch: { branch }
commit: { commit_hash }
review: https://git.opendaylight.org/gerrit/#/c/{review_id}

release-notes:
This fix will be included in the { milestone } development milestone
and in a future { next_stable } release.

bug-tracker: https://bugs.opendaylight.org/show_bug.cgi?id={ bugzilla_id }

--
{ opendaylight_security_team_member }
OpenDaylight Security Team
```

Statement for non-security issues

In the cases where a potential security flaw has been reported but the OpenDaylight security team have determined that it is not to be fixed under the embargoed security process a statement indicating why this decision has been made.

```
#security-status: wontfix
```

statement:

```
The OpenDaylight security team has analysed this report and determined
that it is not a security issue.

{ reason }

This bugs privacy status will now be changed to public, and the bug should
be fixed following the normal development process. We would like to thank
{ reporter } for bringing this issue to our attention.
```

Information to include in commit message

The commit message for a vulnerability fix should include the following tag on a line by itself:

```
#security-fix: { cve_id }
```