ONAP Vulnerability Management - deprecated

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.
Security SME	A security SME is a specialist who is familiar with the ONAP security vulnerability procedures and security in general
Peer reviewed	In the context of a patch, the term peer reviewed refers to the patch having been reviewed by the ONAP vulnerability sub-committee 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.

Security Response Procedure

Reference Procedures

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

- The Linux Kernel process for reporting security issues
- The OpenDaylight vulnerability management process
- Recommendations for a minimal security response process
- The fd.io vulnerability management process

Operating Structure

This activity, is approved and supported by the ONAP TSC and operates under the ONAP vulnerability sub-committee. The sub-committee functions are as described below. The committee has a chair, appointed by the membership from among the membership, who is responsible for seeing that work proceeds and serving as a point of contact for the TSC and community to the security sub-committee. The chair and membership, as well as pointers to this charter and the relevant email lists are document at link-to-page.

Security supported projects

All ONAP projects are currently in scope for vulnerability support. The participants of the ONAP projects are expected to support the ONAP vulnerability procedures when required.

Security supported versions

All versions of ONAP still supported by the project, and affected by the security issue, must be patched. This will usually start with the latest version of an affected project. Following that, the vulnerability sub-committee will work with downstream maintainers to ensure that the patch is applied to all maintained and affected versions.

Note: The ONAP vulnerability sub-committee 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 (i.e. dependencies) are only in scope for security support if they are statically compiled or otherwise bundled by an ONAP project. Dynamically linked dependencies should patch security issues independent of ONAP.

Dependencies on managed functions (eg. VNFs)

Vulnerabilities of managed functions (e.g. VNFs) are out of the scope of ONAP, however if a ONAP vulnerability has a dependence with a managed function, the managed functions vulnerability procedures will be used to coordinate the issue.

Vulnerability Management Workflow

Workflow for private security issues

Reception

A public page must be made available detailing the ONAP vulnerability procedures, and providing a single point of contact for contacting the ONAP vulnerability sub-committee. This should be a private email list (onap-security@lists.onap.org) that only members of the ONAP vulnerability sub-committee have access to.

The ONAP vulnerability sub-committee as well as the developers should 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 ONAP vulnerability sub-committee needs to complete the following tasks.

Extent of disclosure:

- Original Reporter
- ONAP vulnerability sub-committee

Next Steps:

1. Send *reception confirmation email*
2. Create private security bug
3. Add reporter to private security bug
4. Add project security contact to help triage the flaw

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 and fixed by following the normal development process.

Should all parties agree that the issue is a security flaw then all parties need to work on determining the affected code, assessing the risk to users, and proposing a fix to the flaw. All of this work **must** be done under embargo. Proposed fixes must not be committed to Source Code Management (SCM), and the problem should not be discussed outside of those that have been added to the bug.

Extent of disclosure:

- Original Reporter
- ONAP vulnerability sub-committee
- Subject matter expert (optional)

Next Steps (status: confirmed):

1. Post the *confirmed security issue* notification in the bug
2. Determine which versions of the project are affected by the flaw
3. Draft an impact description
4. Confirm whether 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 bugs security status from private to public
3. Follow the normal development process to get the issue fixed if necessary

Pre-disclosure

When a patch has been developed and peer reviewed, by a subject matter expert, it is then possible to start planning on how and when to announce the issue. This involves agreeing on a disclosure date. Extent of Disclosure:

- Original Reporter
- ONAP vulnerability sub-committee
- Subject Matter Expert (optional)

Next Steps:

1. Send CVE *request email* to NIST/NVD (TBD)
2. Agree on disclosure date with original reporter. This will most likely need to fall on a Tuesday, Wednesday, or a Thursday. Ensure a developer is available at that time to push up the fix.
3. 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 ONAP vulnerability sub-committee 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. Open the bug to the public
3. Coordinate the submission of the patch. The fix should be fast tracked as it has already been peer reviewed.
4. Create an advisory
5. When the commit has been merged into the code an announcement must be sent individually to the following mailing lists:, , **TBD**

.

Post-disclosure

Post disclosure the standard development process applies. Some optional additional tasks that the ONAP vulnerability sub-committee could undertake, or coordinate with the projects, 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 CVE for this flaw
- Write an automated reproducer of the flaw and add it to the regression tests
- Write a static analysis / lint rule to detect the pattern that lead to the flaw

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 a different workflow is applicable, as there is no longer any need to maintain an embargo. The private security issue workflow can be followed from the "Disclosure date" step onwards.

Communication

Reception confirmation email

Upon reception of a security report the ONAP vulnerability sub-committee needs to clearly indicate the expectation of how the issue will be handled.

```
Sample letter:

Thank you for reporting a security issue to the ONAP vulnerability sub-committee. We have created a private
security issue in JIRA to track this issue.
Please provide us with your JIRA username so we can add you to the issue.
All communications and decisions about how this issue will be handled will be recorded on this issue to provide
proper tracking.

  {jira_issue_url}

  Thanks

 { onap_vulnerability_ sub-committee _member}, on behalf of the ONAP vulnerability sub-committee
```

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 ONAP vulnerability sub-committee.

```
 #security-status: confirmed

This issue has been confirmed as a security vulnerability in { project } and is to be fixed under the ONAP
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 issue only.
```

Confirmed public security issues

When an issue is leaked

```
 #security-status: confirmed-leaked

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

Risk Assessment

The ONAP vulnerability sub-committee should provide a judgment 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.
- **High**: 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 high 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 endeavor 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.

```
A vulnerability was discovered in { project }, part of the ONAP project (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

 {onap_vulnerability_sub-commitee_member}, on half of the ONAP vulnerability sub-committee
```

Roadmap

Action Items

Topic	Assignee	Description	Status
organizational	Stephen	Send out a call for participation and form the ONAP vulnerability sub-committee	
infrastructure	Phil	Create a private mailing list for vulnerability management sub-committee	
organizational	Vulnerability committee	Elect a chair	
infrastructure	Phil	Enable private security issues in JIRA	
infrastructure	security sub-committee	Create a public page indicating contact information for the ONAP vulnerability sub-committee.	

documentation	Stephen	Create a public page detailing the vulnerability management process and how to report security problems to ONAP	
documentation	security sub-committee	Create a single page listing the security issues fixed in ONAP projects (advisories)	
communication	security sub-committee	Ensure the new security process is announced on all major mailing lists.	

References

1. Common Vulnerabilities and Exposure (https://cve.mitre.org/about/faqs.html)
2. CVE numbering authorities (https://cve.mitre.org/cve/cna.html)
3. CVE FAQ (https://cve.mitre.org/about/faqs.html#what_is_cve_identifier)