Security Issue Policies and Processes

Overview

Recent events have shown that formal policies and processes need to be put in place for handling of security issues. This spans the entire process, including but not limited to: reporting a bug, triage and risk assessment, implementing fixes, cutting releases, and making announcements. The plan is to form a group to discuss and draft these policies/processes and present them to the Technical Council (TC).

Who

The aim was to comprise this group with diverse representation to get as many viewpoints as possible. At the same time we're aiming to keep the group on the small side to keep things moving, mitigate scheduling conflicts, etc.

Name	Representation
John Malconian	DevOps
Julian Ladisch	Developers
Hongwei Ji	Developers
Johannes Drexl	SysOps
Ingolf Kuss	SysOps
Craig McNally	Tech Council / Developers
Adam Dickmeiss ???	Developers
Jakub Skoczen	Platform PO / Acrhitects
Stephen Pampell	SysOps
Michal Kuklis	UI Developers
Mark Deutsch	UI Developers
Vince Bareau	Tech Council / Architecture
Ian Walls	FOLIO Implementation
Peter Murray	Tech Council / Open Source Community
Philip Robinson	SysOps, User Management

When

Each Monday Starting October 7, 2019 - 11:00 - 12:00 ET in the "#security_policy_group" slack channel.

Security Policies and Processes.ics

How

Details are still TBD but so far the plan is to try to meet next week (10/7) and go from there. Ideally we can come up with a rough draft or outline from this first meeting with most of the high level decisions being made, and schedule additional meetings to refine as needed.

Proposed agendas:

10/7/2019 (Kickoff)

- Introduction & Goals (5 min)
- Quickly review the very high-level proposal from the TC (5 min)
- Create a bullet list / outline of policies & processes (~40 min)
- Start formalizing the outline into one or more policy documents (time permitting)

10/14/2019

- Review comments left in the notes section and refine
- Goal: Have a strawman in place to review for on 10/21

10/21/2018

- Review the rough draft/strawman

10/28/2019

- Review the revised document
- Discuss feedback from TAMU's security professional (if available)
- Goal: Address feedback and add final touches before presenting to the TC

11/11/2019

- 30m - Vetting Trusted Parties
- 30m - Accompanying recommendations - Related things that don't belong in the policy document, but we want to formally suggest to the TC and /or Security Team (once formed)
    - Public-facing documentation: reporting process, policy document, etc.
    - Highlight the need for a support policy
    - Any volunteers to pull this together?
- Goal: Address these two remaining issues and wrap things up.

What

Proposals/options taken straight from the TC meeting notes - a place to start:

- There should be a security team for FOLIO who can review and assess impact ('blast radius')
- That team decides things like
    - how public/private details of vulnerability are
    - urgency of action
    - Solicit input into specific actions to be taken
- Determine what the communication channels and timings are for all activity
- Likely a private/closed list for public installations to receive information about vulnerabilities
- Establish public policy for Security/Vulnerability patches
- Also determine which releases are supported going forward.

Notes

- Need a glossary - definition of relevant terms
- Dedicated channel for reporting security issues
    - email to security@folio.org
    - form?
    - both? Creates an issue in JIRA automatically?
    - Directly in JIRA?
    - Multiple ways!
    - Needs to be clear how this is done.
- Security team needs to be formed
    - Who comprises this team?
    - Distribution of security issues to wider (**trusted**) audience
        - How do you get onto this distribution list?
            - Sign up with security@duke.edu vs joe.shmo@duke.edu ?
            - needs to be clear how to get on this list.
        - Immediate(?) notification of the community (limited detail w/ score or risk level)
        - What level of detail are we disclosing here? Full disclosure; Access to the reported issues?
        - Julian Ladisch disagrees with keeping security issues private that are in third party software (PostgreSQL, vert.x, linux kernel, ...) that FOLIO uses and where a vulnerability has been published because anyone can take FOLIO source code and check whether it uses third party software with known vulnerabilities. FOLIO should immediately disclose those issue on issues.folio.org; then the general public can watch how FOLIO handles it.
    - Triage and risk assessment? Any good examples of criteria - (low / moderate / high) risk or impact?

            - https://en.wikipedia.org/wiki/Common_Vulnerability_Scoring_System
            - Investigation - reproduce reported issue, assess scope
                - A working proof of concept (PoC, exploit) that runs against a full FOLIO installation is *not* required to reproduce the issue, a unit test or code review is sufficient. Often it is more easy to fix the issue than writing a PoC. An issue is valid and should be fixed if it cannot be exploited in the current FOLIO version but may become exploitable in future versions.
                - App-specific, or system-wide?
                - Frontend or backend?
                - Hosting or Software?
                - Is it possible that sensitive information (e.g. PII) is exposed?
                - Priv. escalation issue?
                - Affected versions?
            - Retroactively assess risk of previously reported security issues - practice and will give an idea of how it works with the types of things that might be reported
            - risk/impact priority mappings
        - Involve responsible team...
            - Try to limit audience if possible
            - All modules NEED to have a responsible team
        - Security team works with responsible team to come up with workarounds / band-aids
            - Communicated to wider (trusted) audience
        - Full disclosure - needs to be defined, as well as other levels of disclosure
            - A fix is available
        - Release a fix
            - Backport - how far back?
                - Security team advises which versions should be fixed
                - Which versions are fixed needs to be part of notifications (obviously)
                - Outside the scope of this group:
                    - Bug fix release cycle? - Proposal coming soon - next sprint review?
                    - LTS versions?
- Idea: hire a consultant to help come up with the scaffolding for this?
    - Will we get this sort of feedback from the security audit (TBD)?

Links

Several links have been shared already in various conversations about this - these might be helpful as models or just as reference.

- https://rubyonrails.org/security/
- https://koha-community.org/security/
- https://fedoraproject.org/wiki/Security_Bugs
- https://wiki.duraspace.org/display/DSPACE/DSpace+Software+Support+Policy
- https://wiki.duraspace.org/display/samvera/Report+a+security+vulnerability
- https://www.debian.org/security/faq.en.html
- https://security-team.debian.org/security_tracker.html
- https://www.debian.org/security/audit/auditing.en.html
- https://www.kernel.org/doc/html/latest/admin-guide/security-bugs.html
- https://www.first.org/global/sigs/vulnerability-coordination/
- https://github.com/features/security
- https://en.wikipedia.org/wiki/Common_Vulnerability_Scoring_System
- http://oss-security.openwall.org/wiki/
- http://ocert.org/