Time Flies

Examining Timeliness in ALC_FLR.3

9th ICCC, 23rd September 2008

Steve Hill
CLEF Technical Manager
Logica
Chaucer House, The Office Park,
Springfield Drive, Leatherhead,
Surrey, KT22 7LP, U.K.
+44 1372-369618, steve.h.hill@logica.com

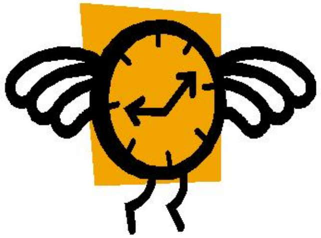
Duncan Harris
Senior Director, Security Assurance
Oracle Corporation
520 Oracle Parkway,
Thames Valley Park, Reading,
Berkshire, RG6 1RA, U.K.
+44 118-924-6201, duncan.harris@oracle.com

Flaw Remediation

- **In CC v2.3, ALC_FLR.3.9C**
  - *"The flaw remediation procedures shall include a procedure requiring timely responses for the automatic distribution of security flaw reports and the associated corrections to registered users who might be affected by the security flaw."*
- **In CC v3.1, ALC_FLR.3.6C**
  - *"The flaw remediation procedures shall include a procedure requiring timely response and the automatic distribution of security flaw reports and the associated corrections to registered users who might be affected by the security flaw."*

Guidance?

- **CC and CEM have no specific criteria to help an evaluator determine if flaw remediation procedures are timely.**
- **CEM work unit ALC_FLR.3-7 (3-11 in CCv2.3):**
  - *"…The issue of timeliness applies to the issuance of both security flaw reports and the associated corrections. However, these need not be issued at the same time. It is recognised that flaw reports should be generated and issued* ***as soon as*** *an interim solution is found, even if that solution is as drastic as Turn off the TOE. Likewise, when a more permanent (and less drastic) solution is found, it should be issued* ***without undue delay****. …"*
- **I.e., there is nothing to say how "as soon as" or "without undue delay" can be measured.**



Timeliness

- **Dictionary**
  - *"coming or occurring at a suitable time, an appropriate time or at the correct time; well-timed; opportune; in good time."*

Oracle’s Vulnerability Fix Lifecycle

- Quarterly Critical Patch Updates (CPUs)
  - Cumulative, multiple fixes in single product patches
  - Started January 2005, previously one-off fix patches
- Security Alerts for highest severity vulnerabilities
  - Only one released since CPUs started (August 2008)
- Vulnerability fix policy and process is public
  - “Oracle fixes significant security vulnerabilities in severity order”
  - Fixes included in products in this order:
    - Main code line (future major release)
    - Next patch sets (future minor releases)
    - CPUs (current supported releases)
  - Quarterly fixes, but no timeframe for individual bugs

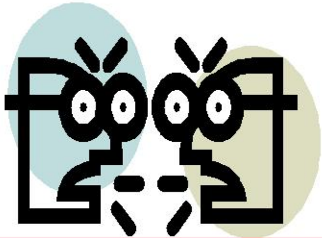
Oracle’s Latest Experience - 1

- CC EAL4 augmented with FLR.3 claimed for 3 products in
  Oracle Identity and Access Manager Suite
- Evaluators:
  1. Oracle’s quarterly Critical Patch Update (CPU) security vulnerability patch cycle isn’t timely.
  2. Some bugs might not be fixed within 3 months. They suspected that some take much longer and wanted us to provide evidence to show this was or was not the case.
- Oracle Response:
  – “Timely” means as quickly as a fix can be produced and tested across all affected versions and platforms
  – Triage system based on severity

Oracle’s Latest Experience - 2

- **Evaluators:**
  - Not good enough. Choices:
    1. Give us dates for the 6 most severe vulnerabilities from Oracle’s October 2007 CPU, or
    2. Drop to ALC_FLR.2
- **Oracle Response:**
  - Tell us what tasks in our vulnerability handling process lifecycle are unnecessary. We think they’re all necessary and they are done as quickly as possible. Reducing thoroughness of any task will result in poor patch quality. Customers will not apply a poor quality patch released just to meet a deadline.
  - Delays for any one vulnerability at any point in the lifecycle are identified, chased and escalated.
  - “Many hands make light work” isn’t the answer here

Mediation



- **Oracle:**
  - **Let’s ask the Certifier (CESG) to mediate.**
- **CESG:**
  - **Proposed criteria that evaluators could follow to decide if a developer is timely in their vulnerability handling. Criteria built exclusively around elapsed time since the problem was first reported to the developer.**
  - **Invitation to comment**
- **Oracle decided on two responses:**
  - **Short term for the particular evaluation**
  - **Long term alternative proposed criteria to measure timeliness**

Oracle’s Short Term Response

- Provided dates requested for 6 most severe vulnerabilities in Oracle’s October 2007 CPU to evaluators
- Added lengthy explanation covering:
  - Other (non-technical) factors that affect severity
  - All tasks performed in handling of a vulnerability and how long each takes on average
  - Internal policies on protecting sensitive information
  - Announcing vulnerability fix in advisory is a guarantee
  - Fixed but yet to be announced vulnerabilities
- Concluding
  - “Our bug response could not be quicker without sacrificing quality”

Short Term Argument Accepted

- **Argument fully accepted from “internal perspective”**
- **Less convinced from the “external perspective”**
  - **Customers, governments and security research community need more evidence that a developer’s vulnerability lifecycle is as quick as possible**

- **3 Oracle products awarded EAL4 augmented with ALC_FLR.3 certificates**

Long Term Alternative Proposal

- **In summary:**
  - **“Provide the quickest response possible that is commensurate with the need for a thorough investigation, the quality of the fix, and the measures taken to ensure high uptake of the fix.”**
- **Asked Logica to draft alternative proposal that:**
  - **Analysed requirements of CC and CEM on evaluators**
  - **Assessed intended spirit of CC and CEM**
  - **Considered large scale vulnerability fix lifecycle such as Oracle’s and other large developers, without neglecting capabilities of smaller developers**
  - **Recommended specific measurable action evaluators can take**

Alternative Proposal Detail - 1

- **Noted CC and CEM requirements require:**
  - **Requires Evaluator to check there is *evidence* of Flaw Remediation procedures**
  - **Does not require Evaluator to check *application* of Flaw Remediation procedures, even if this is available**
  - **This seems odd, but…**
- **Assessed spirit of CC and CEM:**
  - **Application of Flaw Remediation procedures to evaluated TOE results in changes that invalidate evaluation results (without a formal Assurance Continuity exercise)**
  - **Therefore checking existence of Flaw Remediation procedures gives some assurance without checking their application to the TOE itself which would then be meaningless**

Alternative Proposal Detail - 2

- **A vulnerability fix patch will be**
  - ***Too early*** **if it does not <u>correctly and completely</u> fix the vulnerability**
  - ***Too late*** **if it was available for release (correctly and completely fixed the vulnerability), but the developer did not release it to customers until after attackers started exploiting the vulnerability. I.e., not so much** ***Too late*** **but** ***Unduly delayed*****.**
  - ***Too frequent*** **if there are so many patches released that a customer (most of whom understandably want to test patches before applying them to a production system) cannot "consume" one before another appears**
- **Imposing an arbitrary timescale for release of a vulnerability fix implies that we know exactly when:**
  - **An acceptable risk becomes unacceptable to most or all customers**

Alternative Proposal Detail - 3

- **Possible Causes of Undue Delay**
  - **Failure to assign and prioritise adequate resource**
  - **Failure to track and ensure ongoing progress of a vulnerability through its fix lifecycle**
- **Possible Legitimate Reasons for Delay:**
  - **Engineering complexity, e.g. architectural redesign;**
  - **Testing scale, e.g. multiple affected versions, dependent product testing;**
  - **Emerging complexity, e.g. an apparently simple problem slowly appears more complex as more information emerges and as an investigation continues;**
  - **Fix combination, i.e. delaying release of a fix in a product because other fixes in the same product are not yet "release-ready" to avoid *too frequent* fixes.**

Proposed Interpretation

- **Developer's Flaw Remediation Procedures must show:**
  1. **Resources allocated to investigation of vulnerabilities and testing of fixes and workarounds are sufficient**
  2. **A vulnerability's calculated risk is used to prioritise its fix**
  3. **Target timescales for each vulnerability fix are calculated and set appropriately. (Note, this is for tracking progress to avoid a vulnerability getting "stuck" and to record legitimate reasons for delay.)**
  4. **Vulnerabilities are tracked against set timescale**
  5. **Fixes are sufficient to fix the vulnerability correctly and completely**
  6. **Testing ensures a fix is sufficient and thorough**
  7. **Fixes are released to balance high uptake by customers against undue delay**

Summary

- **Current Status:**
  - **Alternative proposal now a UK Draft Interpretation**
  - **To be discussed by CCDB**
  - **Hopefully soon to be:**
    - **A formal UK National Interpretation, or**
    - **Formally incorporated into CC/CEM**

- **Thanks to:**
  - **Logica evaluators, for being thorough (as requested!)**
  - **Oracle’s assigned CESG certifier and deputy certifier for mediating Oracle’s first potential major dispute**
  - **CESG for being open to Oracle’s objections and Logica’s alternative proposal to measure timeliness**