Issues, lessons learned through the eyes of JPCERT/CC on the vulnerability handling framework in Japan

Masaki Kubo, Takayuki Uchiyama
JPCERT Coordination Center
Vulnerability Coordination Group
Agenda

- Introduction to Vulnerability Handling Framework in Japan
- Current Issues
- Lesson Learned
- Moving Forward
Introduction to Vulnerability Handling Framework in Japan

- In Japan, handling activities are specified in “Information Security Early Warning Partnership”

- This partnership was created in accordance with the notification No. 235 issued in 2004 by the Ministry of Economy, Trade and Industry
  —Last updated in 2014

- Handling of website vulnerabilities are also governed here.
  —Today’s focus will be on product vulnerabilities
Timeline

November of 2003:
In response to the effects brought on by Blaster and Sasser worms, Ministry of Economy, Trade and Industry (METI) contracted the Information-technology Promotion Agency (IPA) to conduct a "Study Group on Information System Vulnerability Handling“

April 2004
Study results made public. Recommend that METI issue formal rules for handling vulnerability information. Rules should be generated in discussions with industry organizations and interest groups
Timeline

July 2004
METI issued “Standards for Handling Software Vulnerability Information and Others” to ensure appropriate handling of vulnerability-related information when a vulnerability is reported
—JPCERT/CC assigned to be the designated coordinator for handling vulnerability information
—Joint announcement of “Information Security Early Warning Partnership Guideline” from JPCERT/CC, IPA JEITA, JISA, CSAJ, JNSA

* JEITA - Japan Electronics and Information Technology Industries Association
* JISA - Japan Information Technology Service Industry Association
* CSAJ - Computer Software Association of Japan
* JNSA - Japan Network Security Association
Timeline

July 2004 (cont.)
JEITA / JISA jointly release “Vulnerability information handling guidelines for Product Developers”
JPCERT/CC releases “Bylaws for product developers” to receive vulnerability information also released
—These bylaws were created in an attempt to prevent developers from “just taking” vulnerability information without responding to requests

Each year, "Study Group on Information System Vulnerability Handling“ discusses issues, including operations of the framework to change things as necessary
Timeline

- Major changes that affected operations since the initial 2004 partnership guidelines
  - Notifications to vendors that use third-party libraries (2006)
  - Pre-notifications to critical infrastructure as necessary (2007)
  - Issues in protocol specifications or encryption algorithms are not to be handled (2009)
  - Enable disclosure of a developers list who do not respond (2009)
    - Process actually started in 2011
    - In 2014, guideline was amended to disclose such vulnerability reports
  - After one year, reporter can disclose vulnerability (2011)
Handling Framework Flow

- Domestic Report
- Other Reports
- IPA
- JPCERT/CC
- Vendor
- JVN
- Vendor Disclosure
Introduction to Vulnerability Handling Framework in Japan

In this framework, JPCERT/CC acts as the interface with the vendor
- Reports are received by Information-technology Promotion Agency (IPA)
- IPA interfaces with the reporter
- JPCERT/CC sometimes received reports directly (mostly from overseas researchers)

Domestic vendors need to be “registered” to receive vulnerability information
- Open source developers are exceptions
- Registered vendors are part of “multi-vendor” coordination
In addition to vendor coordination…

- Direct reports from reporters and security vendors

- Collaboration with other coordination centers
  - CERT/CC
  - NCSC-FI
  - CNCERT/CC
  - KrCERT/CC

**
Something new

- Publish list of “Non-responsive” vendors (2011)
  - Long process, just to get it started…
    - Various legal issues that needed sorting
    - Currently limited to issues that can be verified (tested)
    - An outside committee decides whether or not to publish

- In some cases, advisories are published for products developed by such a vendor

- List is updated quarterly
  - Information is uploaded in stages
    (Developer name, Reported product name / version, time limit)
List of unreachable developers

Product Developer Information

Overview
IPA (Information Technology Promotion Agency) and JPCERT Coordination Center are seeking contact from developers or related parties of software products that have been reported through The Information Security Early Warning Partnership.

Targeted Developers
Targeted developers are ones who have had software products reported through The Information Security Early Warning Partnership and have been unreachable through contact information posted on a website, etc. For the list of developers please see the below list.

Contact us: jvn@jvn.jp
Please include the “Inquiry Number” in the subject line.

Unreachable Developer List

<table>
<thead>
<tr>
<th>Inquiry Number</th>
<th>Developer Name</th>
<th>Developer’s Link</th>
<th>Initial List Date</th>
<th>Last Update</th>
<th>Other Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>DID#04630151</td>
<td>LunarNight Laboratory</td>
<td></td>
<td>15/09/18</td>
<td>15/12/25</td>
<td></td>
</tr>
<tr>
<td>DID#16838412</td>
<td>LunarNight Laboratory</td>
<td></td>
<td>15/09/18</td>
<td>15/12/25</td>
<td></td>
</tr>
<tr>
<td>DID#34961442</td>
<td>Remember The Milk</td>
<td></td>
<td>15/09/18</td>
<td>15/12/25</td>
<td></td>
</tr>
<tr>
<td>DID#89907900</td>
<td>CutePHP.com</td>
<td></td>
<td>15/09/18</td>
<td>15/12/25</td>
<td></td>
</tr>
<tr>
<td>DID#11985852</td>
<td>CutePHP.com</td>
<td></td>
<td>15/09/18</td>
<td>15/12/25</td>
<td></td>
</tr>
<tr>
<td>DID#99539461</td>
<td>Medieval Software</td>
<td></td>
<td>15/09/18</td>
<td>15/12/25</td>
<td></td>
</tr>
<tr>
<td>DID#90189163</td>
<td>yamagoya</td>
<td></td>
<td>15/09/18</td>
<td>15/12/25</td>
<td></td>
</tr>
</tbody>
</table>
Stats: private reports and OSS

Private Reports and OSS

<table>
<thead>
<tr>
<th>Year</th>
<th>Private Reports</th>
<th>Open Source Software</th>
</tr>
</thead>
<tbody>
<tr>
<td>2005</td>
<td>112</td>
<td>33</td>
</tr>
<tr>
<td>2010</td>
<td>128</td>
<td>25</td>
</tr>
<tr>
<td>2015</td>
<td>428</td>
<td>194</td>
</tr>
</tbody>
</table>
Stats: # of advisories published

PUBLISHED ADVISORIES

2005 2010 2015

JVN  JVNVU  JVNTA  Total

129  49  21  4
181  66  27  188
338  146  338  188
What was expected to be achieved?

- Coordination between researchers and vendors through 3rd party organization
  - to avoid anonymous full-disclosure
  - researcher can tell someone responsible about what they found
  - ‘responsible’ actions taken by the vendor
  - study of root cause

- Standardization of vendor’s response to vulnerability
  - handling
  - disclosure
What has been achieved?

- Coordination between researchers and vendors through 3rd party organization
  - to avoid anonymous full-disclosure
    - yes: full-disclosure in Japanese rarely seen
  - researcher can tell someone responsible about what they found
    - yes: even the low hanging fruit is handled with care
    - yes: researcher can stay anonymous to vendors
    - Is the framework becoming an impediment to the communication between vendor and researcher??
  - ’responsible’ actions taken by the vendor
    - yes: to some extent. but depends on who you’re talking about
      - new comers are always immature
  - study of root cause
    - not sure… same mistakes are repeated
What has been achieved?

- Standardization of vendor’s response to vulnerability
  - Probably not: vulnerability disclosure guideline was published in 2009
    - but only adopted by major vendors
  - no: “hiding” fixes
    - ’update module’ not ‘security fix’
    - ‘security enhancement’ not ‘vulnerability’
    - No advisories
    - Etc.

- Protecting researcher
  - Yes: JPCERT is a trusted entity (for the most part)
  - Vendors don’t threaten us as much (still receive threats sometimes)
Lessons learned

- While a lot of vendors are responsive, there are still many vendors are not responsive to vulnerability reports
  - No contact information
  - Will not respond to coordination center, etc.

- Some vendors do not want to publish
  - Publicity
  - Negative image, etc.

- Handling large quantity of cases ‘hides’ critical or high-impact cases
  - Currently “need” to handle each case equally
  - A case is a case
Lessons learned

- Reports on old versions tend to not get responses or “put off to the side”
  - Easy to understand support policies would make this easier
  - Should we be asking reporters to test against the most recent version?
  - Should we be asking vendors to fix every version of the product?

- Widely used third party libraries require lots of coordination (OpenSSL, Apache Struts, etc.)
  - Topic of various discussions
  - Vulnerability Coordination SIG
  - Try not to focus on this today
Issues / Limitations of current framework

- System Integrators (S) are out of the scope of the information sharing framework
- Framework designed to coordinate with “product developers”

Members of the early warning framework

System Integrators
Not all bad

- In the last 10+ years, lots of vendors have become receptive to vulnerability handling
  - Unfortunately, there are a lot left
  - What can we do to reach out?

- Creation of platforms for coordination
  - HackerOne
  - BugCrowd
  - Etc.

- Still receiving lots of reports
  - Lots of low impact reports (more on this later)

- Various community efforts discussing multi-party vulnerability coordination
Not all bad

More and more organizations are making policies related to vulnerabilities public
  —Point of Contact or Group for this information
  —What is a vulnerability?
  —How its handled
  —Severity Rankings (and priority)
  —What will / will not be published
  —(What constitutes a bug for a bug bounty)
  —Etc.
Thinking out loud

- How JPCERT should respond to vendors …
  — who won’t disclose vulnerability information to its users
  — who won’t disclose advisory properly
  — who tries every way to avoid public disclosure

- What statistic information would be valuable?
  — To convince organizations that disclosing vulnerabilities is NOT a bad thing

- Share the (emerging) pattern of vulnerability among multiple developers ---- secure coding / development
  — vulnerability of android apps
    - SSL/TLS certificate validation
    - path traversal in Zip file handling
  — Find way to convey common issues in related products before we receive reports on individual products
Thinking out loud

While the framework still serves its purpose, it needs to get “with the times”

— JPCERT/CC should become more of a facilitator in distributing vulnerability information as opposed to a ‘dedicated’ coordinator (coordinate as necessary)

- Allow reporter to directly interface with developer (assist with language barrier as necessary)

- JPCERT/CC can help guide coordination for any reporters or vendors that are new to the process
Thinking out loud

While the framework still serves its purpose, it needs to get “with the times” (cont.)

— Does coordinating reports on CMSs or PHP apps, old CGIs that have extremely small user bases help the community?

We need a metric besides “cases handled” or “JVN publications” to better represent the work that we do.

— While the framework requires a patch/update prior to publishing, should open-source products be subject to this same requirement?

Do they NEED to address vulnerabilities?

— Vendors fix the software, system integrators apply the fixes…

Adjusting the embargo period for products that are widely used in other products.
Any good ideas!? 

For inquiries on JVN: 
[jvn@jvn.jp](mailto:jvn@jvn.jp) 

For vulnerability reports 
[vuls@jpcert.or.jp](mailto:vuls@jpcert.or.jp) 

For any other vulnerability related inquires 
[vultures@jpcert.or.jp](mailto:vultures@jpcert.or.jp)