Using a Vulnerability Description Ontology for vulnerability coordination

- Removing the pain of repetitive analysis of vulnerability reports -

Masanobu Katagi, Takayuki Uchiyama (JPCERT/CC, JP), and Masaki Kubo (NICT, JP)
BIO

- Masanobu Katagi (JPCERT/CC - Vulnerability Coordination Group)
  - Responsible for vulnerability coordination at JPCERT/CC
- Takayuki (Taki) Uchiyama (JPCERT/CC - Technical Committee Member, Panasonic PSIRT)
  - Responsible for activities related to vulnerabilities (identification, analysis, coordination, disclosure)
- Masaki KUBO (Cybersecurity Laboratory, NICT)
  - Responsible for leading technical analysis of darknet monitoring of NICTER as well as NICT-CSIRT operation
Today’s talk

1. Pain Points in vulnerability coordination at JPCERT/CC
2. Vulnerability Description Ontology (VDO)
3. Application of VDO: JPCERT/CC’s case
About JPCERT/CC

Prevent - Vulnerability Coordination
  - Coordinate with developers on unknown vulnerability information
  - Secure Coding

Watch - Information gathering / analysis / sharing
  - Internet Traffic Monitoring
    - Alerts / Advisories

Respond - Incident Handling
  - Mitigating the damage through efficient incident handling
  - Information sharing to prevent similar incidents

Early Warning Information
  Information sharing with critical infrastructure enterprises, etc.

CSIRT Establishment Support
  Capacity building for internal CSIRTs in enterprises / overseas national CSIRTs

Industrial Control System Security
  Activities to protect ICS, such as incident handling and information gathering/sharing

Artifact Analysis
  Analysis on attack methods / behavior of malware (unauthorized program)

Domestic Collaboration
  Collaboration with various security communities in Japan

International Collaboration
  Collaboration with overseas organizations for smoother handling of incidents and vulnerabilities
JPCERT/CC - Vulnerability Coordination

Japan

Overseas

CERT/CC (US)
NCSC-FI
NCSC-NL

Overseas Coordination Centers

Information Security Early Warning Partnership

IPA

JPCERT/CC

End users

Corporate users

System Integrator

ISP

Developers

Media

Advisory (JVN)

Vendors

Reporters

Reporters
Bottlenecks in Coordination

- Sudden increase in vulnerability reports the last few years
  - 2.4 times more reports in 2016
  - Bottlenecks in JPCERT/CC coordination process
    - Delay of delivering reports to vendor
    - Increased risk for the vulnerable software
  - Urgent need to re-think coordination process

Total number of reported vulnerabilities by year (as of 4/25/2018)
Reconsideration of Coordination Processes

Lifecycle of Vulnerability Information at JPCERT/CC
Pain Point #1: Understanding a vulnerability report written in free text format

Free Format!

Triage

Prioritize

Coordinate

Disclose

Hard to interpret what is written in some reports

But I have to make decision based on my understanding ...
Pain Point #2: Extracting elements of information for scoring CVSS

I have to calculate CVSS score for prioritization

But I have to manually go through each parameter of CVSS while extracting corresponding parameter from the text ...
Pain Point #3: Going back to Pain Point #1

I’m coordinating more than a few dozen vulnerability at a time.

Can’t remember them all, so I need to go over the original report over and over again ... (sigh)
Pain Point #4: Writing an advisory

- Triage
- Prioritize
- Coordinate
- Disclose

Free Format!

Have to read the report Again!!
- Writing two JVN drafts (Japanese & English)
- Technical reviewer also need to read the original report
- Evaluate CVSS score and review of scoring by the reviewer
Problem Statement (1)

• Redundancy in coordination process causing:
  • Analysis of the same report (at least) twice throughout the process
  • Since only the original report is stored, the second analysis takes the same amount of time as the first
Problem Statement (2)

• Since vulnerability information is provided in a free format:
  • Technical aspects must be extracted
    • Affected products / versions
    • Vulnerability type / How to exploit / Effects / etc.
  • Requires interpretation of written language
    • What essentially means the same thing can be written in a million different ways
    • Language barriers can cause mis-interpretation of subtle nuances
Solution: Convert Reports into a Machine Readable Format

Automation: CVSS basic score

Automation: Creating JVN advisories

Free Format!

Machine Readable Format!!
How to convert free formatted vulnerability information into a machine readable format?
Today’s talk

1. Pain Points in vulnerability coordination at JPCERT/CC

2. Vulnerability Description Ontology (VDO)

3. Application of VDO: JPCERT/CC’s case
Recent CVE Activities

1999 Recent Activities Archives

July 27, 2000

- Tivoli Makes CVE Compatibility Declaration
  Tivoli Systems Inc., an IBM company, has declared that their SecureWay Risk Manager is CVE-compatible. For additional information about this and other CVE-compatible products, visit the CVE Compatible Products page.

July 21, 2000

- CVE Referenced in Computerworld Article
  CVE was referenced in a recent article on Computerworld.com entitled, "Security, the Way It Should Be". The article discusses various approaches to improving security and in a section on code review refers to CVE as "a widely accepted archive of security problems found in software and hardware" along with a link to the CVE web site.

How have we captured vulnerability information?

• almost 20 years industry experience in cataloging vulnerability
  • MITRE CVE project started in 1999
  • DoE/CIAC around 2000
  • CERT/CC Vulnerability started in 2000
  • JVN started around 2002
  • etc...

• Common elements of information
  • Title, summary, affected products, description, impact, patch, workaround...
Existing standardization efforts about describing vulnerability

• **Common Security Advisory Framework (CSAF) Version 1.2 (2017)**
  https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=csaf

• **Application Vulnerability Description Language (AVDL) v1.0 (2004)**
  https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=avdl

• **Vulnerability Data Model (2013)**
  https://www.ietf.org/archive/id/draft-booth-sacm-vuln-model-02.txt
VDO – Vulnerability Description Ontology

• Draft NISTIR 8138

Vulnerability Description Ontology (VDO): a Framework for Characterizing Vulnerabilities (2016)

• Goals of VDO
  • to enable automated analysis using metrics like CVSS
  • provide a baseline of the minimum information needed for a vulnerability management process
What is VDO?

• **Conceptual model** of vulnerability
  • Defines a **set of fundamental building blocks** of a vulnerability as well as their definitions, relationships and constrains
  • Helps you represent **semantics** of a vulnerability
  • Forces you to look at vulnerability in a VDO way

• **Information model** of vulnerability
  • VDO is **NOT**
    • a data model
    • advisory format
    • reporting format
Understanding of vulnerability information

Which software is affected?
• product name, version

Technical details?
• vulnerability type,
• attack surface
• conditions of exploitation
• difficulty of exploitation

Where an attack comes from?

Impact, Severity?
• Which sector uses it?
• Consequence if it is exploited

Building blocks (noun groups) of VDO

Product
Type
Impact Method
Entity Role
Barrier
Context
Attack Theater
Criticality
Scope
Building blocks of VDO

• VDO is composed of
  • **noun groups** ... key elements of vulnerability
    • noun group definitions
    • usage (mandatory, recommended, optional)
  • **noun group values** ... valid values are enumerated and values are chosen from them
    • noun group value definitions
  • **relationships** ... how each noun groups are related to each other

• Let’s take a look at the example...
Example of noun group - **Context**

- **Definition of Context**
  - the entity where the impacts are realized from successful exploitation

- **Possible Values**
  - Hypervisor
  - Firmware
  - Host OS
  - ...
  - Hardware

- **Relationships:** *Entity Role, Impact Method, Mitigation, Privilege Required, Victim Type*
  - Zero or many Entity Role values should be associated with Context.
  - One or many Impact Method values shall be associated with Context.
  - Zero or many Mitigation values may be associated with Context.
  - ...

Description of a vulnerability

Directory traversal vulnerability in the XCloner plugin 3.1.1 for WordPress and 3.5.1 for Joomla! allows remote administrators to read arbitrary files via a .. (dot dot) in the file parameter in a json_return action in the xcloner_show page to wp-admin/admin-ajax.php.
Mapping description to VDO

**Directory traversal** vulnerability in the **XCloner plugin 3.1.1** for WordPress and 3.5.1 for Joomla! allows **remote administrators** to **read arbitrary files** via a .. (dot dot) in the file parameter in a `json_return` action in the `xcloner_show` page to `wp-admin/admin-ajax.php`.

Technical details (specific to this case) necessary to create PoC code will be lost in VDO :-(

<table>
<thead>
<tr>
<th>Type</th>
<th>Product</th>
</tr>
</thead>
<tbody>
<tr>
<td>Entity Role</td>
<td>Context</td>
</tr>
</tbody>
</table>

**Product Context Barrier Type Impact Method Attack Theater Entity Role Scope**
Vulnerability: cve.mitre.org CVE-2014-8606
Provenance: http://www.vapid.dhs.org/advisories/wordpress/plugins/Xcloner-v3.1.1/
Scenario: 1
Type: cve.mitre.org CWE-22
Products:
cpe.nist.gov
cpe:2.3:a:xcloner:xcloner:3.1.1:*:*:*:*:wordpress:*:*
cpe:2.3:a:xcloner:xcloner:3.5.1:*:*:*:*:joomla!*:*:
Attack Theater: Remote
   Remote Type: Internet
   Barriers: Privilege Required
      Privilege Level: Administrator
      Relating to Context: Application
Context: Application
Entity Roles: Primary Authorization
Entity Roles: Vulnerable
Impact Method: Trust Failure
   Trust Failure Type: Failure to Verify Content
Logical Impact: Read(Direct)
   Scope: Limited
   Criticality: Low
Context: HostOS
Entity Roles: Secondary Authorization
Impact Method: Code Execution
Logical Impact: Read(Direct)
   Scope: Limited
   Criticality: High
The attacker is required to have administrator rights within the application prior to exploit
The Application is the initial authorization scope
Since the user is already an administrator of the application, the criticality is Low
The attack can read files on the HostOS. Since the file in the example supplied is etc/passwd the criticality can be High.

https://csrc.nist.gov/publications/detail/nistir/8138/draft
Goals of VDO

Coordination Body

Researcher/Reporter
- Shared vocabulary
- Formalized reporting

Vendor
- Lightweight triage

Statistics
- Enables detailed analysis of data

Advisory
- Become part of
- Automated advisory generation

Consumer
- Automation

JPCERT CC® STIX

VDO
Today’s talk

1. Pain Points in vulnerability coordination at JPCERT/CC
2. Vulnerability Description Ontology (VDO)
3. Application of VDO: JPCERT/CC’s case
Solution: Convert Reports into a Machine Readable Format using VDO

Free Format!

VDO

Machine Readable Format !!
Benefit #1: Time saving in Coordinate Phase

Free Format!

Triage

Prioritize

Coordinate

Disclose

I'm coordinating more than a few dozen vulnerability at a time

But, easy to understand using VDO
Benefit #2: More Efficient Coordination Process

Free Format!

Automation: CVSS basic score

Automation: Creating JVN advisories CVSS basic score
JPCERT’s case: Toward Automating Advisory Generation

• Define Data representation of VDO
• Implement tools
  • VDO to CVSS basic score
  • VDO to JVN advisory
Define Data representation of VDO

VDO is Information model, NOT a data model

A format to encode VDO
VDO data in a JSON format

• Choose JSON format
• Why? A lot of scripts/tools are utilized

```
{
  "Vulnerability": {
    "VulnID": {
      "cve": "CVE-2014-8506"
    },
    "Provenance": {
      "url": "http://www.avidl.dhs.org/advisories/wordpress/plugins/XcIoner-v3.1.1/"
    },
    "Scenario": {
      "VulnType": {
        "CVE-22: Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal')"
      },
      "Product": {
        "ProductName": "XcIoner plugin for WordPress",
        "Version": "3.1.1"
      },
      "AttackVector": {
        "Remote": {
          "RemoteType": [
            "Internet"
          ]
        }
      }
    }
  }
}
```
Writing a VDO instance

• How to Write a VDO instance?
  • Some editors support autocompletion using a JSON Schema
    • Visual Studio Code, Atom
  • Generating an HTML form from a JSON Schema
    • JSON Editor
      • https://github.com/json-editor/json-editor

• Defined & Implemented JSON Schema for VDO
Writing a VDO instance with autocompletion
JSON Schema for VDO

- VDO JSON Schema
  - https://github.com/JPCERTCC/vdo-json-schema
On-going projects (1/2)

• Tools for automatic advisory generation
  • Mapping VDO data to CVSS base score
    • VDO includes CVSS v2/v3 concept
      • NISTIR 8138 in Appendix shows partial mapping logic
    • The “entire” mapping logic needs to be developed
  • Conversion VDO data to descriptive text (JVN advisory)
    • Our idea
      • Use templates of advisory depended on CWE
      • “Fill in the blanks” of templates from VDO data
On-going projects (2/2)

• Refine NISTIR 8138
  • The framework of VDO is not mature
    • Some noun groups should be discussed
  • 1st round of comments sent to NIST and VRDX SIG
    • Our findings from the feasibility study in JPCERT/CC
    • Discussions on comments to follow
Future work: VDO as a common language

Triage
Prioritize
Coordinate
Disclose

Free Format!

VDO

Lightweight triage

automation

Vendor
Consumer
Concluding remarks

• Vulnerability Description Ontology (VDO)
  • Core information model to describe vulnerability information
  • Has huge potential to aid
    • A format to automatically manage vulnerability information
    • A common language (Taxonomy) for understanding and exchanging vulnerability information

• JPCERT/CC
  • Defined VDO in a JSON format and implemented JSON Schema
  • Started a feasibility study of VDO to improve vulnerability management
Thank you!

JPCERT/CC
Vulnerability Coordination Group
E-mail: vultures@jpcert.or.jp