Vulnerability Discovery Models: Which works, which doesn’t?

Viet Hung Nguyen and Fabio Massacci
University of Trento, Italy
{vhnguyen, massacci}@disi.unitn.it

ASIACCS’12, Seoul, Korea, 02 – 04 May, 2012
The Roadmap

Targets of Analysis
- Precondition & applications for the study

Data Collection
- For each target, collect all available data sets

Data Fit
- Fit data to vulnerability discovery model

Analysis
- Perform analysis on result
Basic Concepts

stå

Vulnerability

✓ An instance of human mistake in specification, development, or configuration of software such that its execution can violate the security policy [Krsul98]

Vulnerability Discovery Model (VDM)

✓ A post-release stage where people identify and report security flaws of a released software
✓ Usually represented as mathematic curves

Existing VDMs

* Alhazmi-Malaiya Logistic (AML)
* Anderson Thermodynamic (AT)
* Linear (LN)
* Logarithmic Poisson (LP)
* Rescolar’s Exponential (RE)
* Rescolar’s Quadratic/Linear (RQ)
The Fallacy of Measurement

🌟 How to measure vulnerabilities?

- Different definitions/sources of vulnerabilities
- Eg. Firefox:
  - Mozilla Bugzilla (only security-relevant bugs)
  - Mozilla Foundation Security Advisory (MFSA)
  - National Vulnerability Database (NVD)
- What is the number of vulns?
  - 6 MFSA, 10 NVD, 14 (security) Bugzilla.

Vulnerability space of Firefox

![Vulnerability space of Firefox diagram]
Research Questions

[* RQ1: which VDM works, which doesn’t?*
  ✓ Do the existing VDMs work?

[* RQ2: how do different ways of counting vulns impact to the performance of VDMs?*
  ✓ Do VDMs behave differently with different types of data set?

[* RQ3: in which definition of vuln, VDMs yield more stable results?*
  ✓ Which type of data set is most appropriate for VDM study?

[* RQ4: which VDM is globally superior?*
  ✓ Which VDM yields better results during software’s lifetime?
Types of Vulnerability Data Set

✔ Release X (eg. FF3.0)

✔ NVD(X) : 1 vuln is 1 NVD entry which mentions X

✔ NVD.Advice(X) : 1 vuln is 1 NVD entry which mentions X, and has a reference to an advisory confirmed by X’s vendor

✔ NVD.Bug(X) : 1 vuln is 1 NVD entry which mentions X, and has a reference to a bug confirmed by X’s vendor

✔ NVD.Nbug(X) : 1 vuln is 1 bug confirmed by X’s vendor, and is referred to by 1 NVD entry mentioning X

✔ Advice.Nbug(X) : 1 vuln is 1 bug confirmed by X’s vendor, and is directly or indirectly referred to by an NVD entry mentioning X
Targets of Analysis

* Targets of Analysis: 17 releases of Browsers
  ✓ IE: v4 - v8
  ✓ Firefox: v1.0 - v3.6
  ✓ Chrome: v1.0 - v6.0

* Why should they be browsers?
  ✓ Complex enough (like a small operating system)
  ✓ Quickly evolve
  ✓ Targets of many attacks

* Why should they be IE, Firefox and Chrome?
  ✓ Top three most popular browsers
Data Collection

* Data sources
  ✓ IE: NVD
  ✓ Firefox: MFSA, Bugzilla, NVD
  ✓ Chrome: ChromeIssue, NVD

* Data collection
  ✓ 58 data sets of 17 releases

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Chrome</td>
<td>●</td>
<td>●</td>
<td>—</td>
<td>●</td>
<td>—</td>
<td>6 (v1.0–v6.0)</td>
</tr>
<tr>
<td>Firefox</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>6 (v1.0–v3.6)</td>
</tr>
<tr>
<td>IE</td>
<td>●</td>
<td>—</td>
<td>●</td>
<td>—</td>
<td>—</td>
<td>5 (v4.0–v8.0)</td>
</tr>
</tbody>
</table>

Bullets (●) indicate enabled data sets. Dashes (—), otherwise, mean there is no data sources available to collect the data sets.
Goodness of Fit (GoF) Analysis

* Fit data to VDMs
  - Non-linear regression method, implemented in R (www.r-project.org)

* Chi-square test for Goodness-of-Fit (GoF)
  - $O_i$ - observed values
  - $E_i$ - expected values

  $$\chi^2 = \sum_{i=1}^{n} \frac{(O_i - E_i)^2}{E_i}$$

* The meaning of Chi-square test
  - Measure the difference between observed and expected values
  - Use p-value of the chi-square test to know whether VDM works or not
RQ1: Which VDM works, which doesn’t?

The goodness of fit of a VDM is based on $p$-value in the $\chi^2$ test. $p$-value $< 0.05$: not fit (-), $p$-value $\geq 0.95$: good fit (X), and inconclusive fit (?) otherwise.

<table>
<thead>
<tr>
<th>NVD Data set</th>
<th>Firefox</th>
<th>Chrome</th>
<th>IE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model</td>
<td>1.0</td>
<td>1.5</td>
<td>2.0</td>
</tr>
</tbody>
</table>

$p$-value $\geq 0.95$: FIT (X)  
$p$-value $< 0.05$: NOT FIT (-)  
$p$-value $0.05 \leq p$-value $< 0.95$: INCONCLUSIVE (?)
RQ1: Which VDM works, which doesn’t?

The goodness of fit of a VDM is based on $p$-value in the $\chi^2$ test. $p$-value < 0.05: not fit (-), $p$-value \(\geq 0.95\): good fit (X), and inconclusive fit (?) otherwise.
### RQ2: The Impact of Types of Data Set

<table>
<thead>
<tr>
<th>VDM</th>
<th>v1.0</th>
<th>v2.0</th>
<th>v3.0</th>
<th>v3.5</th>
<th>v3.6</th>
</tr>
</thead>
<tbody>
<tr>
<td>AML</td>
<td>??</td>
<td>??</td>
<td>??</td>
<td>??</td>
<td>X</td>
</tr>
<tr>
<td>AT</td>
<td></td>
<td>X</td>
<td></td>
<td>?</td>
<td></td>
</tr>
<tr>
<td>LN</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LP</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RQ</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Each column has five cells corresponding to Advice.Nbug, NVD, NVD.Advice, NVD.Bug, NVD.NBug.

**Opposite results for the same models**

*Opposite results are obtained from different data sets*:

- **Same model**
- **Same target (i.e., same software release)**
- **But different counting methods (different types of data set)**
RQ2: The Impact of Data Sets

Advice.Nbug, NVD, NVD.Advice, NVD.Bug, NVD.NBug

Each column has five cells corresponding to Advice.Nbug, NVD, NVD.Advice, NVD.Bug, NVD.NBug

Opposite results for the same models

Different types of data set would strongly impact to VDM’s GoF
Temporal Analysis on Goodness-of-Fit

Temporal Analysis on GoF

Release

6 months since release

8 months

7 months

9 months

Last day data is collected

GoF Analysis

<table>
<thead>
<tr>
<th>App.</th>
<th>Data Set</th>
<th>VDM</th>
<th>Time</th>
<th>GoF</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>nvd</td>
<td>AML</td>
<td>NF</td>
<td></td>
</tr>
<tr>
<td>X</td>
<td>nvd</td>
<td>AML</td>
<td>NF</td>
<td></td>
</tr>
<tr>
<td>X</td>
<td>nvd</td>
<td>AML</td>
<td>I</td>
<td></td>
</tr>
<tr>
<td>X</td>
<td>nvd</td>
<td>AML</td>
<td>F</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>X</td>
<td>nvd</td>
<td>AML</td>
<td>NF</td>
<td></td>
</tr>
</tbody>
</table>

14, 817 data points in total
Temporal Analysis on Goodness-of-Fit

★ The GoF Entropy of VDM
- The chaotic of VDM’s GoF from time $t-1$ to $t$
- Measured by using the GoF transition diagram
- Higher entropy, lesser stability

\[
E_\beta(t) = \frac{|\text{small jump}|_t + \beta \cdot |\text{big jump}|_t}{|\text{unchanged}|_t + |\text{small jump}|_t + \beta \cdot |\text{big jump}|_t}.
\]

★ The Quality of VDM
- How good a VDM is
- Measured by the #GoF at time $t$

\[
Q_\omega(t) = \frac{|\text{Fit}|_t + \frac{1}{\omega} \cdot |\text{Inconclusive}|_t}{|\text{Fit}|_t + |\text{Inconclusive}|_t + |\text{Not Fit}|_t}.
\]
RQ3: The Stability of VDMs in Data Sets

The trend of GoF Entroy

- VDM stability in NVD.Bug is likely the worst
- VDM stability in NVD.Advice is likely the best
RQ4: The Quality of VDMs

* VDM Quality
  ✓ AML is the winner
  ✓ AT is the loser

\[ Q_w(t) = \frac{|\text{Fit}_t| + \frac{1}{w} \cdot |\text{Inconclusive}_t|}{|\text{Fit}_t| + |\text{Inconclusive}_t| + |\text{NotFit}_t|} \]
Conclusion and Future Work

* Summary
  ✓ 6 VDMs are analyzed in 58 data sets of 17 browser releases

* The findings
  ✓ VDM doesn't work: AT (for browsers)
  ✓ VDM (probably) work well: AML (for browsers)
  ✓ VDMs might work: LN, LP, RE, RQ (for browsers)
  ✓ Different types of data set would strongly impact to VDM’s GoF
  ✓ VDMs likely yield more stable result in Vulnerability-as-an-NVD entry confirmed by vendors’ advisories data set (NVD.Advice)

* Future work
  ✓ Replicate experiment in other types of application
    • E.g., Web Servers, Operating Systems,...
Thank you

Viet Hung Nguyen – vhnguyen@disi.unitn.it
Fabio Massacci – massacci@disi.unitn.it