Key Details Phrasing

Jonathan Evans, MITRE
Why have a Description?

• Find the CVE for the vulnerability you are looking for

• Determine that the vulnerability does not exist in the CVE corpus.
What does that have to do with writing a description?

• Including the correct amount and type of information in a description is important.

• If you underreport key details, you may not be able to make the appropriate match later on.

• If you overreport details, you can obscure the distinguishing details and are more prone to introduce errors.
Generic Templates

• [VULNTYPE] in [COMPONENT] in [VENDOR] [PRODUCT] [VERSION] allows [ATTACKER] to [IMPACT] via [VECTOR].

• [COMPONENT] in [VENDOR] [PRODUCT] [VERSION] [ROOT CAUSE], which allows [ATTACKER] to [IMPACT] via [VECTOR].
Default Detail Phrasing

• The following is what to do if you do not have information about a key detail.

• Vulnerability Type: Skip if applicable
  — At this level you should never encounter a vulnerability where you need to skip the type phrasing.

• Component: Skip
• Vendor: Skip
• Product: You MUST have a product name.
• Version: Skip
• Attacker: Use “attackers”
• Impact: Use “unspecified impact”
• Vectors: Use “via unspecified vectors”
Product

• [VULNTYPE] in [COMPONENT] in [VENDOR] [PRODUCT] [VERSION] allows [ATTACKER] to [IMPACT] via [VECTOR].

• [COMPONENT] in [VENDOR] [PRODUCT] [VERSION] [ROOT CAUSE], which allows [ATTACKER] to [IMPACT] via [VECTOR].
<table>
<thead>
<tr>
<th>Name Type</th>
<th>Phrasing</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Product Name</strong></td>
<td>[PRODUCT_NAME]</td>
<td>Notepad</td>
</tr>
<tr>
<td><strong>Vendor Name</strong></td>
<td>[VENDOR_NAME] [PRODUCT_NAME]</td>
<td>Microsoft Notepad</td>
</tr>
<tr>
<td><strong>Typo</strong></td>
<td>Put in keywords</td>
<td>N/A</td>
</tr>
<tr>
<td><strong>Capitalization</strong></td>
<td>Use the same as vendor</td>
<td>NotePad</td>
</tr>
<tr>
<td><strong>Alternate Name</strong></td>
<td>[PRODUCT_NAME] (aka [ALT_NAME])</td>
<td>Notepad (aka WordPad)</td>
</tr>
<tr>
<td><strong>Acronyms</strong></td>
<td>[PRODUCT_NAME] ([ACRONYM])</td>
<td>Notepad (NP)</td>
</tr>
<tr>
<td><strong>Change in Name</strong></td>
<td>[PRODUCT_NAME] (formerly [OLD_NAME])</td>
<td>Notepad (formerly WordPad)</td>
</tr>
<tr>
<td><strong>Shared Code-base</strong></td>
<td>[PRODUCT_NAME] and [OTHER_PRODUCT_NAME]</td>
<td>Notepad and WordPad</td>
</tr>
<tr>
<td><strong>Bundled</strong></td>
<td>[PRODUCT_NAME], as used in [BUNDLING_PRODUCT]</td>
<td>Notepad, as used in WordPad</td>
</tr>
<tr>
<td><strong>Platforms</strong></td>
<td>[PRODUCT_NAME] [COMPONENT_TYPE] for [PLATFORM]</td>
<td>Notepad component for WordPad</td>
</tr>
</tbody>
</table>
Version

• [VULNTYPE] in [COMPONENT] in [VENDOR] [PRODUCT] [VERSION] allows [ATTACKER] to [IMPACT] via [VECTOR].

• [COMPONENT] in [VENDOR] [PRODUCT] [VERSION] [ROOT CAUSE], which allows [ATTACKER] to [IMPACT] via [VECTOR].
# Versions

<table>
<thead>
<tr>
<th>Information available</th>
<th>Example Disclosure Phrasing</th>
<th>CVE Phrasing</th>
<th>CVE Example</th>
<th>CVE Example with multiple versions</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>List vulnerable version(s)</strong></td>
<td>Tested: 1.2.3</td>
<td>The version</td>
<td>1.2.3</td>
<td>1.2.3, 2.3.1, and 3.1.2</td>
</tr>
<tr>
<td><strong>Vulnerable version(s) with indications earlier versions are affect</strong></td>
<td>Tested 1.2.3. Earlier versions are affected.</td>
<td>use &quot;and earlier&quot; after the version</td>
<td>1.2.3 and earlier</td>
<td>1.2.3, 2.3.1, 3.1.2, and earlier</td>
</tr>
<tr>
<td><strong>Fixed/updated version(s)</strong></td>
<td>Fixed in 1.2.3</td>
<td>use &quot;before&quot; before the version</td>
<td>before 1.2.3</td>
<td>before 1.2.3, 2.x before 2.3.1, and 3.x before 3.1.2</td>
</tr>
<tr>
<td><strong>Vulnerable range (e.g. listed min and max version or list of consecutive versions)</strong></td>
<td>1.2.3 to 2.3.1 or Tested: 2.3.1. Introduced in 1.2.3</td>
<td>use &quot;through&quot; between min and max</td>
<td>1.2.1 through 1.2.3</td>
<td>1.2.1 through 1.2.3 and 2.0.1 through 2.3.1</td>
</tr>
<tr>
<td><strong>Vulnerable version with indications later versions are affect</strong></td>
<td>1.2.3 and later</td>
<td>no official phrasing</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td><strong>Mixed version information</strong></td>
<td>-</td>
<td>use the version phrasing where appropriate</td>
<td>N/A</td>
<td>1.2.3, 2.0.3 before 2.3.1, and 3.0.1 through 3.1.2</td>
</tr>
<tr>
<td><strong>Multiple Products</strong></td>
<td>Product A 1.2.3 and Product B 2.3.4</td>
<td>versions follows product names</td>
<td>Product A 1.2.3 and Product B 4.5.6</td>
<td>Product A 1.2.3, 2.3.1, and 3.2.1 and Product B 4.5.6, 5.6.4, and 6.5.4</td>
</tr>
<tr>
<td><strong>Starting &quot;v&quot;</strong></td>
<td>v1.2.3</td>
<td>do not include the v</td>
<td>1.2.3</td>
<td>N/A</td>
</tr>
</tbody>
</table>
Attacker

• [VULNTYPE] in [COMPONENT] in [VENDOR] [PRODUCT] [VERSION] allows [ATTACKER] to [IMPACT] via [VECTOR].

• [COMPONENT] in [VENDOR] [PRODUCT] [VERSION] [ROOT CAUSE], which allows [ATTACKER] to [IMPACT] via [VECTOR].
Attacker Types

• remote attackers
• remote authenticated users
• local users
• physically proximate attackers
• remote [TYPE] servers
• guest OS users
• guest OS administrators
• context-dependent attackers
• attackers
• [EXTENT] user-assisted [ATTACKER]
• man-in-the-middle attackers
Flaw Type/Root Cause

• **[VULNTYPE]** in [COMPONENT] in [VENDOR] [PRODUCT] [VERSION] allows [ATTACKER] to [IMPACT] via [VECTOR].

• [COMPONENT] in [VENDOR] [PRODUCT] [VERSION] [ROOT CAUSE], which allows [ATTACKER] to [IMPACT] via [VECTOR].
Cross-site scripting (XSS) vulnerability in [COMPONENT] in [VENDOR] [PRODUCT] [VERSION] allows remote attackers to inject arbitrary web script or HTML via the [PARAM] parameter.

Multiple cross-site scripting (XSS) vulnerabilities in [VENDOR] [PRODUCT] [VERSION] allow remote attackers to inject arbitrary web script or HTML via the [PARAM] parameter to (1) [COMPONENT1], (2) [COMPONENT2], ..., or (n) [COMPONENTn].

Multiple cross-site scripting (XSS) vulnerabilities in [COMPONENT] in [VENDOR] [PRODUCT] [VERSION] allow remote attackers to inject arbitrary web script or HTML via the (1) [PARAM1], (2) [PARAM2], ..., or (n) [PARAMn] parameter.

Multiple cross-site scripting (XSS) vulnerabilities in [VENDOR] [PRODUCT] [VERSION] allow remote attackers to inject arbitrary web script or HTML via the (1) [PARAM1] or (2) [PARAM2] parameter to [COMPONENT1]; the (3) [PARAM3] parameter to [COMPONENT2]; ...; or (n) [PARAMn] parameter to [COMPONENTm].
## SQL Injection

<table>
<thead>
<tr>
<th>#Params</th>
<th>#Comp</th>
<th>Template</th>
</tr>
</thead>
<tbody>
<tr>
<td>One</td>
<td>One</td>
<td>SQL injection vulnerability in [COMPONENT] in [VENDOR] [PRODUCT] [VERSION] allows [ATTACKER] to execute arbitrary SQL commands via the [PARAM] parameter.</td>
</tr>
<tr>
<td>One</td>
<td>Multiple</td>
<td>Multiple SQL injection vulnerabilities in [VENDOR] [PRODUCT] [VERSION] allow [ATTACKER] to execute arbitrary SQL commands via the [PARAM] parameter to (1) [COMPONENT1], (2) [COMPONENT2], ..., or (n) [COMPONENTn].</td>
</tr>
<tr>
<td>Multiple</td>
<td>One</td>
<td>Multiple SQL injection vulnerabilities in [COMPONENT] in [VENDOR] [PRODUCT] [VERSION] allow [ATTACKER] to execute arbitrary SQL commands via the (1) [PARAM1], (2) [PARAM2], ..., or (n) [PARAMn] parameter.</td>
</tr>
<tr>
<td>Multiple</td>
<td>Multiple</td>
<td>Multiple SQL injection vulnerabilities in [VENDOR] [PRODUCT] [VERSION] allow [ATTACKER] to execute arbitrary SQL commands via the (1) [PARAM1] or (2) [PARAM2] parameter to [COMPONENT1]; the (3) [PARAM3] parameter to [COMPONENT2]; ...; or (n) [PARAMn] parameter to [COMPONENTm].</td>
</tr>
</tbody>
</table>
Component/Vector

• [VULNTYPE] in [COMPONENT] in [VENDOR] [PRODUCT] [VERSION] allows [ATTACKER] to [IMPACT] via [VECTOR].

• [COMPONENT] in [VENDOR] [PRODUCT] [VERSION] [ROOT CAUSE], which allows [ATTACKER] to [IMPACT] via [VECTOR].
Vectors/Components Definitions

• Vector – The inputs and/or processes required to exploit the vulnerability

• Component – Part of the product
  – Trigger point – The part of the product where the error occurs (may be multiple places)
  – Interaction point – The part of the product that accepts the vectors
  – Neither are a requirement. You can skip them if there is no information for them.

• Payload -
Components

- Generic Template has 2 component locations
  - After the vulnerability type, but before the product name
  - After the vector
- Trigger point goes before the product name
- Interaction point goes after the vector
- Default to before the product if
  - You are unsure which type of component it is
  - You think the component can be both a trigger and interaction point.
- For multiple component/vector pairs
  - Components always go after the vector, no matter their type
  - Dot notation is used
Vectors cont.

- Vectors have the greatest variation in phrasing.
- Sometimes there can be multiple vectors for a single vulnerability.
- Vector phrasing tends to vary by flaw type, with vector phrasing being more consistent within flaw types.
Dot Notations

• When we merge multiple vulnerabilities, we want to give them a number so that we can reference the individual vulnerabilities.

• Ex: CVE-0000-0000
  – Multiple cross-site scripting (XSS) vulnerabilities in Product 1.0 allow remote attackers to inject arbitrary web script or HTML via the id parameter to (1) comp1.html or (2) comp2.html
  – CVE-0000-0000.1 vs CVE-0000-0000.2
Dot Notation Cont.

• Not all lists use dot notation.
• We don’t use dot notation for:
  – Products
  – Versions
  – Attackers
• Impacts only receive dot notation when it is believed that they indicate multiple vulnerabilities, e.g. CSRF vulns.