Quantitative Security

Colorado State University
Yashwant K Malaiya
CS559
L10

CSU Cybersecurity Center
Computer Science Dept
L10 Outline

• Vulnerabilities
  – Finders
  – Classification
  – CVE numbering system
  – Rewards and sale
  – Data bases
  – Life Cycle
Quantitative Security
Colorado State University
Yashwant K Malaiya
CS559
Vulnerabilities

CSU Cybersecurity Center
Computer Science Dept
Security Holes: Types

• Software holes: Vulnerabilities
  – CVSS scores involving exploitability and impact is a type of risk measure.
• System/physical holes
• Personnel/Procedural holes:
  – e.g. Phishing
• Exploitation may involve multiple holes, perhaps of different types
• Classify them:
  – Target 2013 breach: credentials stolen from a HVAC contractor
  – Equifax 2017 breach: vulnerability patch not applied
Defects vs vulnerabilities

• Software defects
• Software Vulnerabilities
• Testing:
  – prior to release
  – Usage
  – Field testing
Components of Likelihood of Exploitation

• Internal
  – Presence of a vulnerability (*Vulnerability Discovery*)
  – Vulnerability not patched

• External
  – Attacker’s motivation level
  – Technical capabilities, *exploit availability*
  – Network access to vulnerable system

• Interface
  – *Attack surface* of vulnerable system
  – *Reachability* of vulnerability

We have published research related to these. See publications 2005-2016.
Vulnerabilities Trend

https://www.cvedetails.com/browse-by-date.php

Vulnerability Data-bases
- NVD database
- CVEDetails
- VulnDB
- ExploitDB
Vulnerability Lifecycle

Vulnerabilities: "defect which enables an attacker to bypass security measures"

Exploit code ("exploit"): usually available after disclosure
Vulnerability density and defect density

- **Vulnerability densities**: 95/98: 0.003-0.004  NT/2000/XP: 0.01-0.02
- \( V_{KD}/D_{KD} \): 0.68-1.62%  about 1%

<table>
<thead>
<tr>
<th>System</th>
<th>MSLOC</th>
<th>Known Defects (1000s)</th>
<th>( D_{KD} )/Kloc</th>
<th>Known Vulnerabilities</th>
<th>( V_{KD} )/Kloc</th>
<th>Ratio ( V_{KD} /D_{KD} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Win 95</td>
<td>15</td>
<td>5</td>
<td>0.33</td>
<td>46</td>
<td>0.0031</td>
<td>0.92%</td>
</tr>
<tr>
<td>NT 4.0</td>
<td>16</td>
<td>10</td>
<td>0.625</td>
<td>162</td>
<td>0.0101</td>
<td>1.62%</td>
</tr>
<tr>
<td>Win 98</td>
<td>18</td>
<td>10</td>
<td>0.556</td>
<td>84</td>
<td>0.0047</td>
<td>0.84%</td>
</tr>
<tr>
<td>Win2000</td>
<td>35</td>
<td>63</td>
<td>1.8</td>
<td>508</td>
<td>0.0145</td>
<td>0.81%</td>
</tr>
<tr>
<td>Win XP</td>
<td>40</td>
<td>106.5*</td>
<td>2.66*</td>
<td>728</td>
<td>0.0182</td>
<td>0.68%*</td>
</tr>
</tbody>
</table>

### Who discovers vulnerabilities?

<table>
<thead>
<tr>
<th>DISCOVERERS</th>
<th>SAFARI’S VULNERABILITIES</th>
<th>PERCENTAGE</th>
<th>CHROMIUM’S VULNERABILITIES</th>
<th>PERCENTAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>PRODUCT’S COMPANY DISCOVERERS</strong></td>
<td>17</td>
<td>20%</td>
<td>0</td>
<td>0%</td>
</tr>
<tr>
<td><strong>PRODUCT’S COMPANY DISCOVERERS AND OTHERS</strong></td>
<td>0</td>
<td>0%</td>
<td>35</td>
<td>35%</td>
</tr>
<tr>
<td><strong>OUTSIDE DISCOVERERS</strong></td>
<td>66</td>
<td>80%</td>
<td>63</td>
<td>64%</td>
</tr>
<tr>
<td><strong>UNKNOWN DISCOVERERS</strong></td>
<td>0</td>
<td>0%</td>
<td>1</td>
<td>1%</td>
</tr>
</tbody>
</table>

Vulnerability discoverers from July 1, 2012 to December 31, 2012: insiders or outsiders

---

## THE TOP VULNERABILITIES DISCOVERERS ON OSVDB

<table>
<thead>
<tr>
<th>Discoverer</th>
<th>Country</th>
<th>Period</th>
<th># Vuln</th>
<th># Vuln types</th>
<th>Why they’re interested</th>
<th>Stopped/Continued</th>
</tr>
</thead>
<tbody>
<tr>
<td>r0t</td>
<td>Latvia</td>
<td>2005-08-09 to 2010-09-16</td>
<td>810</td>
<td>10</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Janek Vind &quot;waraxe&quot;</td>
<td>Estonia</td>
<td>2003-08-08 to 2013-03-21</td>
<td>319</td>
<td>8</td>
<td>Vulnerability website</td>
<td>N/A</td>
</tr>
<tr>
<td>Lostmon Lords</td>
<td>Spain</td>
<td>2004-06-20 to 2009-08-15</td>
<td>279</td>
<td>8</td>
<td>Security Researcher</td>
<td>Worked until July 2012</td>
</tr>
<tr>
<td>rgod</td>
<td>Italy</td>
<td>2005-06-06 to 2012-08-29</td>
<td>277</td>
<td>12</td>
<td>Hacker</td>
<td>Worked until Aug. 2012</td>
</tr>
<tr>
<td>Luigi Auriemma</td>
<td>Italy</td>
<td>2000-07-08 to 2013-03-16</td>
<td>267</td>
<td>9</td>
<td>Hobby</td>
<td>N/A</td>
</tr>
<tr>
<td>Russ McRee</td>
<td>USA</td>
<td>2008-01-14 to 2012-03-02</td>
<td>237</td>
<td>4</td>
<td>Specialist in security</td>
<td>N/A</td>
</tr>
<tr>
<td>Aliaksandr Hartsuyeu</td>
<td>Lithuania</td>
<td>2005-12-28 to 2011-02-03</td>
<td>229</td>
<td>6</td>
<td>Security Company</td>
<td>Still working 2012</td>
</tr>
<tr>
<td>James Bercegay</td>
<td>USA</td>
<td>2003-06-03 to 2008-09-04</td>
<td>200</td>
<td>12</td>
<td>Web developer</td>
<td>Worked until 2011</td>
</tr>
<tr>
<td>Kacper</td>
<td>Poland</td>
<td>2006-05-12 to 2007-08-10</td>
<td>199</td>
<td>3</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>luny</td>
<td>N/A</td>
<td>2006-05-18 to 2006-07-13</td>
<td>142</td>
<td>6</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Diabolic Crab</td>
<td>N/A</td>
<td>2004-09-25 to 2005-07-12</td>
<td>140</td>
<td>6</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>JeiAr</td>
<td>USA</td>
<td>2003-05-29 to 2004-05-04</td>
<td>120</td>
<td>7</td>
<td>Web developer</td>
<td>Worked until 2011</td>
</tr>
<tr>
<td>Tan Chew Keong</td>
<td>Singapore</td>
<td>2004-07-29 to 2009-09-28</td>
<td>102</td>
<td>9</td>
<td>Information Security Specialist</td>
<td>N/A</td>
</tr>
<tr>
<td>Stefan Esser</td>
<td>Germany</td>
<td>2000-11-09 to 2012-06-03</td>
<td>86</td>
<td>10</td>
<td>Security Consultant</td>
<td>Still do jailbreak until 2012</td>
</tr>
<tr>
<td>M.Hasran Addahroni</td>
<td>Indonesia</td>
<td>2006-02-09 to 2009-02-07</td>
<td>80</td>
<td>2</td>
<td>Security Gossiper&amp;Bugs Hunter</td>
<td>N/A</td>
</tr>
</tbody>
</table>

What happened to discoverers?

Most of the top discoverers here are credited with discovering the vulnerabilities during the first three years. However, a few discoverers have continued to discover vulnerabilities for several years.

Why do some very successful discoverers disappear from the scene after two to three years?
• A possible explanation is that, during those two to three years, they acquire the notoriety of being accomplished vulnerability discoverers.
• After that, they start offering their services to software developers or security service companies on a contract basis or as employees.
• Some of them may be able to start their own small organizations.
Vulnerability finders in 2019

Participants in bug bounty programs
Classification of Vulnerabilities

**OWASP Top Ten** Open Web Application Security Project a nonprofit that works to improve the security of software

- **A1 Injection**: Sending hostile data to an interpreter (e.g. SQL, LDAP, command line)
- **A2 Broken Authentication**:
  - Weak session management
  - Credential stuffing
  - Brute force
  - Forgotten password
  - No multi-factor authentication
  - Sessions don’t expire
- **A3 Sensitive Data Exposure**
  - Clear-text data transfer
  - Unencrypted storage
  - Weak crypto or keys
  - Certificates not validated
  - Exposing PII or Credit Cards
- **A4 XML External Entities (XXE)**
  - The application accepts XML, and assumes it is safe
- **A5 Broken Access Control**
  - Access hidden pages
  - Elevate to an administrative account
  - View other people’s data
  - Modifying cookies or JWT tokens

[Introduction to the OWASP Top Ten Video]
Classification of Vulnerabilities

OWASP Top Ten 2017

• A6 Security Misconfiguration
  – Security features not configured properly
  – Unnecessary features enabled
  – Default accounts not removed
  – Error messages expose sensitive information

• A7 Cross-Site Scripting (XSS)
  – HTML mixes content, presentation and code into one string (HTML+CSS+JS)
  – If an attacker can alter the DOM, they can do anything that the user can do.
  – XSS can be found using automated tools.

• A8 Insecure Deserialization
  – Programming languages allow you to turn a tree of objects into a string that can be sent to the browser.
  – If you deserialise untrusted data, you may allow objects to be created, or code to be executed.

• A9 Using Components with Known Vulnerabilities
  – Modern applications contain a lot of third-party code.
  – It’s hard to keep it all up to date.
  – Attackers can enumerate the libraries you use and develop exploits.

• A10 Insufficient Logging & Monitoring
  – You can’t react to attacks that you don’t know about.
  – Logs are important for: Detecting incidents, Understanding what happened, Proving who did something
https://www.cvedetails.com/vulnerabilities-by-types.php
(1999-2019)
Note: A vulnerability can have multiple types.
<table>
<thead>
<tr>
<th>Year</th>
<th># of Vulnerabilities</th>
<th>DoS</th>
<th>Code Execution</th>
<th>Overflow</th>
<th>Memory Corruption</th>
<th>Sql Injection</th>
<th>XSS</th>
<th>Directory Traversal</th>
<th>Http Response Splitting</th>
<th>Bypass something</th>
<th>Gain Information</th>
<th>Gain Privileges</th>
<th>CSRF</th>
<th>File Inclusion</th>
<th># of exploits</th>
</tr>
</thead>
<tbody>
<tr>
<td>1999</td>
<td>894</td>
<td>177</td>
<td>112</td>
<td>172</td>
<td>2</td>
<td>7</td>
<td>25</td>
<td>16</td>
<td>103</td>
<td>48</td>
<td>19</td>
<td>139</td>
<td></td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>2000</td>
<td>1020</td>
<td>257</td>
<td>208</td>
<td>206</td>
<td>2</td>
<td>4</td>
<td>20</td>
<td>46</td>
<td>19</td>
<td>12</td>
<td>139</td>
<td>144</td>
<td></td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>2001</td>
<td>1677</td>
<td>403</td>
<td>403</td>
<td>297</td>
<td>7</td>
<td>34</td>
<td>123</td>
<td>83</td>
<td>74</td>
<td>199</td>
<td>2</td>
<td>144</td>
<td></td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>2002</td>
<td>2156</td>
<td>498</td>
<td>553</td>
<td>435</td>
<td>2</td>
<td>41</td>
<td>206</td>
<td>103</td>
<td>127</td>
<td>74</td>
<td>199</td>
<td>144</td>
<td></td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>2003</td>
<td>1527</td>
<td>381</td>
<td>477</td>
<td>371</td>
<td>2</td>
<td>49</td>
<td>129</td>
<td>60</td>
<td>1</td>
<td>62</td>
<td>69</td>
<td>144</td>
<td></td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>2004</td>
<td>2451</td>
<td>580</td>
<td>614</td>
<td>410</td>
<td>3</td>
<td>148</td>
<td>291</td>
<td>111</td>
<td>12</td>
<td>145</td>
<td>96</td>
<td>134</td>
<td></td>
<td></td>
<td>5</td>
</tr>
<tr>
<td>2005</td>
<td>4935</td>
<td>838</td>
<td>1627</td>
<td>657</td>
<td>21</td>
<td>604</td>
<td>786</td>
<td>202</td>
<td>15</td>
<td>280</td>
<td>261</td>
<td>221</td>
<td></td>
<td>11</td>
<td>16</td>
</tr>
<tr>
<td>2006</td>
<td>6610</td>
<td>893</td>
<td>2719</td>
<td>663</td>
<td>91</td>
<td>967</td>
<td>1302</td>
<td>322</td>
<td>8</td>
<td>267</td>
<td>271</td>
<td>184</td>
<td></td>
<td>18</td>
<td>849</td>
</tr>
<tr>
<td>2007</td>
<td>6520</td>
<td>1101</td>
<td>2601</td>
<td>954</td>
<td>95</td>
<td>706</td>
<td>884</td>
<td>339</td>
<td>14</td>
<td>267</td>
<td>324</td>
<td>242</td>
<td></td>
<td>69</td>
<td>760</td>
</tr>
<tr>
<td>2008</td>
<td>5632</td>
<td>894</td>
<td>2310</td>
<td>699</td>
<td>128</td>
<td>1101</td>
<td>807</td>
<td>363</td>
<td>7</td>
<td>288</td>
<td>270</td>
<td>189</td>
<td></td>
<td>83</td>
<td>170</td>
</tr>
<tr>
<td>2009</td>
<td>5736</td>
<td>1035</td>
<td>2185</td>
<td>700</td>
<td>188</td>
<td>963</td>
<td>851</td>
<td>322</td>
<td>9</td>
<td>337</td>
<td>302</td>
<td>223</td>
<td></td>
<td>115</td>
<td>138</td>
</tr>
<tr>
<td>2010</td>
<td>4652</td>
<td>1102</td>
<td>1714</td>
<td>680</td>
<td>342</td>
<td>520</td>
<td>605</td>
<td>275</td>
<td>8</td>
<td>234</td>
<td>282</td>
<td>238</td>
<td></td>
<td>86</td>
<td>73</td>
</tr>
<tr>
<td>2011</td>
<td>4155</td>
<td>1321</td>
<td>1324</td>
<td>720</td>
<td>351</td>
<td>294</td>
<td>462</td>
<td>108</td>
<td>7</td>
<td>197</td>
<td>409</td>
<td>206</td>
<td>58</td>
<td>17</td>
<td>557</td>
</tr>
<tr>
<td>2012</td>
<td>5297</td>
<td>1425</td>
<td>1459</td>
<td>843</td>
<td>423</td>
<td>243</td>
<td>758</td>
<td>122</td>
<td>13</td>
<td>344</td>
<td>389</td>
<td>250</td>
<td></td>
<td>166</td>
<td>624</td>
</tr>
<tr>
<td>2013</td>
<td>5191</td>
<td>1455</td>
<td>1186</td>
<td>859</td>
<td>386</td>
<td>156</td>
<td>650</td>
<td>110</td>
<td>7</td>
<td>352</td>
<td>511</td>
<td>274</td>
<td></td>
<td>123</td>
<td>205</td>
</tr>
<tr>
<td>2014</td>
<td>7946</td>
<td>1598</td>
<td>1574</td>
<td>848</td>
<td>420</td>
<td>305</td>
<td>1105</td>
<td>204</td>
<td>12</td>
<td>457</td>
<td>2106</td>
<td>239</td>
<td>264</td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>2015</td>
<td>6484</td>
<td>1791</td>
<td>1826</td>
<td>1083</td>
<td>749</td>
<td>218</td>
<td>778</td>
<td>150</td>
<td>12</td>
<td>577</td>
<td>474</td>
<td>248</td>
<td>5</td>
<td>127</td>
<td>5</td>
</tr>
<tr>
<td>2016</td>
<td>6447</td>
<td>2028</td>
<td>1494</td>
<td>1324</td>
<td>717</td>
<td>94</td>
<td>497</td>
<td>99</td>
<td>15</td>
<td>444</td>
<td>843</td>
<td>609</td>
<td>87</td>
<td>7</td>
<td>1</td>
</tr>
<tr>
<td>2017</td>
<td>14714</td>
<td>3154</td>
<td>2004</td>
<td>2495</td>
<td>745</td>
<td>508</td>
<td>1518</td>
<td>279</td>
<td>11</td>
<td>620</td>
<td>1639</td>
<td>459</td>
<td>527</td>
<td>18</td>
<td>6</td>
</tr>
<tr>
<td>2018</td>
<td>16556</td>
<td>1853</td>
<td>2041</td>
<td>2368</td>
<td>400</td>
<td>517</td>
<td>2042</td>
<td>531</td>
<td>11</td>
<td>708</td>
<td>1424</td>
<td>247</td>
<td>461</td>
<td>31</td>
<td>4</td>
</tr>
<tr>
<td>2019</td>
<td>12174</td>
<td>919</td>
<td>2277</td>
<td>1247</td>
<td>236</td>
<td>410</td>
<td>1593</td>
<td>280</td>
<td>4</td>
<td>495</td>
<td>900</td>
<td>129</td>
<td>398</td>
<td>40</td>
<td>6</td>
</tr>
<tr>
<td>Total</td>
<td>122774</td>
<td>23603</td>
<td>22718</td>
<td>18081</td>
<td>5339</td>
<td>7853</td>
<td>15203</td>
<td>4130</td>
<td>166</td>
<td>6375</td>
<td>10989</td>
<td>5006</td>
<td>2521</td>
<td>2235</td>
<td>4333</td>
</tr>
<tr>
<td>% Of All</td>
<td>19.2</td>
<td>26.6</td>
<td>14.7</td>
<td>4.3</td>
<td>6.4</td>
<td>12.5</td>
<td>3.4</td>
<td>0.1</td>
<td>5.2</td>
<td>9.0</td>
<td>4.1</td>
<td>2.1</td>
<td>1.8</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Top ten software flaws used by crooks

According to the Recorded Future Annual Vulnerability report in 2019:

- CVE-2018-15982 – Adobe Flash Player
- CVE-2018-8174 – Microsoft Internet Explorer
- CVE-2017-11882 – Microsoft Office
- CVE-2018-4878 – Adobe Flash Player
- CVE-2019-0752 – Microsoft Internet Explorer
- CVE-2017-0199 – Microsoft Office
- CVE-2015-2419 – Microsoft Internet Explorer
- CVE-2018-20250 – Microsoft WinRAR
- CVE-2017-8750 – Microsoft Internet Explorer
- CVE-2012-0158 – Microsoft Office

The year is the year of disclosure. Most vulnerabilities are disclosed and patched at the same time.

https://www.zdnet.com/article/these-are-the-top-ten-software-flaws-used-by-crooks-make-sure-youve-applied-the-patches
CVE numbering system

CVE, (Common Vulnerabilities and Exposures), is a list of publicly disclosed computer security flaws.

- A CVE, usually means the CVE ID number assigned to a security flaw (“vulnerability”).
- Security advisories issued almost always mention at least one CVE ID.
- CVE is overseen by the MITRE corporation
  - funded from the Cybersecurity and Infrastructure Security Agency, part of the U.S. Department of Homeland Security
- CVE identifiers are assigned by a CVE Numbering Authority (CNA).
  - There are about 100 CNAs, including major IT vendors as well as security companies and research organizations. MITRE can also issue CVEs directly.
  - CVE reports can come from a vendor, a researcher, or anyone who has discovered a flaw
  - There are mechanisms for responsible disclosure.
  - a CVE ID is assigned before a security advisory is made public. Vendors generally keep security flaws secret until a fix has been developed and tested.
  - Once made public, a CVE entry includes
    - the CVE ID (in the format "CVE-2019-1234567"),
    - a brief description of the security vulnerability or exposure, and
    - references, which can include links to vulnerability reports and advisories.
CNAs have the discretion to determine whether something is a vulnerability. [https://cve.mitre.org/cve/cna/rules.html#section_7-1 what_is_a_vulnerability]

- Root CNAs may provide additional guidance to their child CNAs. This allows the program to adapt to definitions used in different industries, legal regimes, and cultures.
  - If a product owner considers an issue to be a vulnerability in its product, then the issue MUST be considered a vulnerability, regardless of whether other parties (e.g., other vendors whose products share the affected code) agree.
  - If the CNA determines that an issue violates the security policy of a product, then the issue SHOULD be considered a vulnerability.
  - If a CNA receives a report about a new vulnerability that has a negative impact, then the reported vulnerability MAY be considered a vulnerability.

- If a weakness cannot be exploited by an attacker, it is a weakness, not a vulnerability. CWE stands for Common Weakness Enumeration.
Responsible Disclosure

A vulnerability discoverer may

• Submit a report to the company with a Vulnerability Disclosure Program
  – If they have a disclosure program, they will acknowledge the report.
  – Evaluate the report to see if the report is valid.
  – Provide a reward if they have a Reward Program
  – If they do not respond within a fixed time (90-120 days), the discoverer may disclose the vulnerability.

• Sell it to a vulnerability broker (TrendMicro ZDI, iDefence VCP etc.)
• Sell it to a government organization
• Sell it in the black market
## Vulnerability Reward programs (2014)

<table>
<thead>
<tr>
<th>Program</th>
<th># Vulns. type</th>
<th>Max reward</th>
<th>Min reward</th>
<th># of beneficiaries</th>
<th>Trend</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Vulnerability Reward Program for Google web properties</strong></td>
<td>5</td>
<td>$20,000</td>
<td>$100</td>
<td>2010: 51</td>
<td>Increase</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2011: 122</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2012: 189</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2013: 226</td>
<td></td>
</tr>
<tr>
<td><strong>Chrome Vulnerability Reward Program</strong></td>
<td>Any security bug</td>
<td>&gt;= 10,000</td>
<td>$500</td>
<td>543</td>
<td>N/A</td>
</tr>
<tr>
<td><strong>The Mozilla Security Bug Bounty Program</strong></td>
<td>Certain bugs depending on some criteria</td>
<td>$3000 (US) cash reward and a Mozilla T-shirt</td>
<td>$500</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td><strong>Facebook</strong></td>
<td>Certain qualifying security bugs</td>
<td>No maximum</td>
<td>$500</td>
<td>Prior to 2011: 43</td>
<td>Increase</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2011: 46</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2012: 111</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2013: 235</td>
<td></td>
</tr>
<tr>
<td><strong>WordPress Security Bug Bounty Program</strong></td>
<td>11</td>
<td>$1000</td>
<td>$25</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td><strong>CCBill Vulnerability Reward Program</strong></td>
<td>7</td>
<td>$500</td>
<td>$300</td>
<td>42</td>
<td>Hold</td>
</tr>
<tr>
<td><strong>Secunia Vulnerability Coordination Reward Program (SVCRP)</strong></td>
<td>Most bugs depending on some criteria</td>
<td>Most Valued Contributor &amp; Most Interesting Coordination Report</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td><strong>ZDI Rewards Program (TippingPoint)</strong></td>
<td>Particular bugs depending on some criteria</td>
<td>$25,000</td>
<td>$1000</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td><strong>iDefense (Verisign)</strong></td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>Significant number</td>
<td>N/A</td>
</tr>
</tbody>
</table>
Bugcrowd Priority One Report 2019

Total payouts increased 83% year over year.

- The average payout for a critical vulnerability in 2019 is $2,669.92, a 27% increase year over year.
- In the first half of 2019, we saw a 29% increase in the number of programs launched versus the same time the year before and a 50% increase in public programs launched.
- Submissions have increased 92% overall, with submissions on IoT targets increasing more than any other at 384%.
- BugCrowd (?), HackerOne ($23 million in 2018)
..As if Pereira's story isn't enough, we have to mention another 19-year-old South American who is killing the bug bounty game: Argentina's Santiago Lopez, the first person to top $1 million in earnings on HackerOne's platform.

The self-taught hacker says he got his start by watching YouTube videos and reading blogs on his own, but the thing that jumpstarted his interest in hacking? What else? The 1995 movie Hackers. (Photo by United Artists/Getty Images)

https://www.pcmag.com/news/7-huge-bug-bounty-payouts
# Google Reward Programs

<table>
<thead>
<tr>
<th>Issue Description</th>
<th>High-quality report with functional exploit</th>
<th>High-quality report</th>
<th>Baseline</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sandbox escape / Memory corruption in a non-sandboxed process</td>
<td>$30,000</td>
<td>$20,000</td>
<td>Up to $15,000</td>
</tr>
<tr>
<td>Universal Cross Site Scripting (includes Site Isolation bypass)</td>
<td>$20,000</td>
<td>$15,000</td>
<td>Up to $10,000</td>
</tr>
<tr>
<td>Renderer RCE / memory corruption in a sandboxed process</td>
<td>$10,000</td>
<td>$7,500</td>
<td>Up to $5,000</td>
</tr>
<tr>
<td>Security UI Spoofing</td>
<td>$7,500</td>
<td>N/A[1]</td>
<td>Up to $3,000</td>
</tr>
<tr>
<td>User information disclosure</td>
<td>$5,000 - $20,000</td>
<td>N/A[1]</td>
<td>Up to $2,000</td>
</tr>
<tr>
<td>Web Platform Privilege Escalation</td>
<td>$5,000</td>
<td>$3,000</td>
<td>Up to $1,000</td>
</tr>
<tr>
<td>Exploitation Mitigation Bypass</td>
<td>$5,000</td>
<td>$3,000</td>
<td>Up to $1,000</td>
</tr>
<tr>
<td>Chrome OS</td>
<td>See below</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Chrome Fuzzer Bonus</td>
<td>$1,000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Chrome Patch Bonus</td>
<td>$500 - $2,000</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

[1] For these classes of bugs, high quality reports are expected to demonstrate the UI spoof or show how user information could be disclosed, which we treat as a functional exploit.

**Non-qualifying flaws:** things that are nearly impossible to exploit, legitimate features, which are not Google’s fault, etc.

Vulnerabilities for sale

- Stefan Frei, NSS Labs looked at reports about some of those private vendors
  - Endgame Systems, Exodus Intelligence, Netragard, ReVuln and VUPEN
  - concluded that jointly these firms alone have the capacity to sell more than 100 zero-day exploits per year.

<table>
<thead>
<tr>
<th>Provider</th>
<th>Offering</th>
<th>Remark / Source</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>USD $2.5 million</td>
<td></td>
</tr>
<tr>
<td>Exodus Intelligence</td>
<td>60 exploits/year</td>
<td>Service Offering <a href="https://www.exodusintel.com/rsrc/ExodusIntelligence_EXP.pdf">https://www.exodusintel.com/rsrc/ExodusIntelligence_EXP.pdf</a></td>
</tr>
<tr>
<td>VUPEN</td>
<td>&gt; 7 exploits/year</td>
<td>Minimum estimate by counting list of published exploits here: <a href="http://www.vupen.com/blog/">http://www.vupen.com/blog/</a> (2013-09-27)</td>
</tr>
<tr>
<td></td>
<td>&gt; 15 to 20 binary analysis and private 1-day exploits/month</td>
<td>Service Offering: <a href="http://www.vupen.com/english/services/ba-gov.php">http://www.vupen.com/english/services/ba-gov.php</a></td>
</tr>
</tbody>
</table>

Frei’s minimum estimate of exploits offered by boutique exploit providers each year.

https://krebsonsecurity.com/2013/12/how-many-zero-days-hit-you-today/
Vulnerability Data Bases

• Vulnerabilities are security related defects.
• For ordinary defects found, the developers do not have an obligation to report them.
  – They just have to address them in the next patch/version.
• All recognized vulnerabilities are analyzed and reported in the data-bases.
• The new entries in the data-bases are used by antivirus developers/maintainers to add to their signatures.
• Used by researchers like us.
Major vulnerability databases

• NVD National Vulnerability Database https://nvd.nist.gov/
  – US 2005, includes vulnerability operation, its CVSS rating, and links to any available patches/fixes

• CVE Mitre DB https://cve.mitre.org/data/downloads/index.html

• CWE Common weakness enumeration https://cwe.mitre.org/

• VulnDB https://vulndb.cyberriskanalytics.com

• WhiteSource Vuln DB https://www.whitesourcesoftware.com/vulnerability-database/
  – Includes not yet recognized vulnerabilities

• CVEDetails https://www.cvedetails.com better interface?

• Exploit DB https://www.exploit-db.com available exploits
Example: CVE-2018-8174

A remote code execution vulnerability exists in the way that the VBScript engine handles objects in memory

- Microsoft acknowledgements
- CVE Details cvss2.0, products/versions affected, links
- Exploit-db has an exploit
Quantitative Security

Colorado State University
Yashwant K Malaiya
CS 559
Vulnerability Life Cycle

CSU Cybersecurity Center
Computer Science Dept
Vulnerability Lifecycle

- Discovery
- Disclosure
- Exploit
- Patch available
- Patch installed

Risk Levels:
- Black Risk
- Gray Risk
- White Risk

Window of exposure
Before We Knew It An Empirical Study of Zero-Day Attacks In The Real World

Attack timeline. These events do not always occur in this order, but \( t_a > t_p \geq t_d \)
> \( t_v \) and \( t_0 \geq t_d \).
The relation between \( t_d \) and \( t_e \) cannot be determined in most cases. For a zero-
day attack \( t_0 > t_e \).
Vulnerability Lifecycle

- **Vulnerability introduced.** A bug is introduced in software (time = $t_v$).
- **Exploit released in the wild.** Actors in the underground economy discover the vulnerability, create a working exploit and use it to conduct stealth attacks against selected targets (time = $t_e$).
- **Vulnerability discovered by the vendor.** The vendor learns about the vulnerability, assesses its severity, assigns a priority for fixing it and starts working on a patch (time = $t_a$).
- **Vulnerability disclosed publicly.** The vulnerability is disclosed, either by the vendor or on public forums and mailing lists. A CVE identifier (e.g., CVE-2010-2568) is assigned to the vulnerability (time = $t_0$).
- **Anti-virus signatures released.** Once the vulnerability is disclosed, anti-virus vendors release new signatures (time = $t_s$).
- **Patch released.** On the disclosure date, or shortly afterward the software vendor releases a patch for the vulnerability. After this point, the hosts that have applied the patch are no longer susceptible to the exploit (time = $t_p$).
- **Patch deployment completed.** All vulnerable hosts worldwide are patched and the vulnerability ceases to have an impact (time = $t_a$).
For a single vulnerability, the cumulative risk in a specific system at time $t$ can be expressed as

- probability of the vulnerability being in State 3 at time $t$
- multiplied by
- the consequence of the vulnerability exploitation.

Zero-day attacks

- A **zero-day attack** is a cyber attack exploiting a vulnerability that has not been disclosed publicly.
- There is almost no defense against a zero-day attack:
  - while the vulnerability remains unknown, the software affected cannot be patched and
  - anti-virus products cannot detect the attack through signature-based scanning
- Notable zero-day attacks include (Bilge, Dumitras)
  - the 2010 Hydraq trojan, also known as the “Aurora” attack
  - the 2010 Stuxnet worm, which combined four zero-day vulnerabilities to target
  - industrial control systems and
  - the 2011 attack against RSA.
An Empirical Study of Zero-Day Attacks In The Real World

- Field-gathered data for 11 million real hosts around the world.
- Searching this data set for malicious files that exploit known vulnerabilities indicates which files appeared on the Internet before the vulnerabilities were disclosed.
- They identify 18 vulnerabilities exploited before disclosure, of which 11 were not previously known to have been employed in zero-day attacks.
- They also find that a typical zero-day attack lasts 312 days on average.
- After vulnerabilities are disclosed publicly, the volume of attacks exploiting them increases by up to 5 orders of magnitude.
### Summary of findings

<table>
<thead>
<tr>
<th>Findings</th>
<th>Implications</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zero-day attacks are more frequent than previously thought: 11 out of 18 vulnerabilities identified were not known zero-day vulnerabilities.</td>
<td>Zero-day attacks are serious threats that may have a significant impact on the organizations affected.</td>
</tr>
<tr>
<td>Zero-day attacks last between 19 days and 30 months, with a median of 8 months and an average of approximately 10 months.</td>
<td>Zero-day attacks are not detected in a timely manner using current policies and technologies.</td>
</tr>
<tr>
<td>Most zero-day attacks affect few hosts, with the exception of a few high-profile attacks (e.g., Stuxnet).</td>
<td>Most zero-day vulnerabilities are employed in targeted attacks.</td>
</tr>
<tr>
<td>58% of the anti-virus signatures are still active at the time of writing.</td>
<td>Data covering 4 years is not sufficient for observing all the phases in the vulnerability lifecycle.</td>
</tr>
<tr>
<td>After zero-day vulnerabilities are disclosed, the number of malware variants exploiting them increases 183–85,000 times and the number of attacks increases 2–100,000 times.</td>
<td>The public disclosure of vulnerabilities is followed by an increase of up to five orders of magnitude in the volume of attacks.</td>
</tr>
<tr>
<td>Exploits for 42% of all vulnerabilities employed in host-based threats are detected in field data within 30 days after the disclosure date.</td>
<td>Cyber criminals watch closely the disclosure of new vulnerabilities, in order to start exploiting them.</td>
</tr>
</tbody>
</table>
Impact of disclosure

(a) Attacks exploiting zero-day vulnerabilities before and after the disclosure (time = $t_0$).

(b) Malware variants exploiting zero-day vulnerabilities before and after disclosure (time = $t_0$).

Figure 6: Impact of vulnerability disclosures on the volume of attacks. We utilize logarithmic scales to illustrate an increase of several orders of magnitude after disclosure.
Figure 7: Time before vulnerabilities disclosed between 2008–2011 started being exploited in the field. The histograms group the exploitation lag in 3-month increments, after disclosure, and the red rug indicates the lag for each exploited vulnerability. The zero-day attacks are excluded from this figure.
The zero-day attacks they identify lasted between
19 days (CVE-2010-0480) and
30 months (CVE-2010-2568), and
the average duration of a zero-day attack is 312 days.

Before We Knew It An Empirical Study of Zero-Day Attacks In The Real World