Christian Wieser, Marko Laakso and Henning Schulzrinne

SIP Robustness Testing for Large-Scale Use
Motivation

Software vulnerabilities prevail:

“Fragile and insecure software continues to be a major threat to a society increasingly reliant on complex software systems.”
- Anup Ghosh [Risks Digest 21.30]

Our purpose:

“To study, evaluate and develop methods of implementing and testing application and system software in order to prevent, discover and eliminate implementation level security vulnerabilities in a pro-active fashion. Our focus is on implementation level security issues and software security testing.”
Dominant security problems

- From ICAT vulnerability statics

<table>
<thead>
<tr>
<th>Vulnerability Type</th>
<th>2003</th>
<th>2002</th>
<th>2001</th>
<th>2000</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input Validation Error</td>
<td>526 (52%)</td>
<td>661 (51%)</td>
<td>744 (49%)</td>
<td>359 (36%)</td>
</tr>
<tr>
<td>(Boundary Condition Error)</td>
<td>81 (8%)</td>
<td>22 (2%)</td>
<td>51 (3%)</td>
<td>66 (7%)</td>
</tr>
<tr>
<td>(Buffer Overflow)</td>
<td>236 (23%)</td>
<td>288 (22%)</td>
<td>316 (21%)</td>
<td>190 (19%)</td>
</tr>
<tr>
<td>Access Validation Error</td>
<td>92 (9%)</td>
<td>121 (9%)</td>
<td>126 (8%)</td>
<td>168 (17%)</td>
</tr>
<tr>
<td>Exceptional Condition Error</td>
<td>152 (15%)</td>
<td>117 (9%)</td>
<td>146 (10%)</td>
<td>119 (12%)</td>
</tr>
<tr>
<td>Environment Error</td>
<td>3 (0%)</td>
<td>10 (1%)</td>
<td>36 (2%)</td>
<td>19 (2%)</td>
</tr>
<tr>
<td>Configuration Error</td>
<td>49 (5%)</td>
<td>67 (5%)</td>
<td>74 (5%)</td>
<td>82 (8%)</td>
</tr>
<tr>
<td>Race Condition</td>
<td>17 (2%)</td>
<td>22 (2%)</td>
<td>50 (3%)</td>
<td>21 (2%)</td>
</tr>
<tr>
<td>Design Error</td>
<td>266 (26%)</td>
<td>407 (31%)</td>
<td>339 (26%)</td>
<td>166 (17%)</td>
</tr>
<tr>
<td>Other</td>
<td>18 (2%)</td>
<td>2 (0%)</td>
<td>8 (1%)</td>
<td>14 (1%)</td>
</tr>
</tbody>
</table>

- Dominance of “Input Validation Error”
Our approach - in a nutshell

Today, thousands of gifted and patient, but uncoordinated monkeys are pounding different products in order to reveal vulnerabilities.

Think of us as rather dumb monkeys using a monkey-machine and systematic methodology to eliminate the most trivial ones.
PROTOS project

- Security Testing of Protocol Implementations
- Results:
  - A novel (mini-simulation) vulnerability testing method developed
  - Several papers and test suites published
- Continuation:
  - Spin-off company Codenomicon Ltd
  - OUSPG will continue with public research
c07-sip Robustness Test Suite

- Applying the PROTOS approach in SIP
  - SIP matures from academic interest to an industry deployed protocol
- Extending the work done in
  - SIP Torture Test Messages
- RFC3261 compliant
- Working on the awareness front
  - SIPit’s
  - Interaction during vulnerability process
c07-sip Design

- Mutating SIP INVITE-requests to simulate attacks to the Software Under Test (SUT).
  - 54 test groups
  - 4527 test cases
- Available as Java JAR-package
- UDP as only injection vector
- Teardown with
  - CANCEL/ACK messages
- Valid-case as minimal instrumentation
c07-sip Results

- Approach new to SIP scene
- Alarming rates of failed subjects
  - Nine implementations (6 UA, 3 servers) tested
    - 1 passed
    - 8 failed in various test-groups
  - For demonstration purpose
    - 2 working exploits

“Hitting the Granny with a stick”? 
Vulnerability Process

- **Vulnerability process: Phases**
  - **Development**
    - Creating and wrapping-up the test-suite
    - Internally testing the available implementations
  - **Pre-release**
    - Involvement of neutral third party (in this case CERT/CC)
    - Notifying respective vendors of any vulnerabilities found
    - Distributing the test-suite to identified vendors implementing the chosen protocol
    - Vulnerability and advisory coordination
    - Grace period
  - **Release**
    - Deploying the test-suite for public perusal
    - Collecting feedback
    - Reiterating either with same or next protocol

Timeline:
- **Development**
  - 2002-10-01
  - SiPit11
- **Pre-release**
  - 2002-11-01
  - 2002-12-01
  - 2003-01-01
- **Release**
  - 2003-02-01
  - 2003-03-01

OUSPG [http://www.ee.oulu.fi/research/ouspg]
Summary

- Noticeable amount of vulnerabilities found
- Awareness on Implementation Level Vulnerabilities among vendors non equally distributed
- Vulnerability process seems new to SIP community
- Fair amount of interest
  - as of 2004-02: around 2500 test material downloads
- Further information:
  - http://www.ee.oulu.fi/research/ouspg/protos/testing/c07/sip/