



Christian Wieser and Marko Laakso

SIP Robustness Testing



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

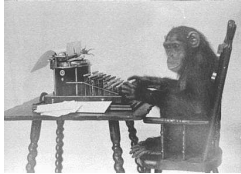
Vulnerability Type	2003	2002	2001	2000
Input Validation Error	526 (52%)	661 (51%)	744 (49%)	359 (36%)
(Boundary Condition Error)	81 (8%)	22 (2%)	51 (3%)	66 (7%)
(Buffer Overflow)	236 (23%)	288 (22%)	316 (21%)	190 (19%)
Access Validation Error	92 (9%)	121 (9%)	126 (8%)	168 (17%)
Exceptional Condition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

- 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.



Visual by <http://www.PDFimages.com>

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

OUSPG (<http://www.ec.europa.eu/research/ouspg>)



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

OUSPG (<http://www.ec.europa.eu/research/ouspg>)



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

OUSPG (<http://www.ec.europa.eu/research/ouspg>)



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



Test case example

- | | |
|--|---|
| <ul style="list-style-type: none"> ❑ #0: validcase <pre> INVITE sip:there@10.10.10.10 SIP/2.0 Via: SIP/2.0/UDP 10.10.1.1:5060;branch=[*] From: 0 <slipme@10.10.1.1>;tag=0 To: Receiver <slipthere@10.10.10.10> Call-ID: 1 @ 10.10.1.1 CSeq: 1 INVITE Contact: 0 <slipme@10.10.1.1> Expires: 1200 Max-Forwards: 70 Content-Type: application/sdp Content-Length: 128 v=0 o=0 0 0 IN IP4 10.10.1.1 s=Session SDP c=IN IP4 10.10.1.1 t=0 0 m=audio 9876 RTP/AVP 0 a=rtpmap:0 PCMU/8000 </pre> | <ul style="list-style-type: none"> ❑ #3: (overflow in SIP-method) <pre> aaaaaaaaaaaaaaaa sip:there@10.10.10.10 SIP/2.0 Via: SIP/2.0/UDP 10.10.1.1:5060;branch=[*] From: 0 <slipme@10.10.1.1>;tag=3 To: Receiver <slipthere@10.10.10.10> Call-ID: 1 @ 10.10.1.1 CSeq: 1 INVITE Contact: 3 <slipme@10.10.1.1> Expires: 1200 Max-Forwards: 70 Content-Type: application/sdp Content-Length: 128 v=0 o=3 3 3 IN IP4 10.10.1.1 s=Session SDP c=IN IP4 10.10.1.1 t=0 0 m=audio 9876 RTP/AVP 0 a=rtpmap:0 PCMU/8000 </pre> |
|--|---|



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



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/>

OUSPG (<http://www.ee.oulu.fi/research/ouspg/>)
