Static Analysis Software Assurance Tools

and

SATE 2010
Agenda

• Introduction
• Dovecot - Review of Results
  – SATE Manual Findings
  – LDRA Analysis
• Conclusion
Secure Code
Dependable
Trustworthy
Resilient
2 types of static analysis; Software Assurance & Defect Detection
- Robert Seacord
LDRA tool suite® = Software Assurance
Software Assurance

Assesses:
Dependability
Trustworthiness
LDRA tool suite Analysis Results

Vulnerabilities: 0

Quality &/or CERT-C Violations: 16066
<table>
<thead>
<tr>
<th>UID</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>43476</td>
<td>Sizeof argument is a pointer.</td>
</tr>
<tr>
<td>48494</td>
<td>Sizeof argument is a pointer.</td>
</tr>
<tr>
<td>38109</td>
<td>Pointer not checked for null before use.</td>
</tr>
<tr>
<td>38509</td>
<td>Pointer not checked for null before use.</td>
</tr>
<tr>
<td>52231</td>
<td>free parameter is not heap item.</td>
</tr>
</tbody>
</table>
Sizeof argument is a pointer

void mail_index_view_clone(struct mail_index_view *dest, ...) {
    memset(dest, 0, sizeof(dest));
    ...
}

CERT-C EXP01-C
Quality issue
Security issue
LDRA Review
Dependability

Objective
verify system quality

Software Assurance
pinpoints potential quality issues
Many areas of high complexity
Testability compromised
Trustworthiness

Objective
Identify potential security violations

Software Assurance
CERT-C coding standard enforcement identifies potential security issues
No NULL pointer checking

Failed memory allocations result in `abort()`
static struct stack_block *mem_block_alloc(size_t min_size)
{
    ...

    #ifndef USE_GC
        block = malloc(SIZEOF_MEMBLOCK + alloc_size);
    #else
        block = GC_malloc(SIZEOF_MEMBLOCK + alloc_size);
    #endif

    if (unlikely(block == NULL)) {
        if (outofmem) {
            if (min_size > outofmem_area.block.left)
                abort();
            return &outofmem_area.block;
        }
    ...

    return block;
}
No evident security violations

**BUT**

Areas of concern:
- Quality issues (e.g. sizeof pointer usage)
- High Complexity implies incomplete testing
- Custom memory management potential security violation
Thank You!!
Nat Hillary
Field Applications Engineer
Tel: +1 (206) 605 3755
Email: nat.hillary@ldra.com

LDRA
www.ldra.com

Lake Amor Office Park,
150 Bayhill Drive, Suite # 360
San Bruno, CA 94066
Tel: +1 (650) 593 8880
Fax: +1 (650) 583 8882

Code Coverage
Source Code Analysis
System and Unit Testing
Requirements Traceability