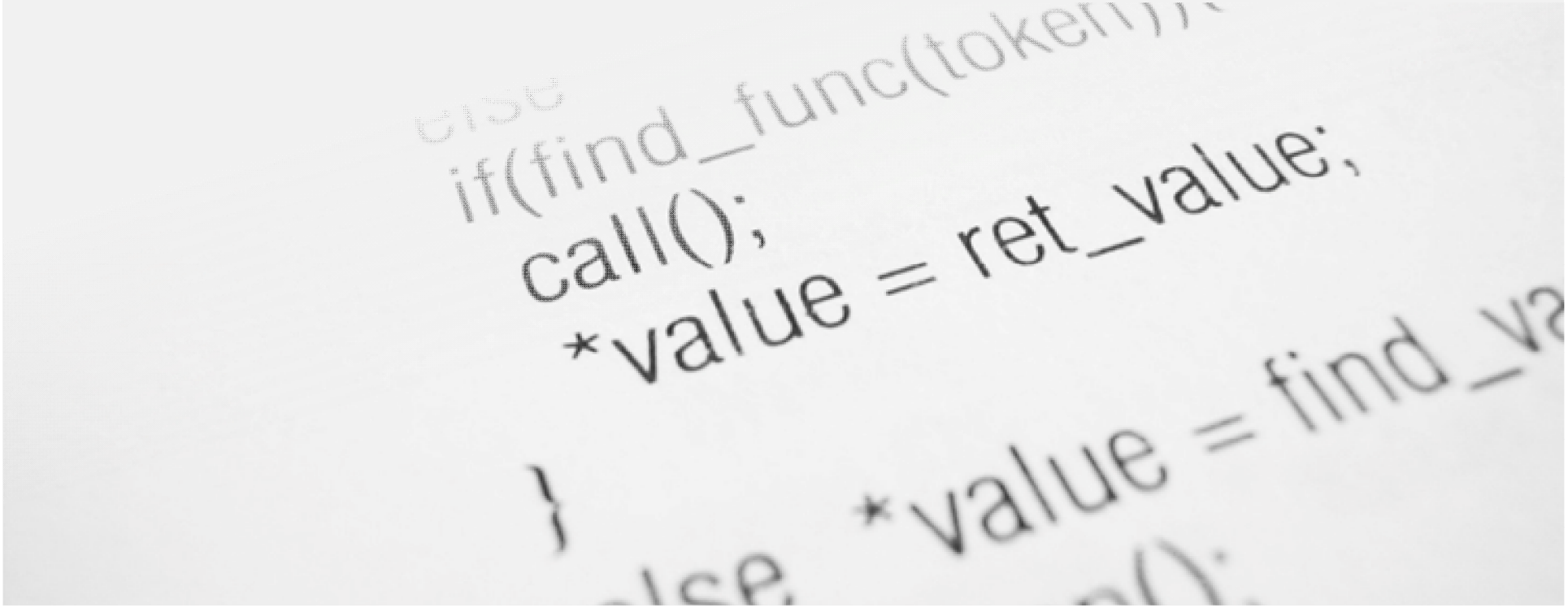
Static Analysis Software Assurance Tools and SATE 2010

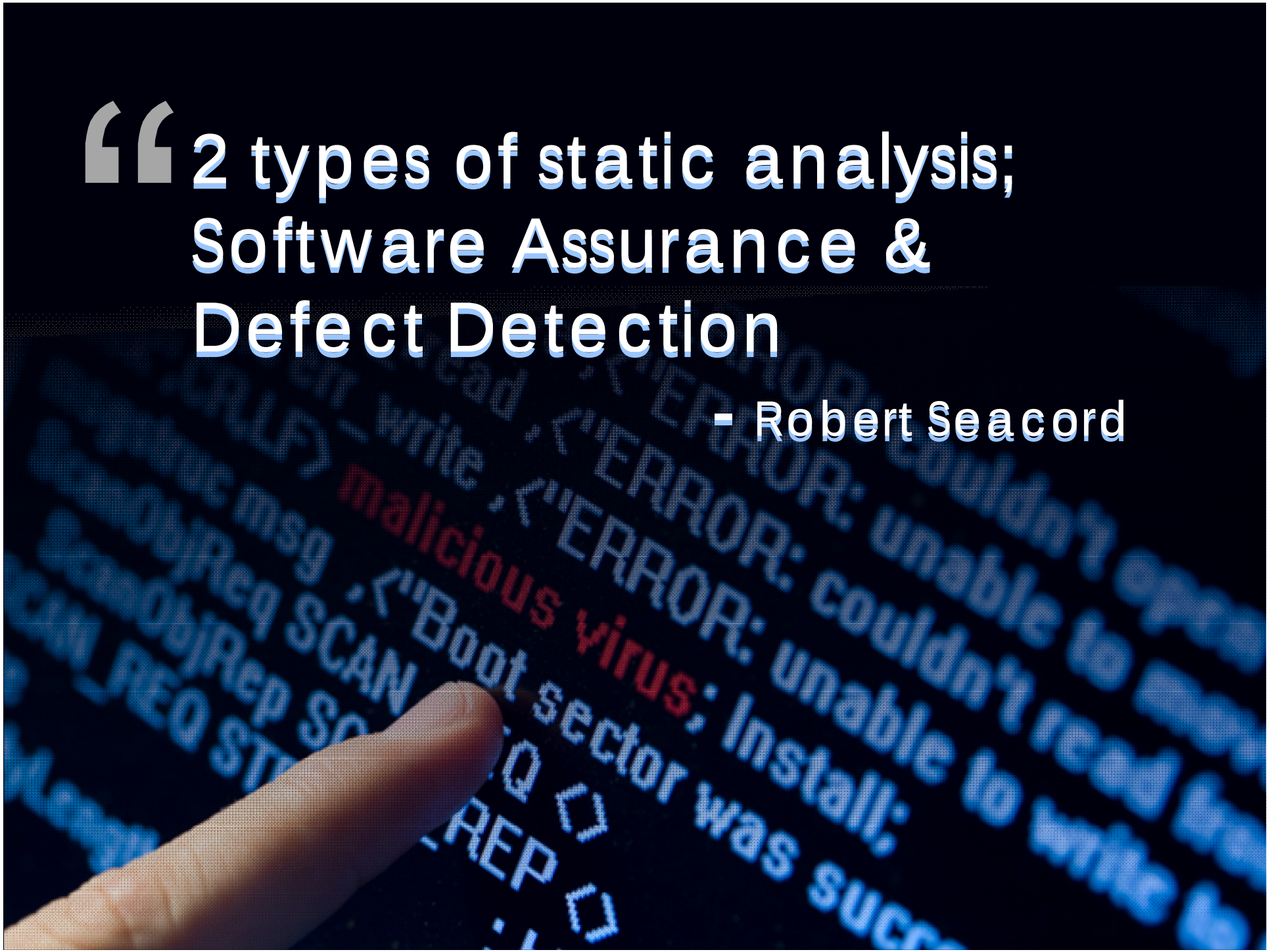
Agenda

- 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

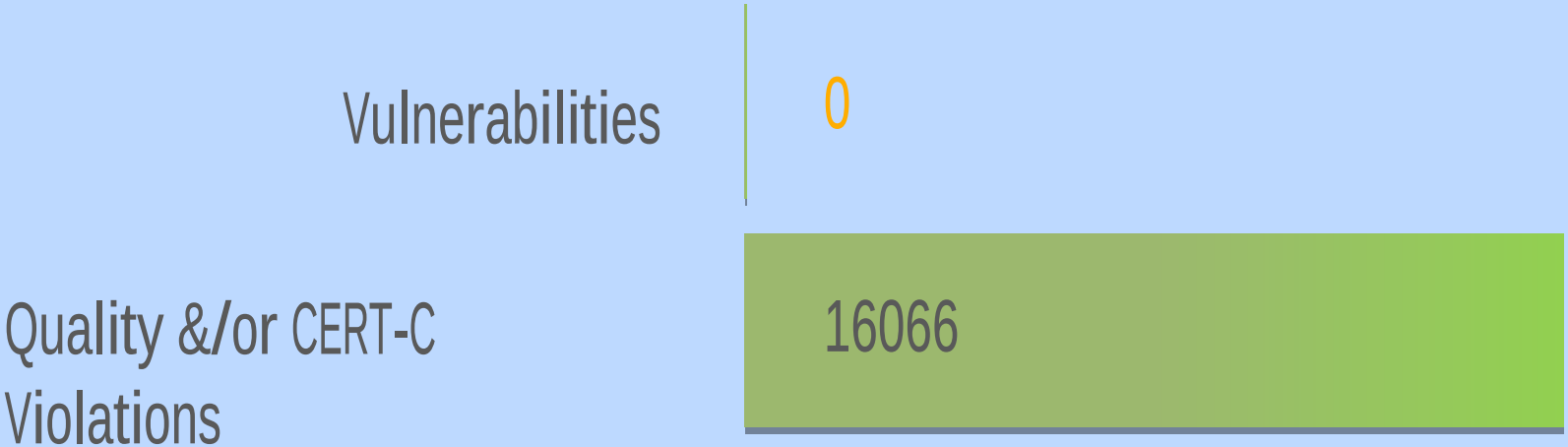

LDRA tool suite®
=
Software Assurance

Software Assurance

Assesses:
Dependability
Trustworthiness


DOVECOT
Inkscape version 0.42.2 - ezechiele 17 09 2005

LDRA tool suite Analysis Results



Manual Results Review

UID	Description
43476	Sizeof argument is a pointer.
48494	Sizeof argument is a pointer.
38109	Pointer not checked for null before use.
38509	Pointer not checked for null before use.
52231	free parameter is not heap item.

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

Software Technology

LDRA Review

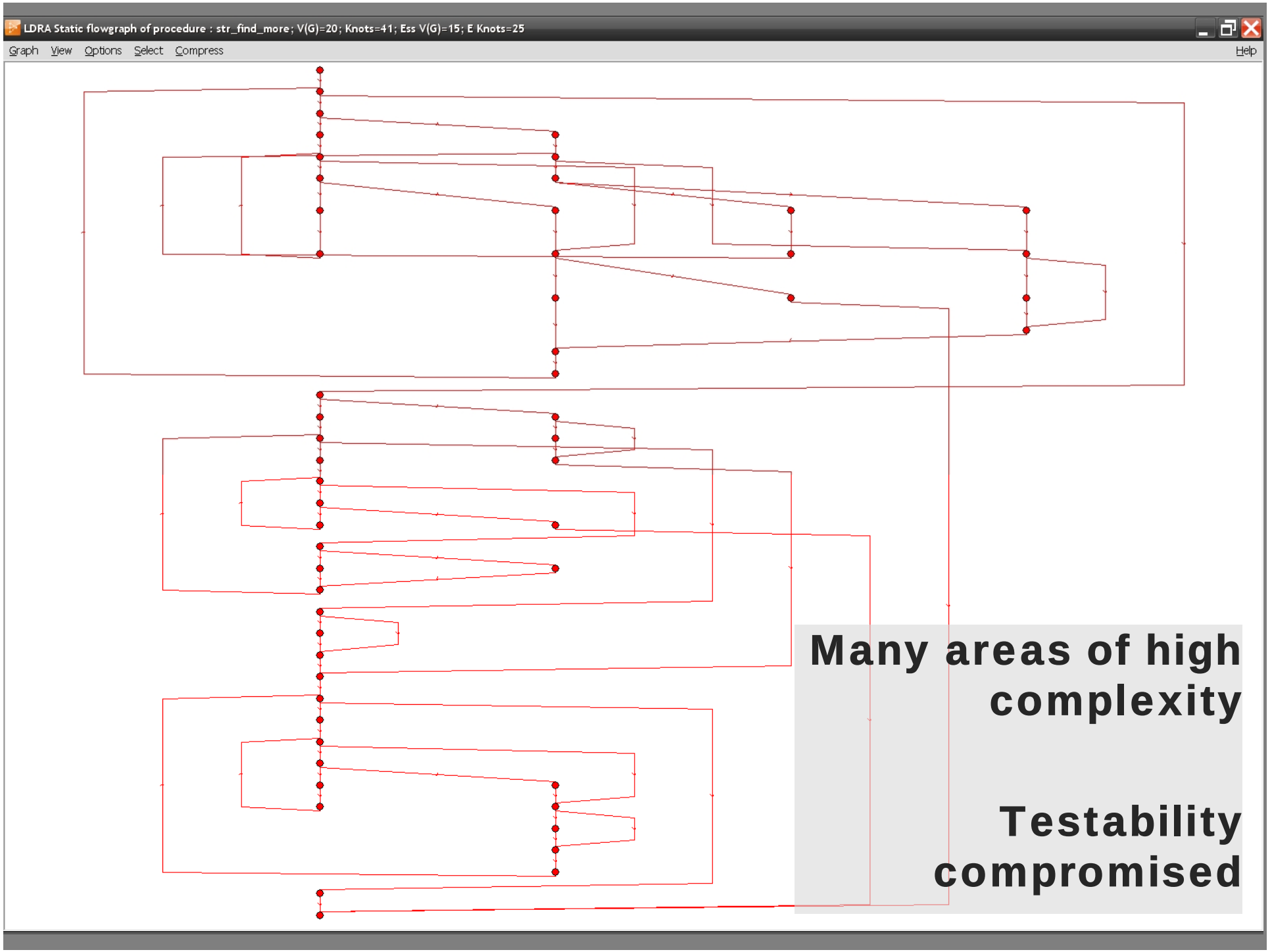
Dependability

Objective

verify system quality

Software Assurance

pinpoints potential quality issues


LDRA Static flowgraph of procedure : str_find_more; V(G)=20; Knots=41; Ess V(G)=15; E Knots=25
Graph View Options Select Compress Help
Many areas of high complexity
Testability compromised

Trustworthiness

Objective

Identify potential security violations

Software Assurance

CERT-C coding standard enforcement identifies potential security issues

CERT-C Standard Review

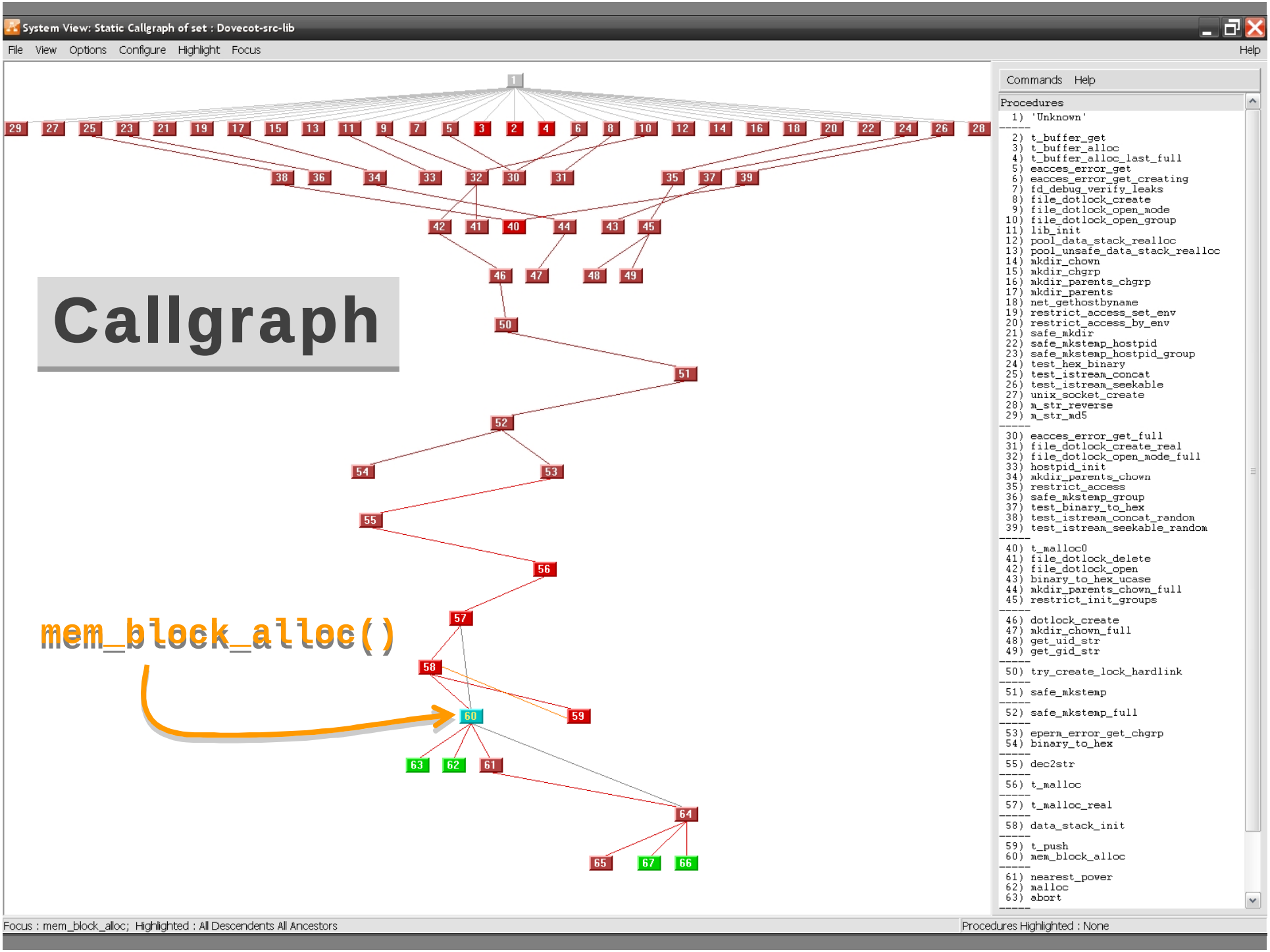
No NULL pointer checking

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 cleanup code
- Security risk per CERT-C ERR06-C


System View: Static Callgraph of set : Dovecot-src-lib
File View Options Configure Highlight Focus Help
Callgraph
mem_block_alloc()
Commands Help
Procedures
1) 'Unknown'
2) t_buffer_get
3) t_buffer_alloc
4) t_buffer_alloc_last_full
5) eacces_error_get
6) eacces_error_get_creating
7) fd_debug_verify_leaks
8) file_dotlock_create
9) file_dotlock_open_mode
10) file_dotlock_open_group
11) lib_init
12) pool_data_stack_realloc
13) pool_unsafe_data_stack_realloc
14) mkdir_chown
15) mkdir_chgrp
16) mkdir_parents_chgrp
17) mkdir_parents
18) net_gethostbyname
19) restrict_access_set_env
20) restrict_access_by_env
21) safe_mkdir
22) safe_mkstemp_hostpid
23) safe_mkstemp_hostpid_group
24) test_hex_binary
25) test_istream_concat
26) test_istream_seekable
27) unix_socket_create
28) m_str_reverse
29) m_str_md5
30) eacces_error_get_full
31) file_dotlock_create_real
32) file_dotlock_open_mode_full
33) hostpid_init
34) mkdir_parents_chown
35) restrict_access
36) safe_mkstemp_group
37) test_binary_to_hex
38) test_istream_concat_random
39) test_istream_seekable_random
40) t_malloc0
41) file_dotlock_delete
42) file_dotlock_open
43) binary_to_hex_ucase
44) mkdir_parents_chown_full
45) restrict_init_groups
46) dotlock_create
47) mkdir_chown_full
48) get_uid_str
49) get_gid_str
50) try_create_lock_hardlink
51) safe_mkstemp
52) safe_mkstemp_full
53) eperm_error_get_chgrp
54) binary_to_hex
55) dec2str
56) t_malloc
57) t_malloc_real
58) data_stack_init
59) t_push
60) mem_block_alloc
61) nearest_power
62) malloc
63) abort
Focus : mem_block_alloc; Highlighted : All Descendents All Ancestors
Procedures Highlighted : None

Dovecot Review - Conclusion

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!!


LDRA
www.ldra.com
Lake Amir Office Park,
1250 Bayhill Drive, Suite # 360
San Bruno, CA 94066
t : +1 (650) 583 8880
f : +1 (650) 583 8881
Nat Hillary
Field Applications Engineer
c : +1 (206) 605-3755
e : nat.hillary@ldra.com
Code Coverage
Source Code Analysis
System and Unit Testing
Requirements Traceability

www.ldra.com

info@ldra.com