Goanna Static Analysis Tool at SATE

Ansgar Fehnker
ansgar.fehnker@RedLizards.com
www.RedLizards.com
About Us

- R&D spin-out
- 5 years technology research
- Funded and backed by NICTA
What We Do

Goanna Static Analysis for C/C++

Inspect code automatically for

- memory corruption and leaks
- software quality issues
- security vulnerabilities
- API rule violation
- coding standards violations
- identifies >100 types of serious defects

Does not execute, but investigate code.
Products

Goanna Studio

IDE integrated static analysis
- Visual Studio 2005-2010 on Windows
- Eclipse on Linux

Goanna Central

Server / command line version
- Linux
- Windows/MSBuild (beta)
Under The Hood
Goanna Architecture

**Build**
- make
- cmake
- scon
- MSVS
- MSBuild

**Languages & Compilers**
- C/C++
- ARM Assembly
- gcc 4.4
- MS Vstudio

**Input:** Check Queries (Language)

**User Defined Checks/Queries**
- False Path Elimination
- Interprocedural Analysis

**Model Generation**
- Interval Constraint Solving

**Model Checking**

**Output:** Warnings & Traces

**IDE & Tools**
- VStudio10
- VStudio08
- VStudio05
- Eclipse CDT

**Warning Manager & Metrics**
Goanna Core Analysis

Source Code

```c
int main(void) {
    int i,a=0;
    int *p = (int *) malloc(sizeof(int));
    for (i=1000; i > 0; i--){
        a = *p + i;
        i = i*2;
    }
    ...
```
Goanna Core Analysis

Source Code
int main(void) {
    int i, a = 0;
    int *p = (int *) malloc(sizeof(int));
    for (i = 1000; i > 0; i--){
        a = *p + i;
        i = i*2;
    ...

Syntactic Pattern

Temporal Pattern

What happens?

When does it happen?
Goanna Core Analysis

```
int main(void) {
    int i,a=0;
    int *p = (int *) malloc(sizeof(int));
    for (i=1000; i > 0; i--){
        a = *p + i;
        i = i*2;
    }
    ...
```
Goanna Core Analysis

### Source Code

```c
int main(void) {
    int i,a=0;
    int *p = (int*) malloc(sizeof(int));
    for (i=1000; i > 0; i--){
        a = *p + i;
        i = i*2;
    ...
```

### Syntactic Pattern

- **Automatic Translation**

### Temporal Pattern

### Model

- **AG decl => A !use W write**

- **decl**
- **write**

- **Model Checker**
Goanna Core Analysis

Source Code

```c
int main(void) {
    int i, a = 0;
    int *p = (int *) malloc(sizeof(int));
    for (i = 1000; i > 0; i--){
        a = *p + i;
        i = i*2;
        ...
    }
}
```

Syntactic Pattern

Automatic Translation

Temporal Pattern

Model

AG decl => A !use W write

decl

write

Warnings

1 Goanna - Pointer p used a
2 Goanna - Uninitialised va
3 Goanna - Dead Code found

Model Checker

Trace

Line 1 Decl
Line 2 Decl *
Line 3 For-loop
Line 4 Exp *
Example: Uninitialized Variable

```c
int foo(int n) {
    int x = 0, y = 1, q, i = 0;
    do {
        int oldy = y;
        y = x;
        q = x + oldy;
        x = q;
        i++;
    } while(i < n);
    return q;
}
```
Example: Uninitialized Variable

int foo(int n) {
    int x = 0, y = 1, q, i = 0;
    do {
        int oldy = y;
        y = x;
        q = x + oldy;
        x = q;
        i++;
    } while(i < n);
    return q;
}
Example: Uninitialized Variable

```c
int foo(int n) {
    int x = 0, y = 1, q, i = 0;
    do {
        int oldy = y;
        y = x;
        q = x + oldy;
        x = q;
        i++;
    } while(i < n);
    return q;
}
```

Temporal Specification

\[ \text{Forall } \var \text{ Never read Before write} \]
Example: Uninitialized Variable

```c
int foo(int n) {
    int x = 0, y = 1, q, i = 0;
    do {
        int oldy = y;
        y = x;
        q = x + oldy;
        x = q;
        i++;
    } while(i < n);
    return q;
}
```

**Temporal Specification**

Forall var Never read Before write

**Output**

Goanna - analyzing file
Number of functions: 1
Total runtime : 0.01 second
Example: Uninitialized Variable

```c
int foo(int n) {
    int x = 0, y = 1, q, i = 0;
    do {
        int oldy = y;
        y = x;
        q = x + oldy;
        x = q;
        i++;
    } while(i < n);
    return q;
}
```

Temporal Specification

Forall var Never read Before write

Output

Goanna - analyzing file
Number of functions: 1
Total runtime : 0.01 second

Note

Completely Automatic Analysis
Goanna Core

- Towards Software Model Checking
- Towards Abstract Interpretation
- Model Generation
- Model Checking
- False Path Elimination
- Interprocedural Analysis
- Interval Constraint Solving
Goanna in SATE
Goanna setup for SATE

55 default checks for C/C++

- Geared towards quality issues
- Targeted at “Must Fix” and “Fix if time” issues.
- Omitted checks for “cosmetic issues”
Results Overall

Number of warnings

<table>
<thead>
<tr>
<th>Program</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chrome 375.54</td>
<td>1079</td>
</tr>
<tr>
<td>Chrome 375.70</td>
<td>1173</td>
</tr>
<tr>
<td>Dovecot</td>
<td>180</td>
</tr>
<tr>
<td>Wireshark 2.0</td>
<td>534</td>
</tr>
<tr>
<td>Wireshark 2.9</td>
<td>532</td>
</tr>
</tbody>
</table>

Top 10

<table>
<thead>
<tr>
<th>Warning Type</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>PTR-null-pos-assign</td>
<td>952</td>
</tr>
<tr>
<td>RED-cmp-never</td>
<td>788</td>
</tr>
<tr>
<td>RED-cmp-always</td>
<td>448</td>
</tr>
<tr>
<td>PTR-null-cmp-aft</td>
<td>328</td>
</tr>
<tr>
<td>SPC-uninit-var-some</td>
<td>189</td>
</tr>
<tr>
<td>PTR-null-assign-fun-pos</td>
<td>144</td>
</tr>
<tr>
<td>RED-unused-val-ptr</td>
<td>124</td>
</tr>
<tr>
<td>PTR-param-unchk-some</td>
<td>94</td>
</tr>
<tr>
<td>RED-unused-var-all</td>
<td>67</td>
</tr>
<tr>
<td>RED-case-reach</td>
<td>46</td>
</tr>
</tbody>
</table>

PTR: Pointer misuse
RED: Redundant code
SPC: Unspecified behavior
A Closer Look
Semantic attributes are a GNU language extension

`uni_ucs4_to_titlecase` has `__attribute__((const))` (see `unichar.h`)

`uint16_find` has not

GNU says: “(...) a function that calls a non-const function usually must not be const”
RED-cmp-never

if (str_array_length(args) != 2 ||
    director_args_parse_ip_port(conn, args, &ip, &port) < 0) {
    i_error("director(%s): Invalid CONNECT args", conn->name);
    return FALSE;
}

- `director_args_parse_ip_port()` only returns TRUE or FALSE
- `director_args_parse_ip_port()<0` never true
- `ip` and `port` might not be assigned, but this failure is not detected
Parameter `nti' is not checked against NULL before it is dereferenced on some paths, but on other paths it is.

- `nti` checked in one branch, but not the other
- Pointer `nti` can be null and is passed to `dissect_smb2_ioctl_data`
- Related to CVE-2010-2283
- Label `execute_next_instruction` in line 335,
- `switch` from line 344 to 2750 with 36 cases,
- 35 `goto execute_next_instruction`
- 34 increments of `used_udvm_cycles`
Not Found: CVE-2010-2286

- Problem: Infinite loop possible.
- Need: Show absence of loop-invariant for a goto-structure
- Do we want to spend resources on find this?
- Or advise to use a proper for-loop.
Not Found: CVE-2010-2286

- Problem: Infinite loop possible.
- Need: Show absence of loop-invariant for a goto-structure
- Do we want to spend resources on find this?
- Or advise to use a proper for-loop.

Programmer wisdom:
Always code as if the person who ends up maintaining your code is a violent psychopath who knows where you live.
Summary

- Goanna is a static analysis solution for C/C++
- Desktop and server version available at redlizards.com
- It uses a combination of model checking and static analysis to find serious bugs
- It did find serious bugs
- It is named after a bug-eating lizard