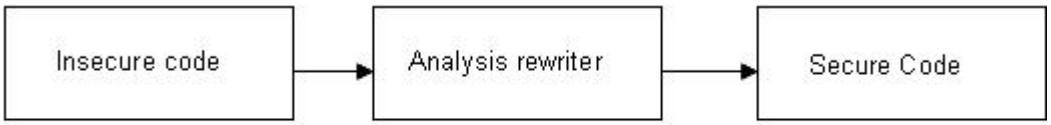
Lecture Notes: February 22, 2007

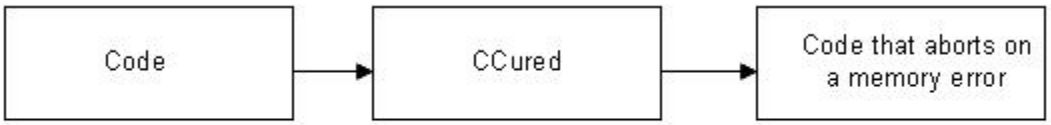
The checking model can be shown as:



- Motivation:
  - To hard to fix all the bugs.
  - Sometimes it is just too hard to write a good static analysis.

  This is the reason to go towards rewriters.

CCured (Memory Safety) (Also known as type safety).



CCured converts code injection vulnerabilities into Denial-of-Service vulnerabilities.

We, therefore, divide 'C' into three different languages:

1) SAFE:

- Just like 'C' except
  - No casts (except pointer -> int)
  - No pointer arithmetic
  - No Arrays
  - No unions

Allowed Instructions:

```
int x;
int *p = &x;
char *p = malloc(1);
int y = (int) q;
free(q);
```

Making SAFE 'safe':

- In the case of NULL pointers ----- insert 'NULL' checks
- Uninitialized pointers ----- always init to NULL.
- For dangling pointers ------ Never free() memory
  (can add garbage collectors to reclaim some memory)

```
//free(q);
if(q == NULL)
  abort();
*q = 5;
```

2) SEQ:

- SAFE plus
  - Pointer arithmetic
  - Casts int -> pointers
  - Arrays

Making SEQ 'safe':

- Initialize all pointers
- No free
- NULL checks
- Bounds check pointers
  (need to keep track of bounds)

Therefore, we use a tuple notation as follows:
pointer = < base, size, p>
int =<i>

When int <i> is cast to a pointer, we convert it to <0,0,i>, which is a pointer that cannot be dereferenced. This is safe, but may break some correct programs.

```
int *p = malloc(10);
int x;
int *q = &x;
p +=5;
*p = 0;
```

The above piece of code can be shown in SEQ as follows:

```
int *t = malloc(10);
int *p = <t, t==NULL?0:10, t>
int x;
int *q = <&x, sizeof(x), &x>
p = <t, 10, t+5>
if( p == NULL || p<t || p>=t+10 -sizeof(*p))
abort();
*p = 0;
```

Question: How does CCured deal with escaping pointers, e.g

```
int *foo(void)
{
    int x;
    return &x;    //This is sort of dangling pointer.
}
```

- Convert SEQ pointers to SAFE pointers

SEQ - > SAFE:

if(P_{SEQ}< base || P>= base + size – sizeof(*p))
    abort();
P_{SAFE}= P_{SEQ};

SAFE -> SEQ:

P_{SEQ} = <P_{SAFE}, sizeof(*P_{SAFE}), P_{SAFE} >

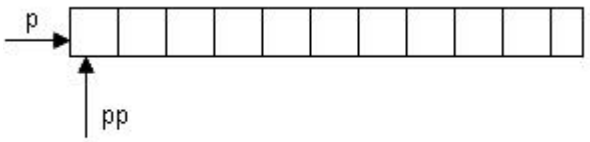
3) DYNAMIC (aka WILD):

- SEQ plus
  - Casts
  - Unions
  - Need all checks of SEQ

Suppose, we have the following case:

```
int **pp;
int *p;
p = malloc(10);
pp = (int **) p;
```

The above statements can be shown as:



- LSB of every word indicates whether it is a pointer or not.
- Runtime checking for type of word

RESULTS: Programs run about 50% slower.
PROBLEM: - May require substantial modifications.
- Shared Libraries.