MANCHESTER 1824
The University of Manchester

Using BMC for Firmware Analysis

Rafael Menezes
Dr. Lucas Cordeiro

MANCHESTER 1824
The University of Manchester

Overview

- ESBMC is a verification tool capable of proving safety property over C/C++ programs.
- It is an open-source tool (4-clause BSD) available at GitHub

MANCHESTER 1824
The University of Manchester

Extensions

- To start the usage of ESBMC for firmware:

1. Performance
2. Support (C extensions)

MANCHESTER 1824
The University of Manchester

GCC Vector

- This adds support to the Vector type which is a C extension to handle vectors;
- Vectors contains a set of operations: add, sub, convertion, and etc.

```
typedef int v4si __attribute__ ((vector_size (16)));
v4si a, b, c;
c = a + b;
```

MANCHESTER 1824
The University of Manchester

Goto Unwind

- This extension manually unroll bounded loops
- It increases the performance considerably for initialization loops.

```
for(int i = 0; i < 3; i++) foo(i);
```

```
int i = 0;
foo(i++);
foo(i++);
foo(i++);
```

MANCHESTER 1824
The University of Manchester

SSA Caching

The caching comes from the observation that for each increment of K, ESBMC verifies the same VCC again.

```
int y = 1;
for(int x = 0; x < 3;
x++)
{
    y *= 2;
    assert(y <= 8);
}
```

MANCHESTER 1824
The University of Manchester

SSA Caching

The caching comes from the observation that for each increment of K, ESBMC verifies the same VCC again.

```
// K = 1
ASSERT(Y=1 && Y*2 <= 8)

// K = 2
ASSERT(Y=1 && Y*2 <= 8)
ASSERT(Y=1 && Y*2*2 <= 8)
```

MANCHESTER 1824
The University of Manchester

Bit-precise model

- ESBMC handles every struct of the program as a having its sizes in bytes.
- This is mostly fine, because when doing pointer arithmetic for the dereferencing we can get the specific byte of the member.
- However, there are bitfields. These structures allow to repurpose a specific quantity of bits of a member.

MANCHESTER 1824
The University of Manchester

Memset, Memcpy

When working with memcpy and memset for huge arrays (> 4kb) took too much time to verify.

If, the size of the array is not symbolic, we can optimize the entire operation

```
char *ptr =
malloc(1000000);
memset(ptr, 0, 50000);
```

MANCHESTER 1824
The University of Manchester

Any questions?