RapidBBAC: Robustness and Performance Enhancement of Baggy Bounds Accurate Check(BBAC) in SAFECode

Zhengyang Liu*, John Criswell**
*Beijing University of Posts and Telecommunications, **University of Rochester
*zhengyang-liu@hotmail.com, **criswell@cs.rochester.edu

Google Summer of Code 2016

RapidBBAC: Overview

Performance Enhancement
- Analyze the overhead of BBAC by gprof
- 3 new passes for runtime functions inline.

Robustness Enhancement
- 26 bugs are found and fixed.
- Works well on several real world software.

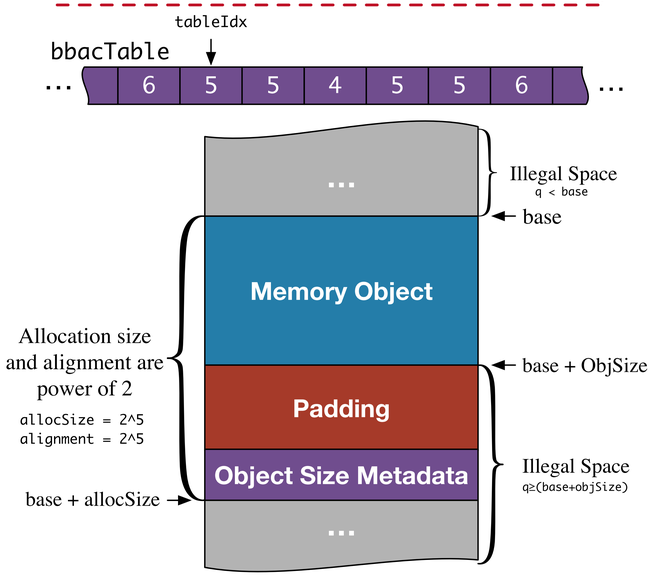
Project URL: https://github.com/zhengyangl/safecode-llvm37

Background: Baggy Bounds Accurate Check

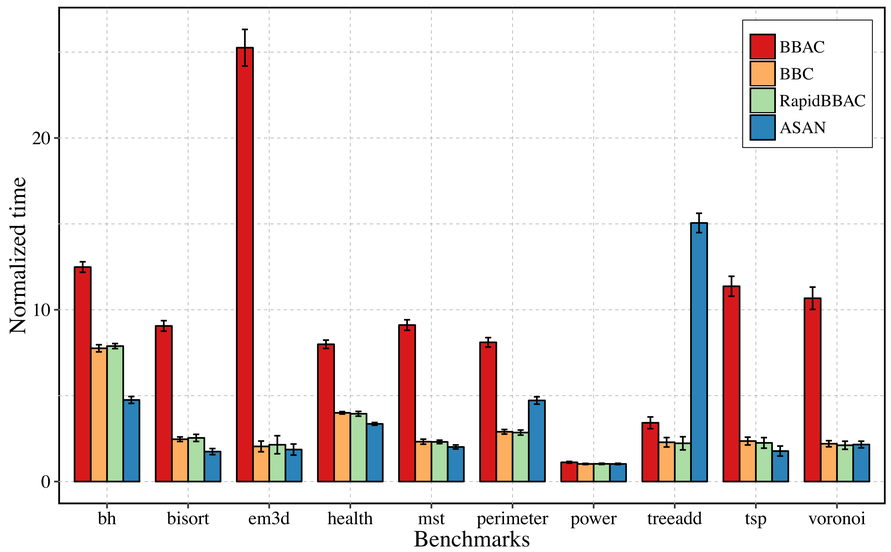
Pointer Arithmetic:
```
q = p + i
```

Get Memory Object Size:
```
tableIdx = p >> log2 (slot_size)
allocSize = 1 << bbacTable[tableIdx]
base = p & ~ (allocSize - 1)
objSize = *(base + allocSize - 4)
```

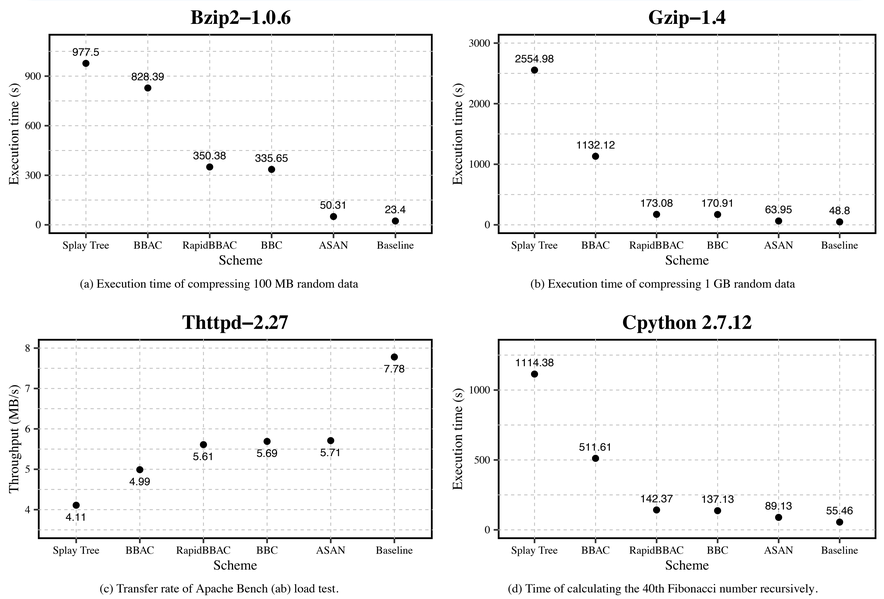
Bound Check:
```
(q >= base) && (q < (base + objSize))
```



Olden Benchmark with Extra Problem Size



Real world software Performance



(a) Execution time of compressing 100 MB random data

(b) Execution time of compressing 1 GB random data

(c) Transfer rate of Apache Bench (ab) load test.

(d) Time of calculating the 40th Fibonacci number recursively.

We thank the Google Summer of Code program for their support.

Robustness Enhancement

Ported from LLVM 3.2.0 to 3.7.0
26 bugs are found and fixed
49 commits, **3799** LOC modified
Tested on Linux, Mac OS X and FreeBSD

Software works well with RapidBBAC

Utilities: gzip, bzip2, git
Interpreters: Cpython, SIOD
Web Servers: Apache Httpd, Nginx, Thttpd
Databases: PostgreSQL, Sqlite3

Future Work

BBAC is more flexible than Address Sanitizer on metadata storage.

Try storing other information on metadata section to support more security hardening techniques.
- Points-to sets → Enforce alias analysis.
- Multithreaded program traces → Find data races.
- Memory access policies → Prevent malicious code injection
- etc…

References

[1] Ding, Baozeng, et al. "Baggy bounds with accurate checking." Software Reliability Engineering Workshops (ISSREW), 2012 IEEE 23rd International Symposium on. IEEE, 2012.
[2] Akritidis, Periklis, et al. "Baggy Bounds Checking: An Efficient and Backwards-Compatible Defense against Out-of-Bounds Errors." USENIX Security Symposium. 2009.