A Robust Shadow Memory Data Structure & API

- Karl Cronburg -

- Oct-25-2014 -

- Polyhack Hackathon -

Scalable & Robust Data Structure

- Can map entire 32-bit address space
- Profile arbitrary desktop / laptop applications
  - Memory patterns:
    - allocations
    - individual machine load & store operations (with help of Valgrind)
    - data-flow analysis
  - Security - integrity, software-based memory permissions

The API

```
22 typedef unsigned char U8; //UChar;
23 typedef int Addr;
24 typedef struct {
25   // granularity == (shadow_bits / application_bits) e.g. 1 shadow bit per 8 bits of
26   //   application memory is all that is required for determining addressability
27   //   (because memory is byte-addressed)
28   short shadow_bits;        // # of shadow bits corresponding to one map entry
29   short application_bits;   // # of app bits corresponding to one map entry
30   void* map;                // pointer to the primary shadow map
31   void* distinguished_maps; // pointer to distinguished maps
32   short num_distinguished;  // # of distinguished maps
33 } ShadowMap;
34
35 // Two primary shadow map operations (get and set)
36 void shadow_get_meta_bits(ShadowMap* PM, Addr a, U8* mbits);
37 void shadow_set_meta_bits(ShadowMap* PM, Addr a, U8  mbits);
38
39 // Initialize and destroy. Initialize sets up the primary map and any distinguished maps.
40 // Destroy frees any memory malloc'd as part of the maps.
41 void shadow_initialize_map(ShadowMap* PM);
42 void shadow_destroy_map(ShadowMap* PM);
43
44 void snapshot(ShadowMap* PM);
```