西安电子科技大学 XIDIAN UNIVERSITY | 广州研究院 Guangzhou institute of technology

Mcds分享报告

汇报人：奚佳新

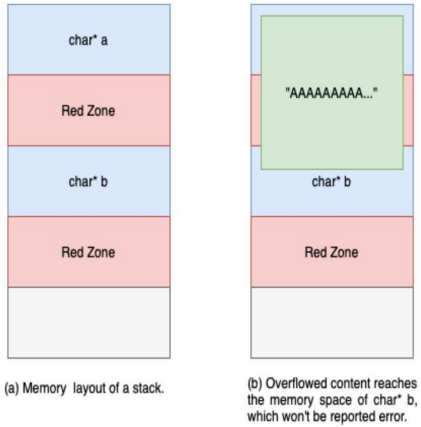
2025-01-17

背景

空间内存损坏是指试图引用超出范围的内存位置。临时内存损坏是指尝试通过已释放的内存指针引用内存位置，或引用曾经有效但现在无效的内存对象。Asan 中使用的红色区域，它能够检测堆栈、堆和全局缓冲区溢出。Asan 通过内存延迟检测临时内存损坏。

当前ASan存在的三个问题

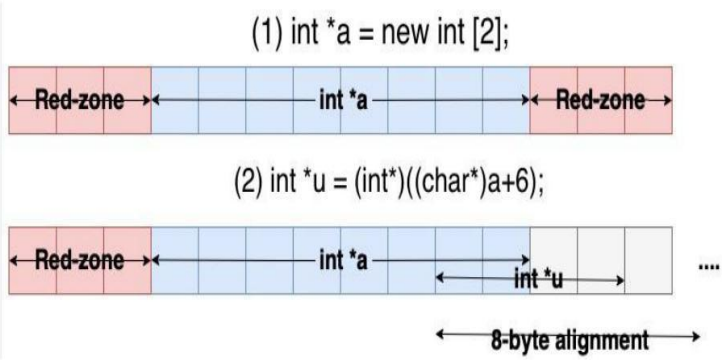
- 案例 1：越界写入超出红色区域。
- 案例 2：部分越界的不对齐访问
- 案例 3：一次性分配和释放了大量内存



(a) Memory layout of a stack.

(b) Overflowed content reaches the memory space of char* b, which won't be reported error.

一个大型的合法写入可能会覆盖整个红色区域并溢出到下一个内存对象的内存空间。Asan不会将此报告为非法访问。因为其只执行粗粒度检查。

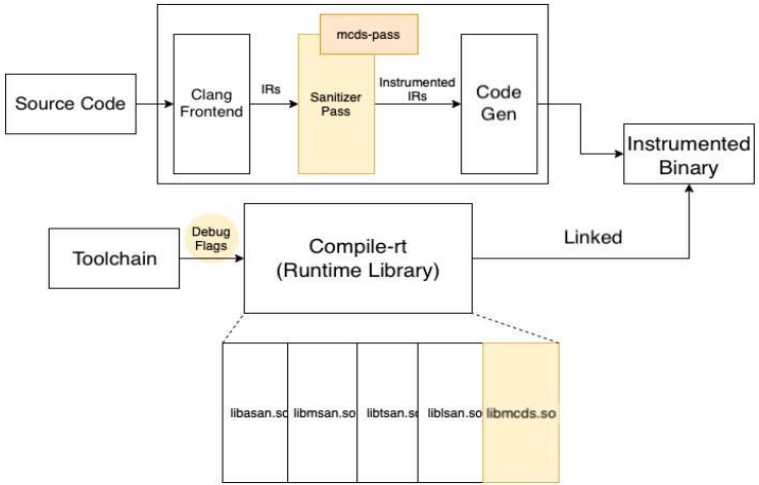


a是一个包含两个 4 字节整数的数组,u是指向一个整数的指针。将u设置为指向a的第二个元素的中间位置。因此,u指向的4字节“整数”与a的红色区域（灰色部分）部分重叠。因此，红色区域变得可访问。此错误无法在编译期间捕获，也无法被当前的Asan捕获。

```
char *a = new char[1 << 20];  // 1MB
delete [] a;  // <<< "free"
char *b = new char[1 << 28];  // 256MB
delete [] b;  // drains the quarantine queue.
char *c = new char[1 << 20];  // 1MB
a[0] = 0;     // "use". May land in 'c'.
```

由于隔离区的大小为256MB，所以导致对于大于256MB的内存回收会直接回收并且清楚隔离区，导致c的内存实际是对a内存的重用，发生了UAF。

Mcds介绍



工作流程

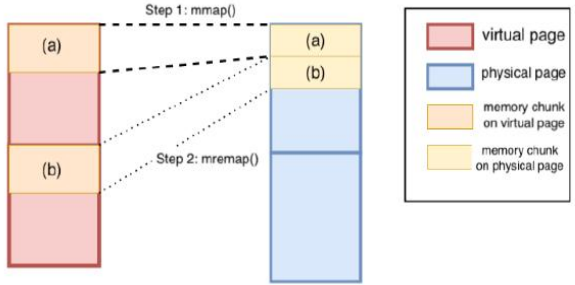
首先，Clang 将源代码编译成 LLVM IR。mcds 将用编译器运行时库中定义的相应 mcds 函数替换对内存内在函数的调用。可执行二进制文件与选定的库链接，生成一个带插桩的二进制文件。一个报告函数也被添加到带插桩的二进制文件中。当发生无效访问或进程崩溃时，报告函数被触发。

Mcds关键思想

mcds是一种内存分配技术，它利用了mprotect和mremap系统调用的特性来创建一个内存分配器，该分配器可以将多个虚拟页面映射到同一个物理页面上。这种方法在虚拟地址空间中创建大的空洞，这些空洞作为“红区”（red zones）。由于多个虚拟页面可以压缩到同一个物理页面中，因此物理内存的浪费很少。访问这些红区会由于虚拟内存机制而引发异常，并且这些异常将被检测到（由分页硬件以很小的运行时开销进行检测）。

内存分配机制（分配）

在分配过程中，每个内存对象被映射到一个单独的虚拟页面。这意味着对于小对象，内存对象之间的间隙几乎是一个页面。如果每个虚拟页面都映射到一个单独的物理页面，这种映射会导致严重的物理内存浪费。利用页面别名来防止浪费内存空间。通过mmap()、munmap()和 mremap()系统调用实现的。它包含两个步骤：



(1) 文件支持的映射

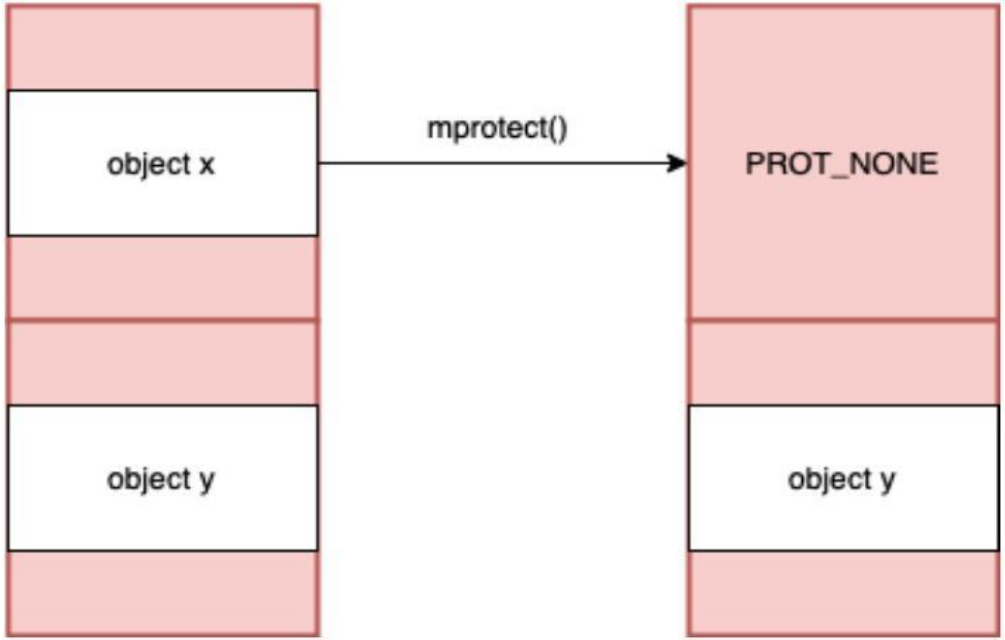
```
mmap(0, 0x1000, PROT_WRITE|PROT_READ, MAP_SHARED);
munmap(addr + size, 0x1000 - size);
```

(2) 重叠

```
mremap(old_addr, 0, 0x1000, MREMAP_FIXED_NOREPLACE,old_addr + 0x1000);
munmap(new_addr + size, 0x1000 - size);
```

内存分配机制（释放）

```
mprotect(freed_page, 0x1000, PROT_NONE);
```



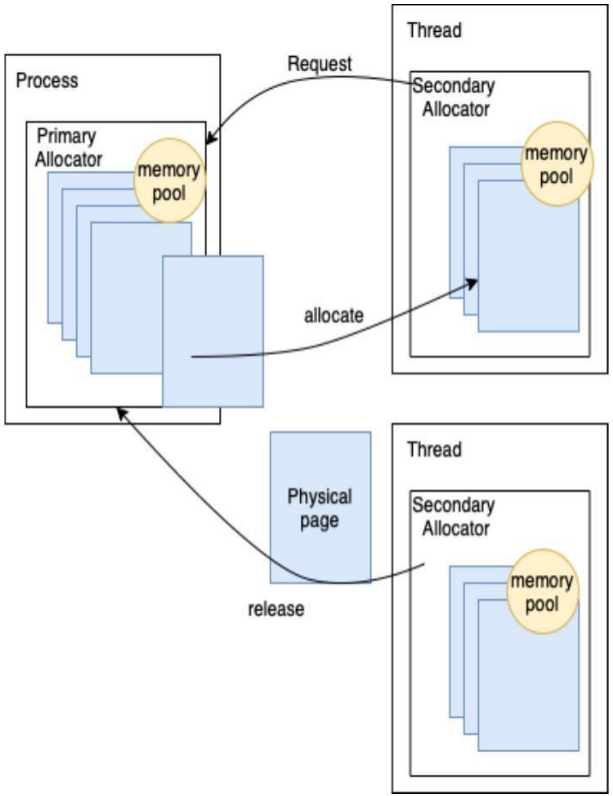
内存分配机制（堆对象、栈对象、全局对象）

堆对象是使用 malloc()、calloc()、realloc() 等函数操作的内存对象。为了管理堆对象，这些分配系统调用被替换为 mmap()、mremap() 和 mprotect()。
栈对象是在函数的栈帧中分配的内存对象。我们区分两种类型的栈对象：隐式分配或显式分配。
全局变量是在二进制文件中的.bss和.data段中分配的内存对象。根据Linux ELF 格式的定义.bss和.data段必须映射为MAP_PRIVATE页面。有一条规则规定，MAP_PRIVATE页面不能与其他页面连续映射到同一个物理内存页面，因为它们不可共享。所以对于全局对象的 mmap()、mremap() 和 mprotect()更改为：
（1）文件支持的映射

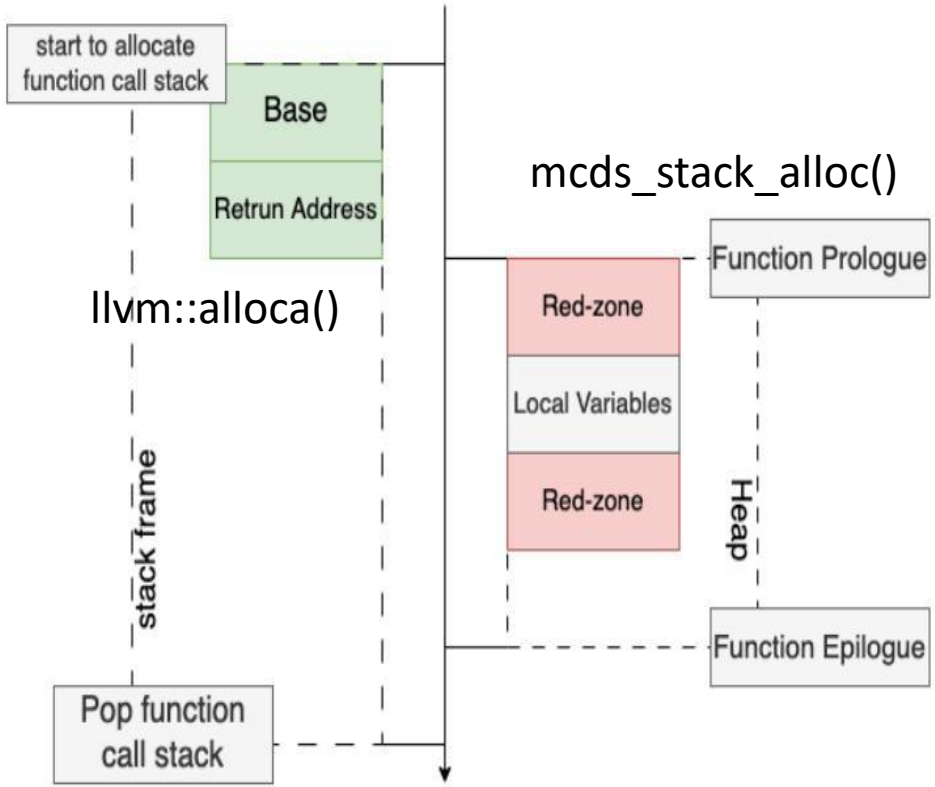
```
mmap(0, 0x1000, PROT_WRITE|PROT_READ,MAP_PRIVATE);
munmap (addr + size, 0x1000 - size);
```

(2) 重叠

```
mremap(old_addr, 0, 0x1000, MREMAP_FIXED_NOREPLACE,old_addr + 0x1000);
munmap(new_addr + size, 0x1000 - size);
```



堆对象分配



堆栈对象的内存布局

基于英特尔 SGX 的影子内存增强

在 SGX 架构中，分配了读写保护标志的虚拟内存页被指定为 TCS 页，而进程允许访问的物理内存空间被称为其安全区。每个进程都有自己的安全区。硬件安全区页缓存映射（EPCM）寄存器本质上是一个表，其中包含每个安全区的条目以及拥有该安全区的进程可以访问的相应 TCS 页面。EPCM 寄存器充当硬件影子内存。与 Asan 使用软件级影子内存记录内存空间验证并使用按位运算验证地址不同，我们跟踪 EPCM 信号作为无效访问的标志。一旦访问违反了 EPCM 寄存器中的记录，就会发出信号，并由信号处理程序捕获。