Toolchain support for objtool

Josh Poimboeuf

Linux Plumbers Conference | Dublin, Ireland Sept. 12-14, 2022

What is objtool?

- Kernel-specific tool
- Reverse engineers control flow graph (CFG) of compiled objects
- Performs various validations and modifications
- Goal: improve robustness/security

What does objtool do?

- Stack unwinding metadata validation / ORC generation
- SMAP validation
- “noinstr” validation
- More...

Why can’t toolchain do it?

- Kernel has a lot of asm / inline-asm
- Manual .cfi annotations suck
- Kernel relies on unwinding correctness
- Toolchain doesn’t know about SMAP, “noinstr” rules

Objtool challenges

- Reverse engineering the control flow graph (CFG) is straightforward, except for:
  - Noreturn function call sites
  - Jump tables (i.e., switch statements)
- Proposal: compiler gives hints to help with "CFG recovery"

GCC/Clang proposal

- **-fannotate-noreturn**
  - Create ELF section referencing all “noreturn” functions or call sites
- **-fannotate-jump-table**
  - Create ELF section describing all jump tables in the file

Upstream discussion

More detailed proposal/discussion:
https://lore.kernel.org/linux-toolchains/

Backup slides

-fannotate-jump-table

- Create **.annotate.jump_table** section:

```
struct annotate_jump_table {
    void  *indirect_jmp;
    long num_targets;
    void *targets[];
};
```

-fannotate-jump-table

```
.Lswitch_jmp:
        // %rax is .Lcase_1 or .Lcase_2
        jmp %rax

.Lcase_1:
        ...
.Lcase_2:
        ...

.pushsection .annotate.jump_table
        // indirect JMP address
        .quad .Lswitch_jmp

        // num jump targets
        .quad 2

        // indirect JMP target addresses
        .quad .Lcase_1
        .quad .Lcase_2
.popsection
```

-fannotate-noreturn

- Create **.annotate.noreturn** section
  - References all noreturn functions (both explicit and implicit)

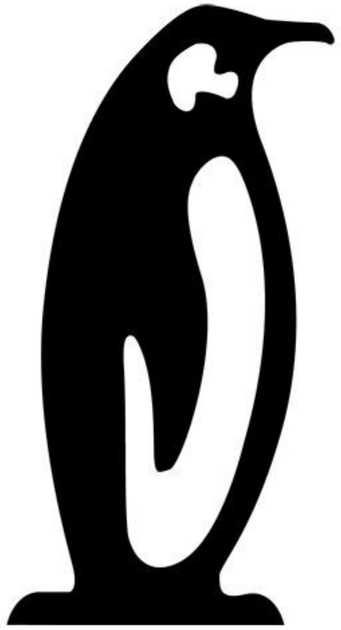
-fannotate-noreturn

```
// explicit noreturn
__attribute__((__noreturn__)) void func1(void)
{
	exit(1);
}

// explicit noreturn (declaration only)
extern __attribute__((__noreturn__)) void func2(void);

// implicit noreturn
static void func3(void)
{
	func2();
}

.pushsection .annotate.noreturn
	.quad func1
	.quad func2
	.quad func3
.popsection
```



Linux Plumbers Conference

Dublin, Ireland September 12-14, 2022