Porting eBPF CO-RE to arm64 Leads to Fix the Kernel

eBPF summit 2022

Francis Laniel
flaniel@linux.microsoft.com

29th September 2022

bpf2go

What it is?

bpf2go enables you to [1]:

> *[compile] a C source file into eBPF bytecode and then emits a Go file containing the eBPF.*

bpf2go

What it is?

`bpf2go` enables you to [1]:

> *[compile] a C source file into eBPF bytecode and then emits a Go file containing the eBPF.*

It permits you to generate eBPF bytecode for several architectures, by calling `clang`, like [2]:

- `amd64`
- `arm64`

Problem and context

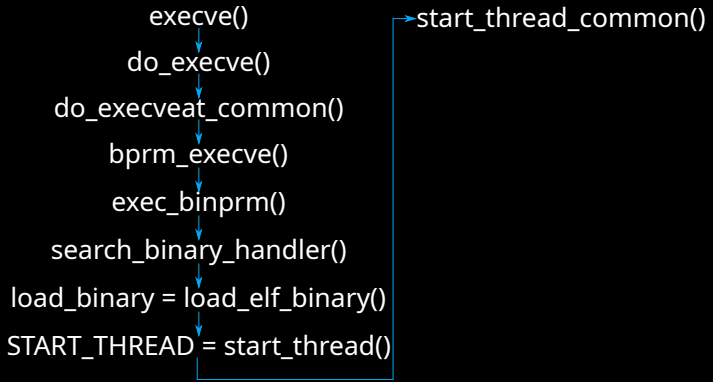
Problem and context

arm

```
# Francis uses execsnoop!
$ ./kubectl-gadget trace exec -A
NODE NAMESPACE POD CONTAINER PID PPID PCOMM RET ARGS
# But nothing happened!
```

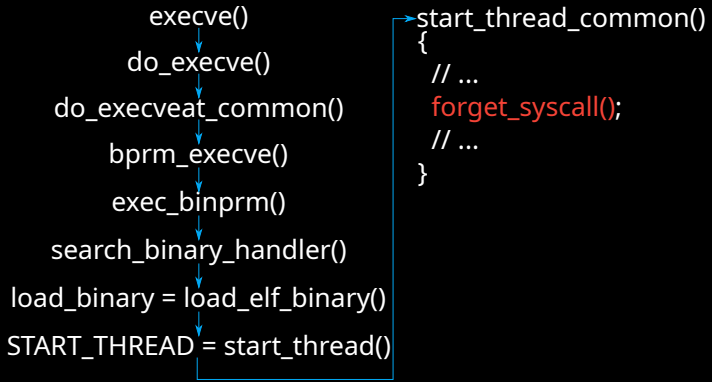
Contribution

Investigate the root cause



Contribution

Investigate the root cause

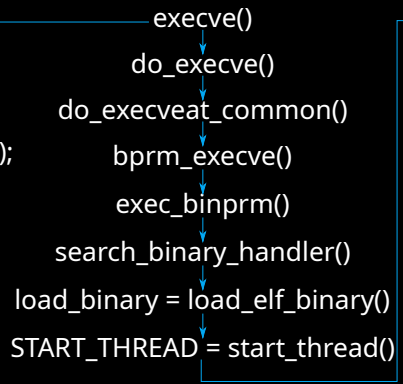


Contribution

Investigate the root cause

```
print_syscall_exit()
{
 // ...
 syscall = trace->nr;
 entry = syscall_nr_to_meta(syscall);

 if (!entry) {
  trace_seq_putc(s, '\n');
  goto out;
 }
 // ...
}
```




```
start_thread_common()
{
 // ...
 forget_syscall();
 // ...
}
```

Contribution

Upstream bug fix and workaround for older kernels

Upstream fix [3]:

```
--- a/arch/arm64/include/asm/processor.h
+++ b/arch/arm64/include/asm/processor.h
@@ -272,8 +272,9 @@ void tls_preserve_current_state(void);

 static inline void start_thread_common(struct pt_regs *regs,
↪  unsigned long pc)
 {
+        s32 previous_syscall = regs->syscallno;
         memset(regs, 0, sizeof(*regs));
-        forget_syscall(regs);
+        regs->syscallno = previous_syscall;
         regs->pc = pc;

         if (system_uses_irq_prio_masking())
```

Contribution

Upstream bug fix and workaround for older kernels

Upstream fix [3]:

```
--- a/arch/arm64/include/asm/processor.h
+++ b/arch/arm64/include/asm/processor.h
@@ -272,8 +272,9 @@ void tls_preserve_current_state(void);

 static inline void start_thread_common(struct pt_regs *regs,
↪  unsigned long pc)
 {
+        s32 previous_syscall = regs->syscallno;
         memset(regs, 0, sizeof(*regs));
-        forget_syscall(regs);
+        regs->syscallno = previous_syscall;
         regs->pc = pc;

         if (system_uses_irq_prio_masking())
```

Use `kprobe` for older kernels [4]:

- ✓ Permits running `execsnoop`
- × Arguments will not be traced

Conclusion and future work

Conclusion:

1. Kernel bug was fixed and backported to stable ones, so you can now trace `execve` syscall family on `arm64` [3, 5].
2. Inspektor Gadget was ported on `arm64` [6].

Conclusion and future work

Conclusion:

1. Kernel bug was fixed and backported to stable ones, so you can now trace `execve` syscall family on `arm64` [3, 5].
2. Inspektor Gadget was ported on `arm64` [6].

Future work:

1. Test Inspektor Gadget `arm64` port on several platforms.

Conclusion and future work

Conclusion:

1. Kernel bug was fixed and backported to stable ones, so you can now trace `execve` syscall family on `arm64` [3, 5].
2. Inspektor Gadget was ported on `arm64` [6].

Future work:

1. Test Inspektor Gadget `arm64` port on several platforms.

Thanks to Jeremi Piotrowski for his help finding this kernel bug!

Bibliography I

[1] cilium contributors, "bpf2go." [Online]. Available: https://github.com/cilium/ebpf/blob/c95b3d3640c9fef72247d33b807f48eaba83e91f/cmd/bpf2go/README.md

[2] ——, "bpf2go architectures." [Online]. Available: https://github.com/cilium/ebpf/blob/c95b3d3640c9fef72247d33b807f48eaba83e91f/cmd/bpf2go/main.go#L44

[3] linux kernel contributors and F. Laniel, "arm64: Do not forget syscall when starting a new thread." Jun. 2022. [Online]. Available: https://git.kernel.org/pub/scm/linux/kernel/git/arm64/linux.git/commit/?id=de6921856f99

[4] inspektor gadget contributors and F. Laniel, "pkg/gadgets: Use kprobe for execsnoop on arm64." Jul. 2022. [Online]. Available: https://github.com/kinvolk/inspektor-gadget/commit/243759db6b19bcef5b7b235eef93206aecc24b66

[5] S. Levin, "[PATCH AUTOSEL 5.10 02/29] arm64: Do not forget syscall when starting a new thread." Aug. 2022. [Online]. Available: https://lore.kernel.org/linux-arm-kernel/20220808013741.316026-2-sashal@kernel.org/

[6] inspektor gadget contributors and F. Laniel, "Add arm64 support for inspektor-gadget," Jul. 2022. [Online]. Available: https://github.com/kinvolk/inspektor-gadget/pull/441