The Container Security in Healthcare Data Exchange System

Bachelor's degree graduation project

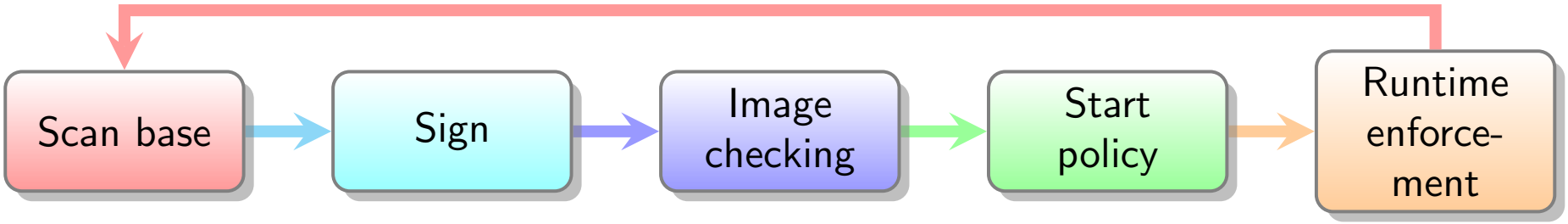
Chih-Hsuan Yang

National Sun Yat-sen University
Advisor: Chun-I Fan

August 6, 2021

Outline

Flow chart


Scan base
Sign
Image checking
Start policy
Runtime enforce-ment

Plans

Plans in this week

- Use the BPF feature and fuzzing technique to collect the "normal" system calls in healthcare data exchange system.
- Keep survey those papers.

I did my best.
But it still have some problems.
Do you want to listen it?

(e)BPF

- (e)BPF is a dynamic tracing technique.
- The bpf() system call first appeared in Linux 3.18.
- The kernel provide a JIT that can execute out Byte Code probe into the kernel dynamically.

cannot attach kprobe, probe entry may not exist

```
→  foo cat ./clone.py && sudo ./clone.py
```

File: ./clone.py

```
1   #!/usr/bin/env python3
2   from bcc import BPF
3
4   # bpf program in restricted C language.
5   prog = """
6   int hello_world(void *ctx) {
7       bpf_trace_printk("Hello, World!\\n");
8       return 0;
9   }
10  """
11
12  b = BPF(text=prog)
13
14  # attaching hello_world function to sys_clone system call.
15  b.attach_kprobe(event="sys_clone", fn_name="hello_world")
16
17  # reading from /sys/kernel/debug/tracing/trace_pipe
18  b.trace_print(fmt="Program:{0} Message:{5}")
19
```

```
cannot attach kprobe, probe entry may not exist
Traceback (most recent call last):
  File "/media/d/foo/./clone.py", line 15, in <module>
    b.attach_kprobe(event="sys_clone", fn_name="hello_world")
  File "/usr/lib/python3/dist-packages/bcc/__init__.py", line 683, in attac
h_kprobe
    raise Exception("Failed to attach BPF program %s to kprobe %s" %
Exception: Failed to attach BPF program b'hello_world' to kprobe b'sys_clon
e'
```

```
→  foo cat /boot/config-$(uname -r) | grep "CONFIG_.*BPF" | sed '/#/d' | grep .
CONFIG_CGROUP_BPF=y
CONFIG_BPF=y
CONFIG_BPF_LSM=y
CONFIG_BPF_SYSCALL=y
CONFIG_ARCH_WANT_DEFAULT_BPF_JIT=y
CONFIG_BPF_JIT_DEFAULT_ON=y
CONFIG_IPV6_SEG6_BPF=y
CONFIG_NETFILTER_XT_MATCH_BPF=m
CONFIG_NET_CLS_BPF=m
CONFIG_NET_ACT_BPF=m
CONFIG_BPF_JIT=y
CONFIG_BPF_STREAM_PARSER=y
CONFIG_LWTUNNEL_BPF=y
CONFIG_HAVE_EBPF_JIT=y
CONFIG_BPF_EVENTS=y
CONFIG_TEST_BPF=m
→  foo cat /boot/config-$(uname -r) | grep "CONFIG_IKHEADERS"
# CONFIG_IKHEADERS is not set
→  foo
```

Recompile the kernel

```
➜  foo cat /boot/config-$(uname -r) | grep -f bpf.conf; uname -a
CONFIG_BPF=y
CONFIG_HAVE_EBPF_JIT=y
CONFIG_BPF_SYSCALL=y
CONFIG_BPF_JIT=y
CONFIG_IKHEADERS=y
CONFIG_NET_CLS_BPF=m
CONFIG_NET_ACT_BPF=m
CONFIG_BPF_EVENTS=y
Linux scc-home 5.13.7 #3 SMP Tue Aug 3 23:13:49 CST 2021 x86_64 GNU/Linux
➜  foo
```

But it still not work, which has same error as previous.

Another front-end wrapper bpftrace

```
➜  foo sudo bpftrace --version
bpftrace v0.13.0-63-g341b
free(): double free detected in tcache 2
[1]    157002 abort      sudo bpftrace --version
➜  foo
```

```
[ 78%] Built target watchpoint
[ 99%] Built target man_man
[100%] Built target adoc_man
[100%] Built target man
Install the project...
-- Install configuration: "Release"
-- Up-to-date: /usr/local/bin/bpftrace
-- Up-to-date: /usr/local/share/bpftrace/tools/bashreadline.bt
```

```
➜  build git:(master) /usr/local/bin/bpftrace --version
bpftrace v0.13.0-63-g341b
free(): double free detected in tcache 2
[1]    162171 abort      /usr/local/bin/bpftrace --version
➜  build git:(master)
```

bpftrace using docker

```
Step 11/11 : ENTRYPOINT ["/bin/sh", "/build.sh"]
 ---> Using cache
 ---> 9fd441beb718
Successfully built 9fd441beb718
Successfully tagged bpftrace-builder-alpine:latest
/media/d/foo/bpftrace
CMake Warning at CMakeLists.txt:46 (message):
  static libc is known to cause problems, consider STATIC_LIBC=OFF.  Procee
d
  at your own risk


-- Please install the libbfd development package (missing: LIBBFD_LIBRARIES
 LIBBFD_INCLUDE_DIRS)
-- Please install the libopcodes development package (missing: LIBOPCODES_L
IBRARIES LIBOPCODES_INCLUDE_DIRS)
-- Found LLVM 9.0.0: /usr/lib/llvm9/lib/cmake/llvm
-- Disabled codegen test for LLVM != 12
CMake Error at /usr/share/cmake/Modules/FindPackageHandleStandardArgs.cmake
:137 (message):
  Could NOT find GTest (missing: GTEST_LIBRARY GTEST_MAIN_LIBRARY)
Call Stack (most recent call first):
  /usr/share/cmake/Modules/FindPackageHandleStandardArgs.cmake:378 (_FPHSA_
FAILURE_MESSAGE)
  /usr/share/cmake/Modules/FindGTest.cmake:197 (FIND_PACKAGE_HANDLE_STANDAR
D_ARGS)
  tests/CMakeLists.txt:107 (find_package)


-- Configuring incomplete, errors occurred!
See also "/media/d/foo/bpftrace/build-release/CMakeFiles/CMakeOutput.log".
See also "/media/d/foo/bpftrace/build-release/CMakeFiles/CMakeError.log".
➜  bpftrace git:(master)
 □ 0  ↑ 22h 55m  1 zsh
```

https://hackmd.io/@25077667/Byvwszd1K#bpftrace-using-docker

The official document says

- `./build-docker-image.sh` - builds just the `bpftrace-builder` Docker image
- `./build-debug.sh` - builds bpftrace with debugging information (requires `./build-docker-image.sh` to have already been run)
- `./build-release.sh` - builds bpftrace in a release configuration (requires `./build-docker-image.sh` to have already been run)

`./build.sh` is equivalent to `./build-docker-image.sh && ./build-release.sh`

```
→  bpftrace git:(master) sudo ./build-debug.sh
CMake Warning at CMakeLists.txt:46 (message):
  static libc is known to cause problems, consider STATIC_LIBC=OFF.
d
  at your own risk


-- Please install the libbfd development package (missing: LIBBFD_LI
 LIBBFD_INCLUDE_DIRS)
-- Please install the libopcodes development package (missing: LIBOP
IBRARIES LIBOPCODES_INCLUDE_DIRS)
-- Found LLVM 9.0.0: /usr/lib/llvm9/lib/cmake/llvm
-- Disabled codegen test for LLVM != 12
CMake Error at /usr/share/cmake/Modules/FindPackageHandleStandardArg
:137 (message):
  Could NOT find GTest (missing: GTEST_LIBRARY GTEST_MAIN_LIBRARY)
Call Stack (most recent call first):
  /usr/share/cmake/Modules/FindPackageHandleStandardArgs.cmake:378 (
FAILURE_MESSAGE)
  /usr/share/cmake/Modules/FindGTest.cmake:197 (FIND_PACKAGE_HANDLE_
D_ARGS)
  tests/CMakeLists.txt:107 (find_package)


-- Configuring incomplete, errors occurred!
See also "/media/d/foo/bpftrace/build-debug/CMakeFiles/CMakeOutput.l
See also "/media/d/foo/bpftrace/build-debug/CMakeFiles/CMakeError.lo
→  bpftrace git:(master)
```

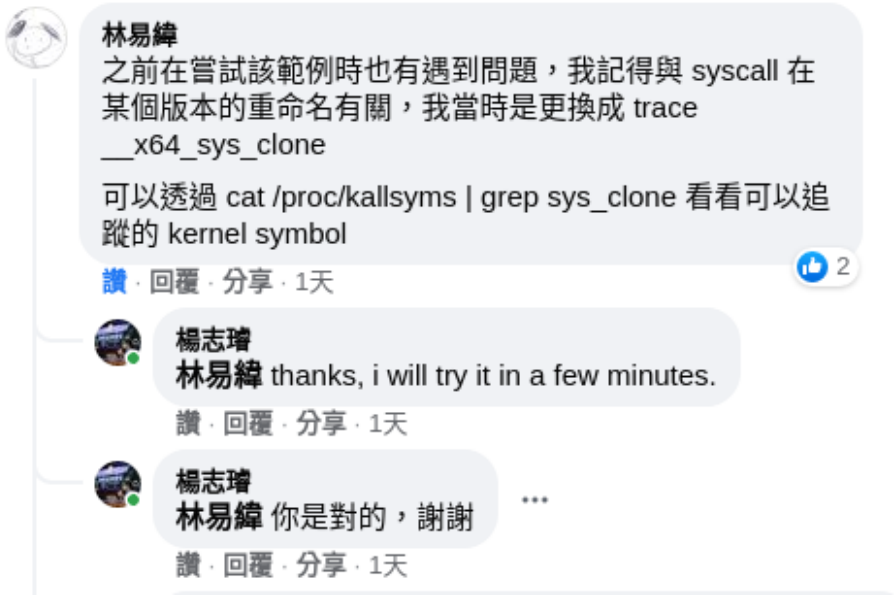
```
→  bpftrace git:(master) ✗ ./b
CMake Error at CMakeLists.txt:
  Dependencies can only be emb

  Enable STATIC_LINKING=ON to


-- Configuring incomplete, err
See also "/tmp/bpftrace/build-
See also "/tmp/bpftrace/build-
→  bpftrace git:(master) ✗
```


static libc is known to cause problems
STATIC_LIBC=OFF
Dependencies can only be embedded for a static build.
STATIC_LIBC=ON
static libc is known to cause problems

Debends on kernel version and distro.



Many debugs. . .

- The kernel what I compiled is 5.13.7.
- Hey, there are no ‘linux-headers-$(uname -a)‘ in debian sid!
- Okay, I can compile the header my my self.
- Hey, the bcc and bpftrace are failed, they cannot get those entry on 5.13.7.
- Okay, I go to fix the bcc and bpftrace source code, and it works on host OS.
- Hey, It does not work in container, failed to compile BPF module. (Next page)
- Okay, I change the front end of eBPF to tracee, which is another open source wrapper.
- It also works on host OS, but failed to load libbpf in container. (Next page)

```
➜  foo sudo docker run --rm -v $(pwd)/data:/app/data bcc_try
/virtual/main.c:29:1: warning: declaration of 'struct tracepoint__raw_syscalls__sys_exit' will not be visible outside of this function [-Wvisibility]
TRACEPOINT_PROBE(raw_syscalls, sys_exit) {
^
/virtual/include/bcc/helpers.h:1263:46: note: expanded from macro 'TRACEPOINT_PROBE'
int tracepoint__##category##__##event(struct tracepoint__##category##__##event *args)
                                             ^
<scratch space>:153:1: note: expanded from here
tracepoint__raw_syscalls__sys_exit
^
/virtual/main.c:50:19: error: incomplete definition of type 'struct tracepoint__raw_syscalls__sys_exit'
    u32 key = args->id;
              ~~~~^
/virtual/main.c:29:1: note: forward declaration of 'struct tracepoint__raw_syscalls__sys_exit'
TRACEPOINT_PROBE(raw_syscalls, sys_exit) {
^
/virtual/include/bcc/helpers.h:1263:46: note: expanded from macro 'TRACEPOINT_PROBE'
int tracepoint__##category##__##event(struct tracepoint__##category##__##event *args)
                                             ^
<scratch space>:153:1: note: expanded from here
tracepoint__raw_syscalls__sys_exit
^
1 warning and 1 error generated.
Traceback (most recent call last):
  File "./syscount.py", line 165, in <module>
    bpf = BPF(text=text)
  File "/usr/lib/python2.7/dist-packages/bcc/__init__.py", line 452, in __init__
    raise Exception("Failed to compile BPF module %s" % (src_file or "<text>"))
Exception: Failed to compile BPF module <text>
➜  foo
 0  1d 6h 26m  1 zsh  2 zsh  3 zsh                                  36°C | 06:01 | 05 Aug  scc  scc-home
```

```
➜  dist sudo docker run --rm -v $(pwd)/data:/app/data
root@838d1a85ff3b:/# app
app
root@838d1a85ff3b:/# app/tracee-ebpf
libbpf: failed to open system Kconfig
libbpf: failed to load object 'embedded-core'
2021/08/05 17:46:45 error creating Tracee: failed to l
root@838d1a85ff3b:/#
```

There also have some distro. dependency issue

- Alpine
- Debian stretch, buster, bulleyes
- Ubuntu Bionic, Focal Fossa
- Linux kernel version, 4.18.0, 5.2.0. . .

```
328  #if LINUX_VERSION_CODE < KERNEL_VERSION(4, 18, 0)
```

version.h /usr/include/linux - 宣告 (1)

```
1  #define LINUX_VERSION_CODE 330286
2  #define KERNEL_VERSION(a,b,c) (((a) << 16) + ((b) << 8) + ((c) > 255 ? 255 : (c)))
3
```

```
34  # DOCKER_BUILDER_KERN_SRC_MNT is the kernel headers directory to mount into the docker builder container. DOCKER_BUILDER_KERN_SRC sh
35  DOCKER_BUILDER_KERN_SRC_MNT ?= $(dir $(DOCKER_BUILDER_KERN_SRC))
36  LINUX_VERSION_CODE := $(shell uname -r | gawk '{split($$0,a,"."); split(a[3], b, "-"); print lshift(a[1], 16) + lshift(a[2],8) + b[1
37
```