哈爾濱工業大學（深圳）
HARBIN INSTITUTE OF TECHNOLOGY, SHENZHEN

面向云原生的内核威胁检测系统设计与实现

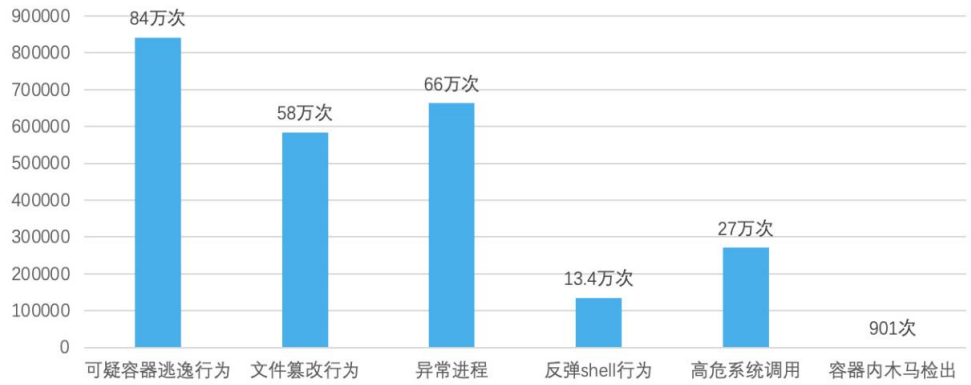
答辩人：丁浩卓　　指导老师：刘川意

哈爾濱工業大學（深圳）
HARBIN INSTITUTE OF TECHNOLOGY, SHENZHEN

面向云原生的内核威胁检测系统设计与实现

选题背景

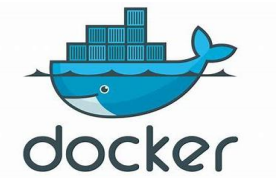
云原生虽然具备着更大的灵活性、业务敏捷性和强扩展性，但同时也遭遇了巨大的安全风险。针对云原生的在野攻击数量巨大，容器成为重要的攻击环境。



2021年，腾讯云容器安全服务监测到的可疑容器逃逸行为84万次

参考来源:
[1] 腾讯云计算（北京）有限公司. 腾讯云容器安全白皮书. 2022.

选题意义



提升监控容器行为的有效性，监控容器生命周期各阶段行为。

提升Rootkit病毒检测的有效性，研究Rootkit病毒行为特征。

应用eBPF技术，提高入侵检测系统的性能 。

主要研究内容

威胁分析

- Docker容器逃逸
- Rootkit：内核态Rootkit、用户态Rootkit

系统设计与实现

- 基于eBPF的内核实时监控模块
  - eBPF学习研究，实现内核运行时信息收集处理
- 威胁规则匹配模块
  - 分析容器逃逸&Rootkit，形成针对入侵攻击的规则库，实现规则匹配引擎

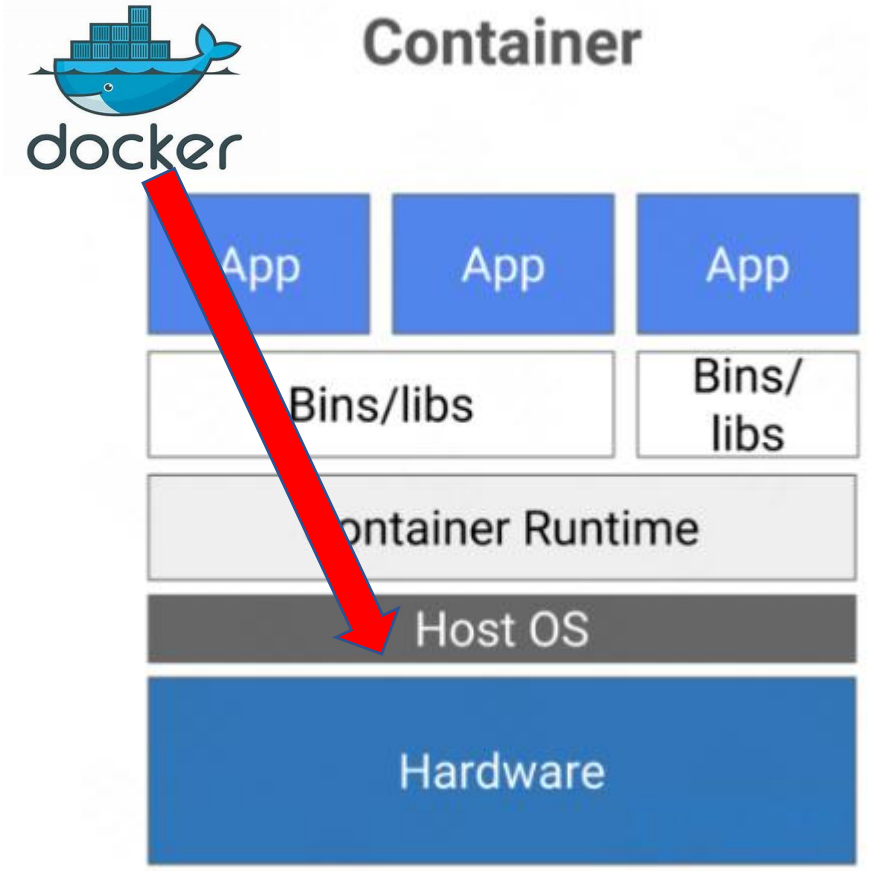
功能测试和性能评估

- 基于功能测试完善功能
  - 容器行为监控测试、Rootkit入侵检测测试
- 进行性能评估并优化性能

容器逃逸

容器生命周期中三个过程的安全：构建时安全、部署时安全、运行时安全。
容器运行时相比于其他两个阶段更直接、也更容易分析出环境中的恶意行为。
根据风险所在层次的不同，容器运行时安全风险可以进一步展开为：

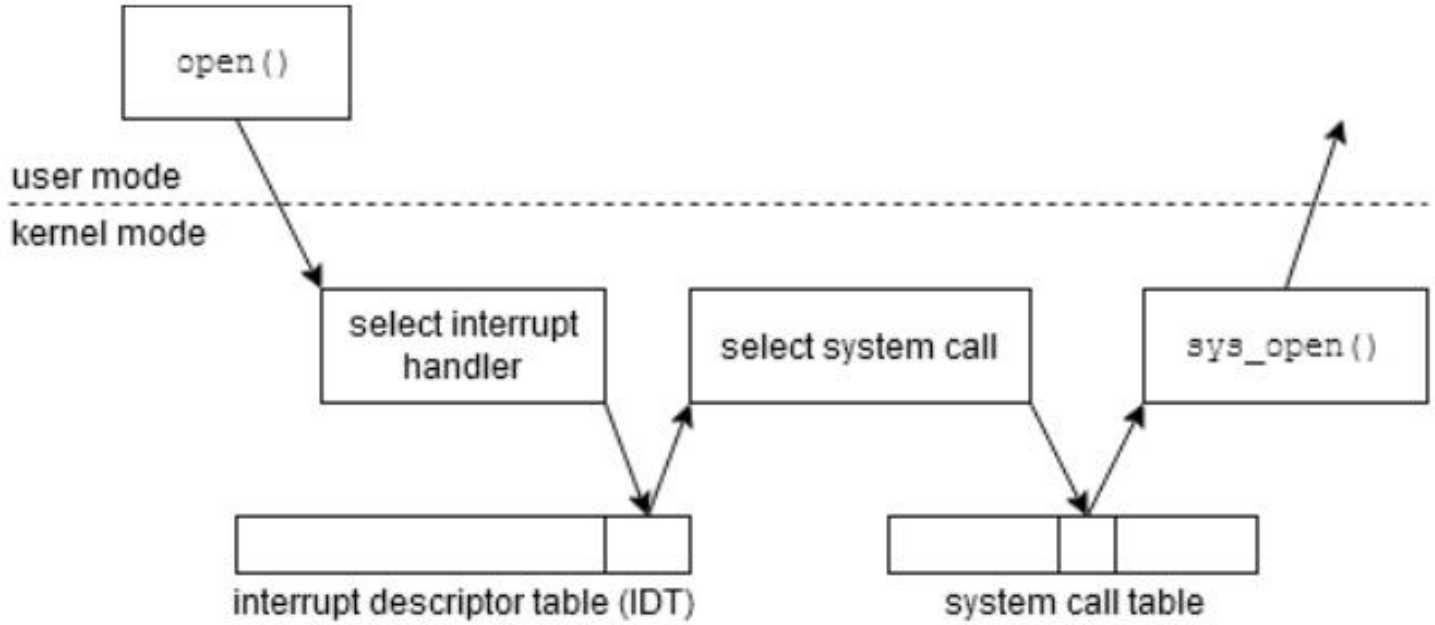
- 危险配置导致的容器安全风险
- 危险挂载导致的容器安全风险
- 相关程序漏洞导致的容器安全风险
- 内核漏洞导致的容器安全风险



容器之间共享操作系统内核，并未实现完全的隔离。若虚拟化软件存在缺陷，或宿主机内核被攻击，将会造诸多的安全问题，包括隔离资源失效、容器逃逸等，影响宿主机上的其他容器甚至整个内网环境的安全

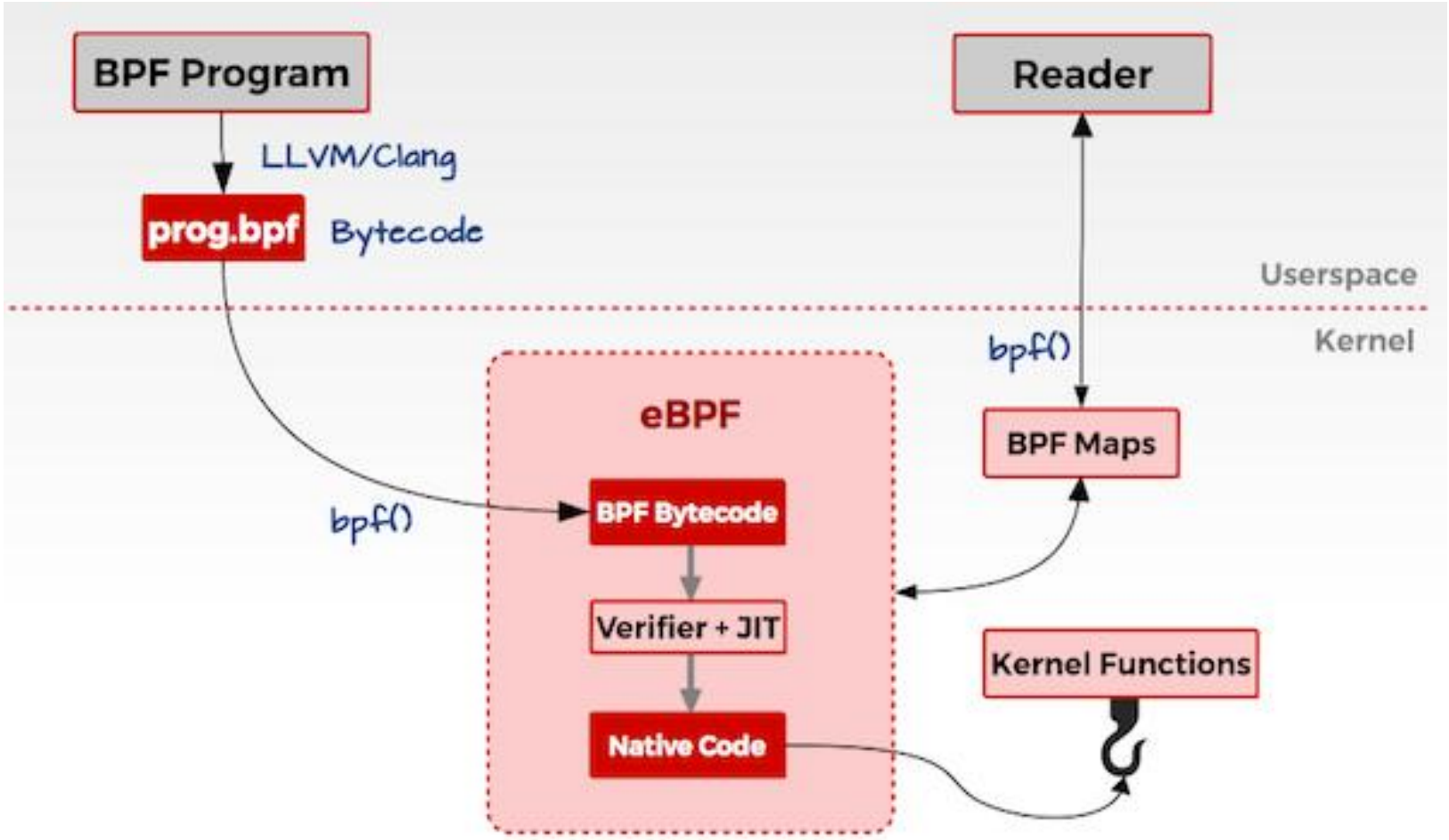
Rootkit攻击

Rootkit的本质是劫持函数，包含对于syscalls或是一般函数的劫持
通过修改返回值或返回的缓冲区内容（数据结构）
可实现：隐藏文件、目录；提权；隐藏进程；提供后门等



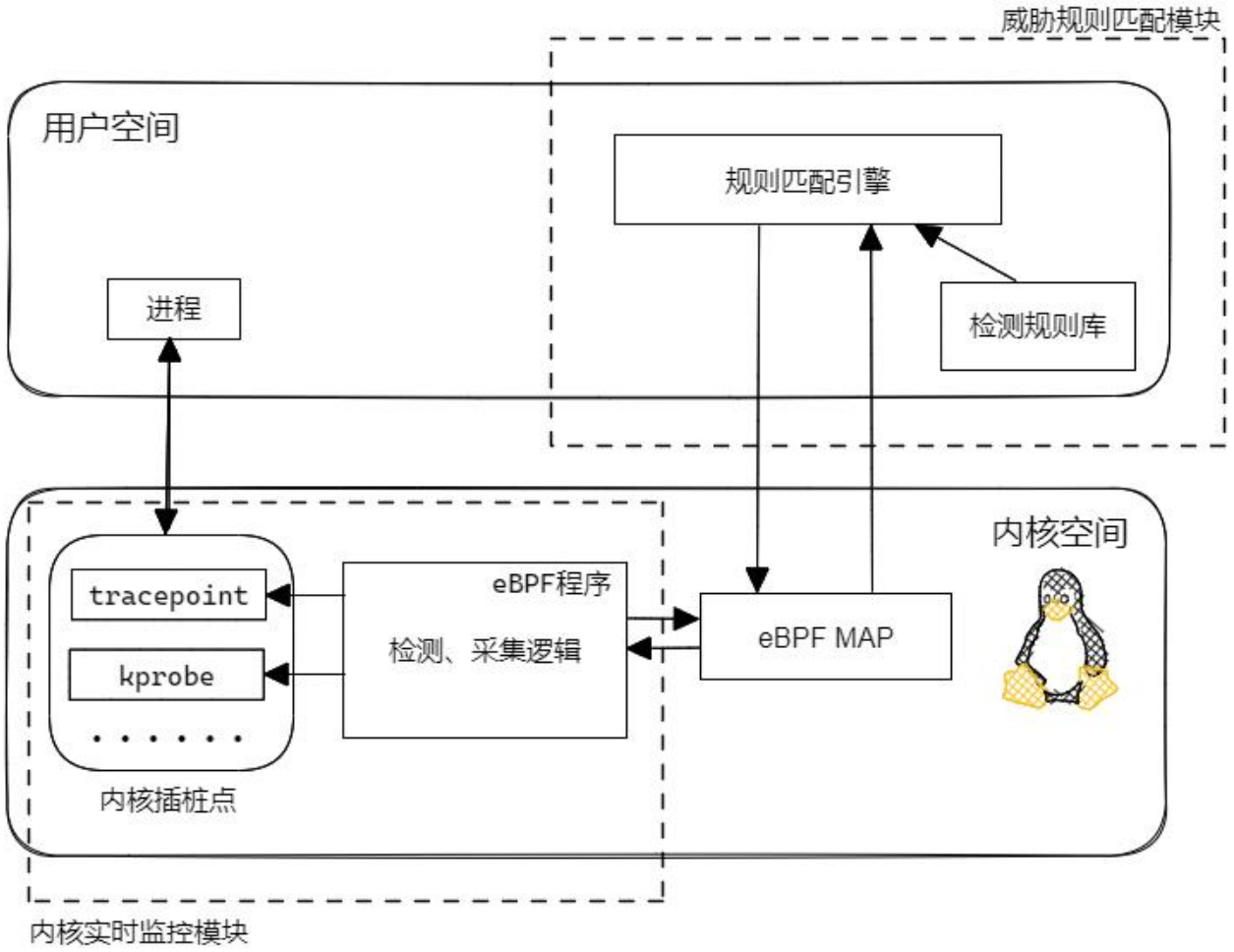
一次正常的系统调用陷入内核并返回的过程

eBPF技术



bpf是运行在内核中的一个虚拟机，支持内核编程实现对收集的数据进行处理

系统设计与实现



内核实时监控模块通过eBPF在关键内核函数进行插桩，实现对内核运行时的细粒度监控。

威胁规则匹配模块将系统运行时上下文信息与规则库进行规则匹配，匹配成功则发出警告。

系统通过eBPF Map进行数据交互。

内核实时监控模块设计与实现

内核实时监控模块由eBPF程序实现，本系统主要基于libbpf开发eBPF程序。主要插桩（hook）的函数如表2-1所示，主要有系统调用、内核函数两大类。通过对关键内核函数调用的监控，实现了内核运行时的细粒度信息收集，提供了全面丰富的运行时信息。

表 2-1 内核插桩点列表

插桩点类型	插桩点名称
tracepoint	module_load
tracepoint	sys_exit_finit_module
tracepoint	sys_enter_mount
tracepoint	sys_exit_mount
tracepoint	sys_enter_open
tracepoint	sys_exit_open
tracepoint	sys_enter_openat
tracepoint	sys_exit_openat
tracepoint	sys_enter_execve
tracepoint	sys_enter_execveat
tracepoint	sys_enter_kill
kretprobe	kprobe_lookup_name
kretprobe	arm_kprobe
kretprobe	insn_init
kretprobe	insn_get_length

功能测试 - 检测危险配置容器启动

开启HIDS后成功检测到高权限容器启动

```
dhz@ubuntu:~/workspace/HIDS-eBPF/hids$ sudo ./hids
=====================================================
TIME     EVENT              COMM         PID      PPID     PID_NS       DESCRIBE
04:18:52 MODULE_LOAD          modprobe             6066    5952    4026531836  load module, module-name is veth !
04:18:52 INSERT_MODULE_FINISH modprobe             6066    5952    4026531836  insert module finished, module-name is veth !
04:18:57 [Container start]    runc:[2:INIT]        6132    6104    4026532633  container-id: 0d85acbc0ef6, cap_effective:3fffffffff , The privileged container start
container is: 0d85acbc0ef693d94cafcb965b32bc3c14e423e2276935580d4d261c951ef23d
```

```
dhz@ubuntu:~$ sudo docker run -it --privileged ubuntu:20.04
[sudo] password for dhz:
root@0d85acbc0ef6:/#
```

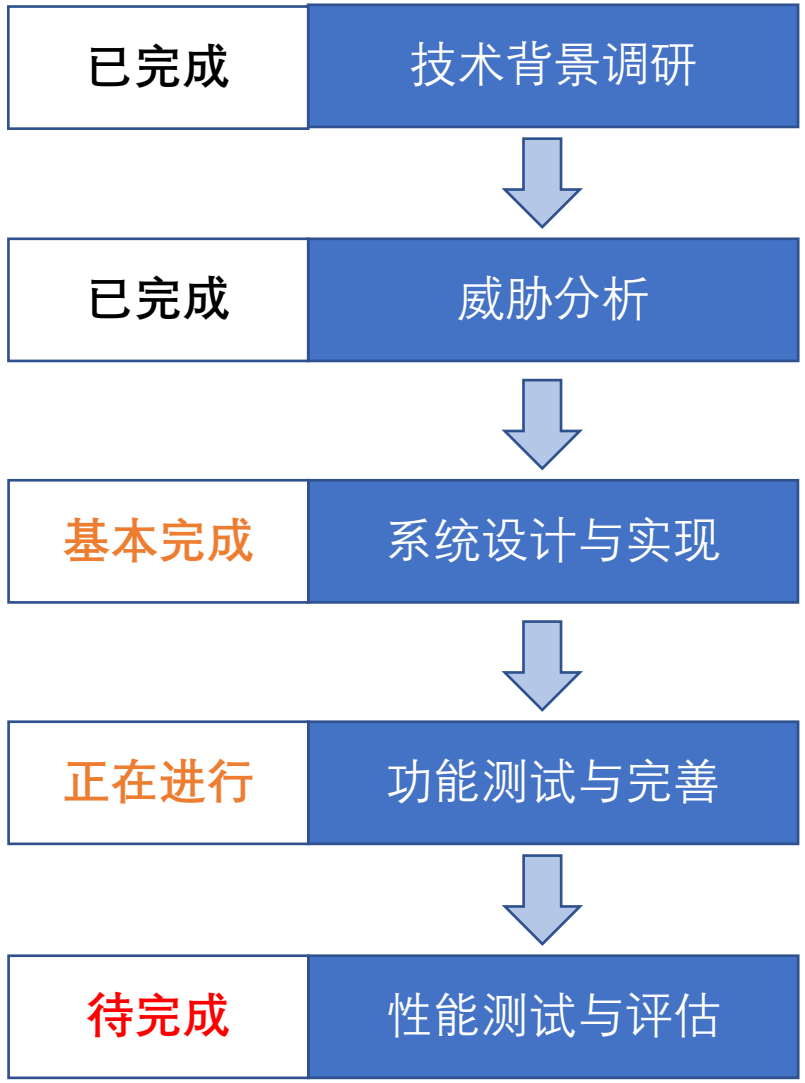
功能测试 - 检测Rootkit加载

开启HIDS后成功检测到Rootkit加载入内核

```
dhz@ubuntu:~/workspace/HIDS-eBPF/hids$ sudo ./hids
=============================================
TIME     EVENT               COMM         PID       PPID      PID_NS       DESCRIBE
06:04:46 MODULE_LOAD          insmod               154436  154435  4026531836  load module, module-name is diamorphine !
06:04:46 KHOOK                insmod               154436  154435  4026531836  using Kernel instruction operation function!
06:04:46 KPROBE               insmod               154436  154435  4026531836  using Kernel KPROBE framework!
06:04:46 SYSCALL_TABLE_HOOK   insmod               154436  154435  4026531836 syscall[62]: be changed. May have been attacked by kernel rootkit !
06:04:46 SYSCALL_TABLE_HOOK   insmod               154436  154435  4026531836 syscall[78]: be changed. May have been attacked by kernel rootkit !
06:04:46 SYSCALL_TABLE_HOOK   insmod               154436  154435  4026531836 syscall[217]: be changed. May have been attacked by kernel rootkit !
06:04:46 INSERT_MODULE_FINISH insmod               154436  154435  4026531836  insert module finished, module-name is diamorphine !
Discover LKM-RootKits!!!  rootkit name is diamorphine !
```

Rootkit加载命令为： insmod diamorphine.ko

后期工作



总结

请各位老师指正

目前毕业设计进度正常，与开题时预期的进度安排基本相符，预计能够按期完成，并取得预期的研究成果。

请各位老师指正