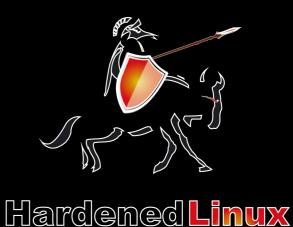
# PaX 安全特性分析

Kaipeng Zeng kaipeng94@gmail.com

### # 个人简介

- \* 曾凯鹏
- \* Day job: TYA infotech



- \* Night job : HardenedLinux contributor
- \* Focus on : Kernel security, testing etc.

### # index

- \* PaX 简介
- \*从Bug 到漏洞到 PaX 的安全特性
- \* PaX 安全特性分析
- \* Linux 社区与安全
- \* HardenedLinux 社区所做的一些工作

## # PaX 简介

- \* PaX/Grsecurity
  - 一组针对内核的安全补丁,盘踞在内核各个角落,不是可加载安全模块一世界上顶尖的安全团队
- \* PaX: Grsecurity 的一部分
  - 专注于 memory corruption
  - 一种 mitigation 的技术

## # Bug --> 漏洞 --> PaX

\* Bug

编码问题, bug, 有可能对应着一个漏洞

\* Bug class

一类可被利用或者有安全隐患的 bug, 有共同特征, 一类漏洞

\* PaX 特性

Kill bug class, kill exploit vector, mitigation

不是漏洞修复补丁,能够解决一类漏洞

## # Code inject & no-execute

- \* Code inject
  shellcode, memory with W^X flag
- \* PaX 的相关特性

PAX\_NOEXEC, PAX\_PAGEEXEC, PAX\_SEGEXEC:

PAX\_PAGEEXEC: page\_fault, no-executable page

PAX\_MPROTECT: keep memory healthy, W^X 不共存

## # Address base exploit & ASLR

#### \* 基于地址的漏洞利用

Code reuse: ret2libc, offset2lib, 取得目标代码地址

#### \* ASLR

用户空间 , 地址空间随机化

PAX\_RANDUSTACK: 随机化 stack base address

PAX\_RANDEXEC: 随机化 main executable address

PAX\_RANDMMAP: 随机化 mmap base address

PaX mmap gap:插入内存空隙, gap, stack clash

\* 针对随机化的爆破: 性能与安全的权衡

### # Reuse attack & RAP

\* 代码复用攻击

**ROP**: return-orirnted programming

\* RAP

**CFI**: Control Flow Integrity

PAX RAP:

Build cfg has added type information by a hashing function

Encrypts the address being returned to

The key used to encrypt the return address is stored in a reserved CPU register

### # ret2usr/ret2dir & SMEP/SMAP &PXN/PAN

\* ret2usr/ret2dir

内核指针指向用户空间的代码

\* PAX\_KERNEXEC / PAX\_UDEREF

Kernel page RO&NX, Userland/kernel separation

\* SMEP/SMAP

特权级内存的执行和访问, x86 硬件支持

\* PXN/PAN

特权级内存的执行和访问, ARM v7 硬件支持

## #针对内核栈的特性

- \* 内核栈溢出 stack smash, usercopy, stack leak
- \* PaX 针对内核栈的加固

PAX\_RANDKSTACK:系统调用时内核栈的随机化

PAX MEMORY KSTACKLEAK: 检查栈溢出

PAX\_USERCOPY:检查目的缓冲区大小,copy\_to/from\_user

\* PaX/Grsecurity: KSATCKOVERFLOW:分离 thread\_info, 栈之间的Guard page

## # 内存泄漏 & infoleak & etc.

\* 内存泄漏

memory sanitize ,内核调用栈擦除

\* infoleak

pfn 进程内存布局 内核日志 等敏感信息的读取限制, 防侧信道攻击

\* PAX\_REFCOUNT:防止内核引用计数的溢出,避免 use after free

## # Linux 社区与安全

\* Linux 社区对安全的态度

"Bug is a bug", slient fix

#### \* KSPP

Kernel Self Protection Project

Feature from PaX/Grsecurity and?

Security feature introduction & exploit verctor introduction

Move to debug option

### # HardenedLinux 社区所做的工作

\* git 仓库

https://github.com/hardenedlinux/grsecurity-101-tutorials

### \* 内容

PaX/Grsecurity 配置使用的文档

Linux kernel mitigation checklist

Linux kernel vulnerablity & exploitation & silent fixes

PaX/Grsecurity 相关特性分析

### # reference

#### \* PaX/Grsecurity:

https://pax.grsecurity.net/

https://grsecurity.net/

#### \* HardenedLinux:

https://github.com/hardenedlinux/grsecurity-101-tutorials

#### \* Kernel mitigation checklist:

https://github.com/hardenedlinux/grsecurity-101-tutorials/blob/master/kernel\_mitigation.md

#### \* Silent fix:

https://github.com/hardenedlinux/grsecurity-101-tutorials/blob/master/kernel\_vuln\_exp.md#silent-fixes-from-linux-kernel-community--welcome-to-add-more-for-fun

#### \* A travel through exploit mitigations in GNU toolchains:

https://github.com/fanfuqiang/doc/blob/master/sacc-osdt-2016-zet.pdf

# # Q&A



Hardened Linux