# Summary

AssassinQ

# Outline

- Linux 基本命令
- 基础汇编指令
- 计算机网络基础知识
- 密码学（CRYPTO）基础知识
- PWN入门之栈：简单栈溢出/Shellcode
- PWN入门之堆：堆结构简介

# Linux 基础命令

- 基础命令（必须掌握）： ls/cd/mkdir/rm/cp/mv/pwd…
- 进阶命令（很有用）： grep/ln/ssh/vim/less/find…
- 高级命令（要用的好很难）： wc/awk/[输入/输出重定向]…

# 基础汇编指令

- 预处理 -> 编译 -> 汇编 -> 链接
- 寄存器（以x86为例）：EAX/EBX/ECX/EDX、ESI/EDI、ESP/EBP、EIP、EFLAGS…
- 汇编指令：MOV/LEA、ADD/SUB(CMP)/MUL/DIV、AND(TEST)/OR/XOR、PUSH/POP、JMP/CALL/RET…

# 基础汇编指令

- Linux下用得到的相关命令：
  - nasm：编译汇编代码
  - objdump：反汇编程序
  - gdb：调试程序

# 基础汇编指令

| Intel | AT&T |
| --- | --- |
| mov eax, 1 | movl $1, %eax |
| mov eax, [ebx + 3] | movl 3(%ebx), %eax |
| mov eax, [ebx + ecx * 2h] | addl (%ebx, %ecx, 0x2), %eax |

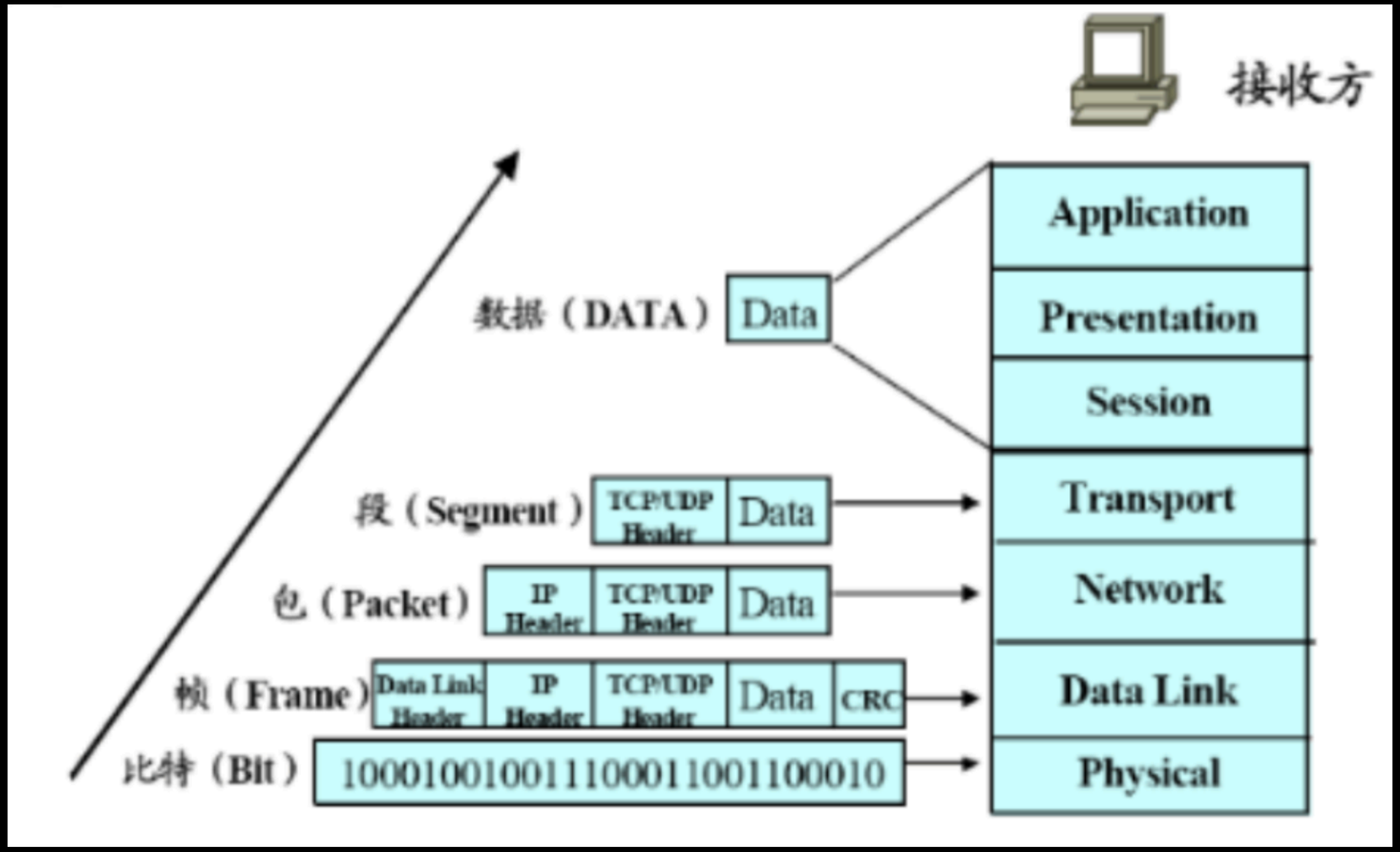
# 基础汇编指令

- objdump -M intel -d ./main
  - alias objdump="objdump -M intel"
- .gdbinit: set disassembly-flavor intel

# 计算机网络基础知识

- OSI 七层模型 -> TCP/IP 五层模型
- 物理层：网线（分组交换）
- 数据链路层：交换机、MAC地址（PPP、CSMA/CD…）
- 网络层：路由器、IP地址（IP、ICMP…）
- 传输层：端口（TCP、UDP…）
- 应用层：HTTP、HTTPS、FTP…

# 计算机网络基础知识


接收方
Application
Presentation
Session
Transport
Network
Data Link
Physical
数据（DATA）
Data
段（Segment）
TCP/UDP Header
Data
包（Packet）
IP Header
TCP/UDP Header
Data
帧（Frame）
Data Link Header
IP Header
TCP/UDP Header
Data
CRC
比特（Bit）
100010010011100011001100010

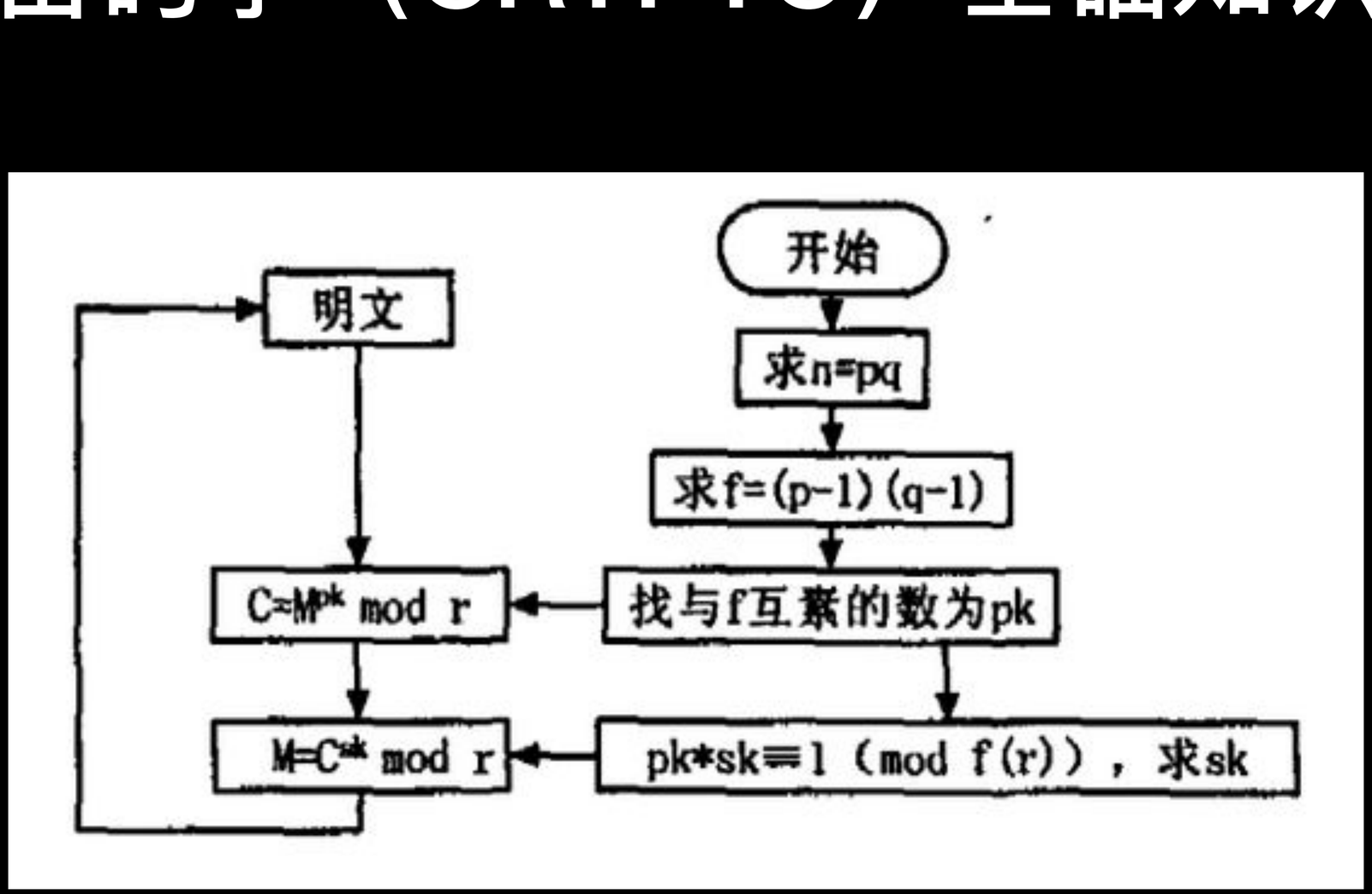
# 计算机网络基础知识

- 局域网（Local Area Network，LAN）：在某一区域内由多台计算机互联成的计算机组
- DHCP（动态主机设置协议）
- IP分类：A类、B类、C类、D类、E类

# 密码学（CRYPTO）基础知识

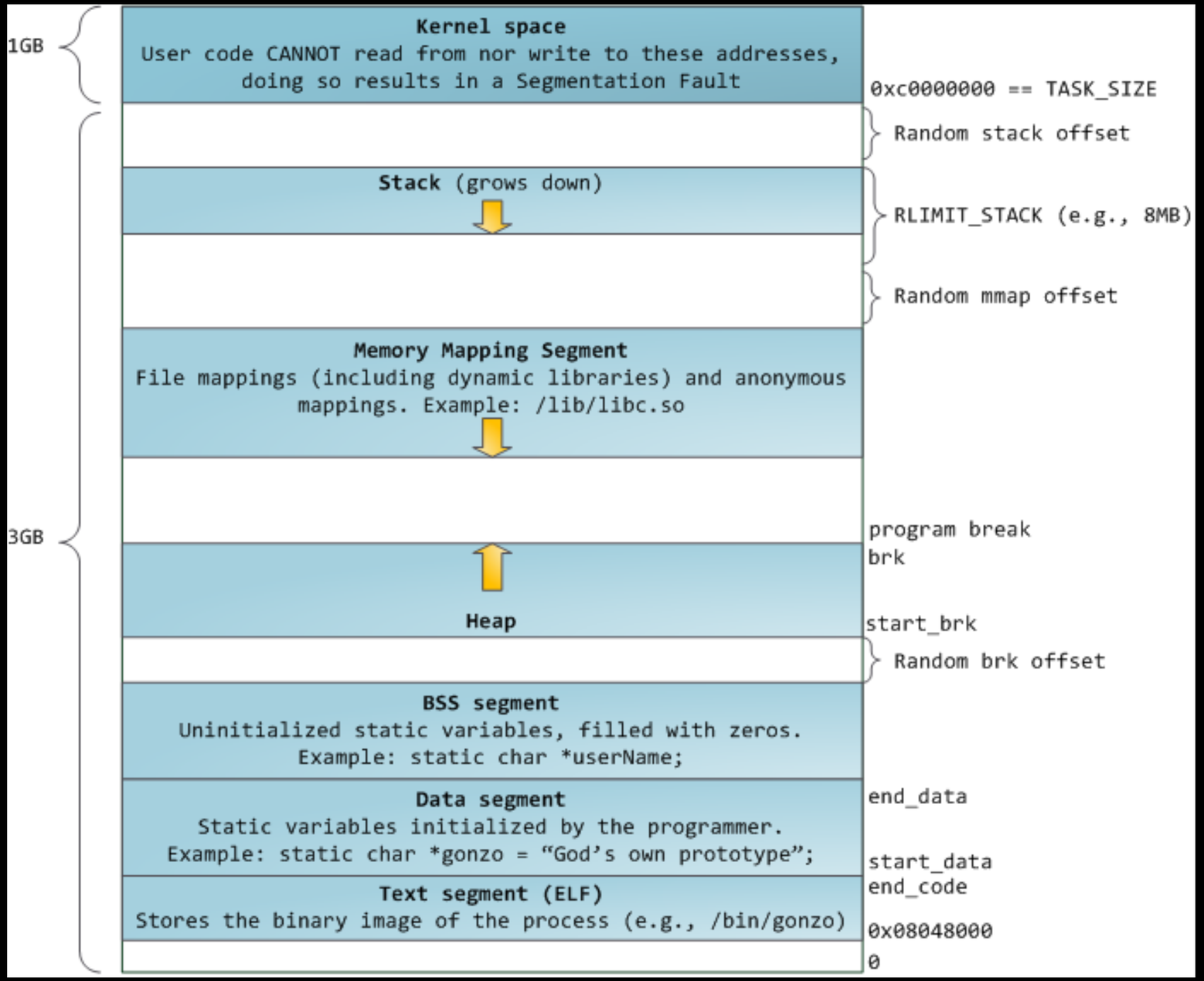
- 密码学
  - 古典密码：置换密码、代换密码
  - 现代密码
    - 对称密码：序列密码、分组密码
    - 非对称密码：公钥密码（RSA）

# 密码学（CRYPTO）基础知识



# 简单栈溢出/Shellcode

- ROP（Return-oriented Programming）
- 关键函数：gets()
  - 因为对读入的字符串长度没有限制，可以覆盖掉函数的返回地址


1GB
3GB
Kernel space
User code CANNOT read from nor write to these addresses, doing so results in a Segmentation Fault
0xc0000000 == TASK_SIZE
Random stack offset
Stack (grows down)
RLIMIT_STACK (e.g., 8MB)
Random mmap offset
Memory Mapping Segment
File mappings (including dynamic libraries) and anonymous mappings. Example: /lib/libc.so
program break
brk
Heap
start_brk
Random brk offset
BSS segment
Uninitialized static variables, filled with zeros.
Example: static char *userName;
Data segment
Static variables initialized by the programmer.
Example: static char *gonzo = "God's own prototype";
end_data
start_data
Text segment (ELF)
Stores the binary image of the process (e.g., /bin/gonzo)
end_code
0x08048000
0

# 简单栈溢出/Shellcode

- 一段可以执行的数据
- 系统调用

# 简单栈溢出/Shellcode

- 介绍一个神器：IDA（Interactive Disassembler Professional）
- 把二进制程序直接反编译成C语言代码

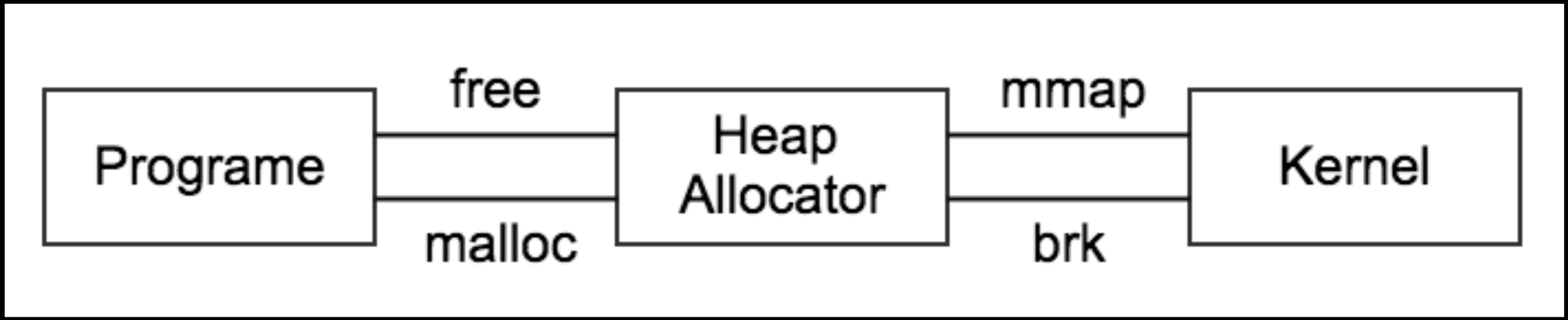
# 堆结构简介

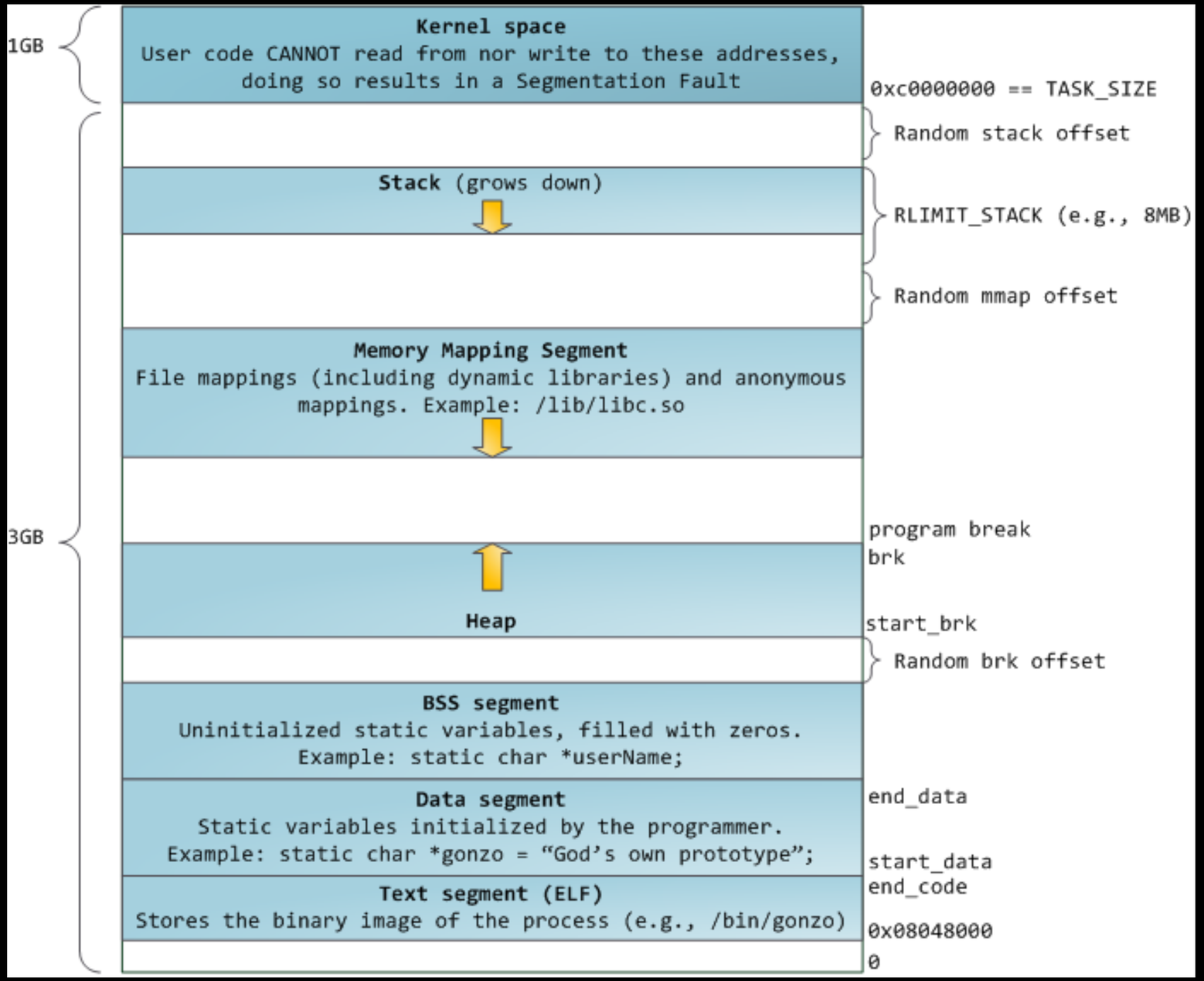
- Heap allocator
- How malloc() & free() work
- Heap structure
- Vulnerabilities

# Heap allocator

- dlmalloc – General purpose allocator
- **ptmalloc2 – glibc**
- jemalloc – FreeBSD and Firefox
- tcmalloc – Google
- libumem – Solaris

# Heap allocator




1GB
3GB
Kernel space
User code CANNOT read from nor write to these addresses, doing so results in a Segmentation Fault
0xc0000000 == TASK_SIZE
Random stack offset
Stack (grows down)
RLIMIT_STACK (e.g., 8MB)
Random mmap offset
Memory Mapping Segment
File mappings (including dynamic libraries) and anonymous mappings. Example: /lib/libc.so
program break
brk
Heap
start_brk
Random brk offset
BSS segment
Uninitialized static variables, filled with zeros.
Example: static char *userName;
Data segment
Static variables initialized by the programmer.
Example: static char *gonzo = "God's own prototype";
end_data
start_data
Text segment (ELF)
Stores the binary image of the process (e.g., /bin/gonzo)
end_code
0x08048000
0

# How malloc() & free() work

- glibc/malloc/malloc.c
- (top) chunk/bin(linked list)/main_arena
- fast bins/unsorted bin/small bins/large bins

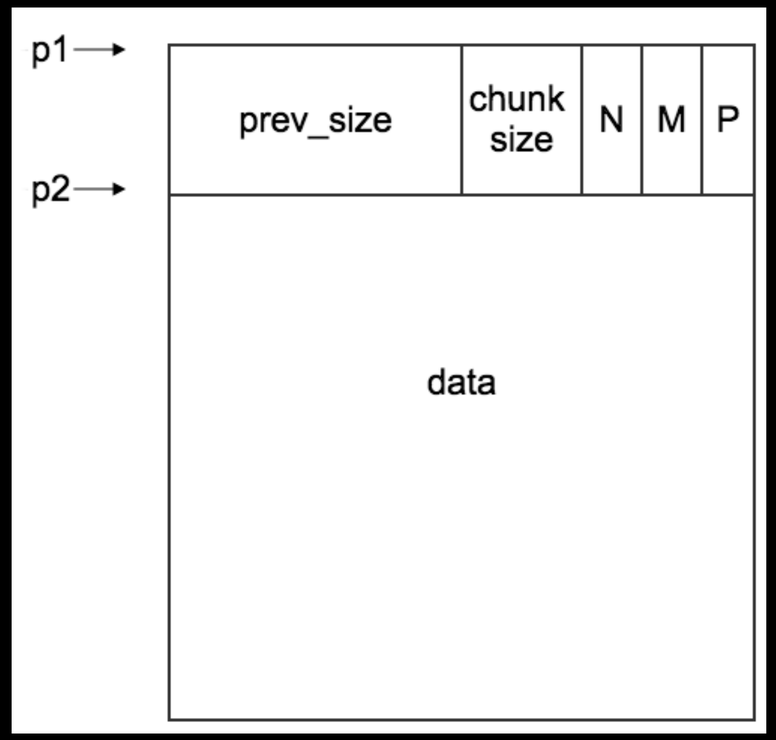
# How malloc() & free() work

- p = malloc(size)
  - find a chunk from bin or take one from top chunk
  - return chunk+16
- free(p)
  - check prev chunk -> inuse?
  - no -> combine -> bins

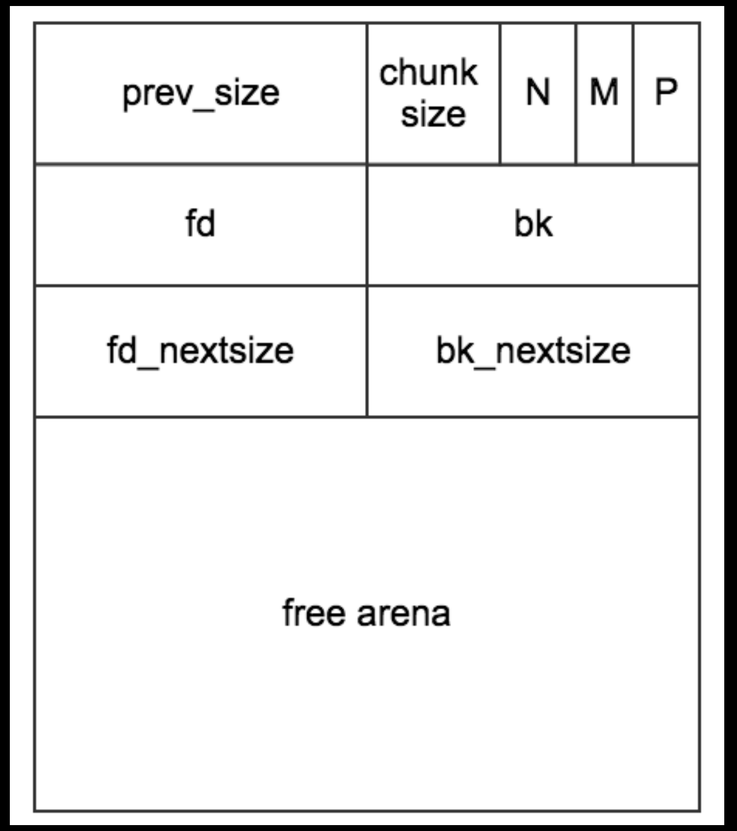
# Heap structure

- next chunk: chunk + size
- prev chunk: chunk - prevsize
- fd/bk: bin (double linked list)
- N: NON_MAIN_ARENA bit
- M: IS_MMAPED bit
- P: PREV_INUSE bit

# Allocated Chunk




# Free Chunk



# Vulnerabilities

- Heap Overflow
- Double Free
- Use After Free
- House of xxx

# Others

- 打CTF
- 参加各种竞赛
- 做项目
- 好好学习准备考研

# Thanks