ROP

面向返回编程（英语：Return-Oriented Programming，缩写：ROP）是计算机安全漏洞利用技术，该技术允许攻击者在安全防御的情况下执行代码，如不可执行的内存和代码签名。攻击者控制堆栈调用以劫持程序控制流并执行针对性的机器语言指令序列（称为Gadgets）。 每一段gadget通常结束于return指令，并位于共享库代码中的子程序。系列调用这些代码，攻击者可以在拥有更简单攻击防范的程序内执行任意操作。

相关资料：https://bbs.pediy.com/thread-223798.htm

ROP突破DEP

DEP：不能在没有执行权限的内存中执行代码

两种突破DEP的思路：

1. 修改内存属性`VirtualProtect`或者申请新的内存`VirtualAlloc`
2. ROP链

相关工具

Immunity Debugger：基于OD，提供了丰富的漏洞的插件，例如mona

- 将`mona.py`放入`Immunity Debugger\PyCommands`目录下

Metasploit：渗透框架

- `msfconsole`终端
- `msfvenom`生成shellcode

mona

在Immunity Debugger中使用mona脚本

1. 寻找溢出点
    - `!mona pattern_create 0x1000`创建4096个字节的环数据
    - `!mona pattern_offset eip`确定溢出点距离缓冲区的大小（需要执行一次返回，使得EIP的值为环数据的一段）
    - 原理：字符串比较
2. 寻找跳板

- 在所有模块中寻找：`!mona jmp -r esp -m *`

3. 自动构造ROP
    - 在所有模块中寻找，且过滤空字符`!mona rop -m * -cpb '\x00'`

Metasploit

生成shellcode

`msfvenom -a x86 --platform Windows -p windows/messagebox -e x86/shikata_ga_nai -b '\x00' -i 3 -f python`

- `-a`：目标架构
- `--platform`：平台
- `-p`：选择payload攻击载荷
- `-e`：选择encoder编码器
- `-b`：过滤硬编码中的指定字节值
- `-i`：迭代次数
- `-f`：输出格式