1. windows 平台的各种典型调试工具及应用的场景；

Run3（应用调试）：OllyDbg（windows），gdb（linux），IDA（windows）

Run0（内核调试）：WinDbg（windows），kgdb（linux）

静态分析工具：IDA and plungins（插件）

动态调试工具：OllyDbg，WinDbg，gdb，kgdb

2. 各种典型的查壳工具名称及对应的用途；

windows 平台查壳：PEID/LoadPE，IDA（手动分析）

linux 平台查壳：DIE，IDA（手动分析）

通用平台查壳：DIE，IDA（手动分析）

查壳工具的用途：二进制查壳，识别加密壳 zprotect、识别压缩壳 UPX 等。

3. 虚拟化环境的搭建过程中涉及的步骤、原理、工具名称；

步骤：增加环境变量，修改 Boot.ini，下载符号表（symbols）并添加路径，选择 COM 对接，配置串口和管道，调整 cmd。（详见图解）

原理：在双机调试中选择 COM 连接，是因为其对硬件和软件复杂度要求最低。符号表是调试所需文件。

工具：WinDbg

互联网终端的固件调试：IDA 联动虚拟化环境 Qemu 的方式。

符号表：链接器自动生成 PED 文件，它由两个部分构成：

① 私有符号数据：函数、全局变量、局部变量，用户定义的结构体、类、数据类型，源文件的名称、源文件中每个二进制指令的行号。

② 公共符号表：静态函数、全局变重。

4. crackme 分析的入手方法；

通常分析 creackme 的入口函数，常见函数：messageboxA，GetDlgItemTextA，GetWindowTextA，GetDlgItemInt 等。

对于加壳软件，隐藏了 OEP（程序入口点），需要用到 ESP 定律去恢复真正的 OEP。

5. 压缩壳的脱壳步骤及对应的原理；

SEH 结构体异常处理 - 调试检测

1、异常处理结构体（Structure Exception Handler）是 Windows 异常处理机制所采用的重要数据结构。

2、GUI 应用触发消息（异常）-> 放入消息队列（SEH 结构体）-> 查找回调函数（异常处理函数）

1、利用 ESP 定律定位到真正 OEP 之后，使用 LordPE 等工具把程序的镜像 dump 出来。

2、但得到的镜像无法运行，因为无法自动获取导入函数的地址，为此需要修复导入表(Import Address Table, IAT)。

3、通常使用 importrec 工具进行修复，从原文件提取信息后，对脱壳文件进行修复。

6. 花指令的原理及意义；

原理：花指令有可能利用各种指令：jmp，call，ret 的一些堆栈技巧，位置运算等。

意义：1、花指令是程序中的无用指令，干扰反汇编静态分析工具，但是程序不受影响。2、加花指令后，可以隐藏自身的程序结构和算法，从而较好的保护自己。

7. olldbg 典型快捷键的名称、功能；（填空题）

①快捷键的使用

F2：设置断点
F3：打开文件
F7：单步步入
F8：单步步过
Ctrl+F7：忽略异常，单步步入
Ctrl+F8：忽略异常，单步步过
F4：运行，到光标位置暂停
F9：运行，到程序结束暂停
Alt+F9：执行到用户代码
Ctrl+F9：执行到返回指令
Alt+M：查看模块
Ctrl+E：编辑数据
Ctrl+F2：重新调试

②IDA 中用何种方式识别目标程序的 API

利用程序的符号表（symtable）识别接口函数并显示函数名。

③exe 文件通常采用什么方式做 UPX 脱壳

UPX 脱壳工具。

8. 调试中不同断点的使用原理及使用条件；

① 软件断点：X86 系统中为中断指令 INT 3。当程序执行到 INT 3 时，引发软件中断。操作系统的 INT 3 中断处理器会寻找注册在该进程上的调试处理程序，因而调试器就有了上下其手的机会。

条件：通常只能设在 RAM 运行的代码上。

② 硬件断点：X86 系统提供 8 个调试寄存器（DR0~DR7）和 2 个 MSR 用于硬件调试。

条件：需要目标 CPU 的硬件支持。

9. 简单了解一下壳偷代码的基本原理。

stolen OEP code 技术

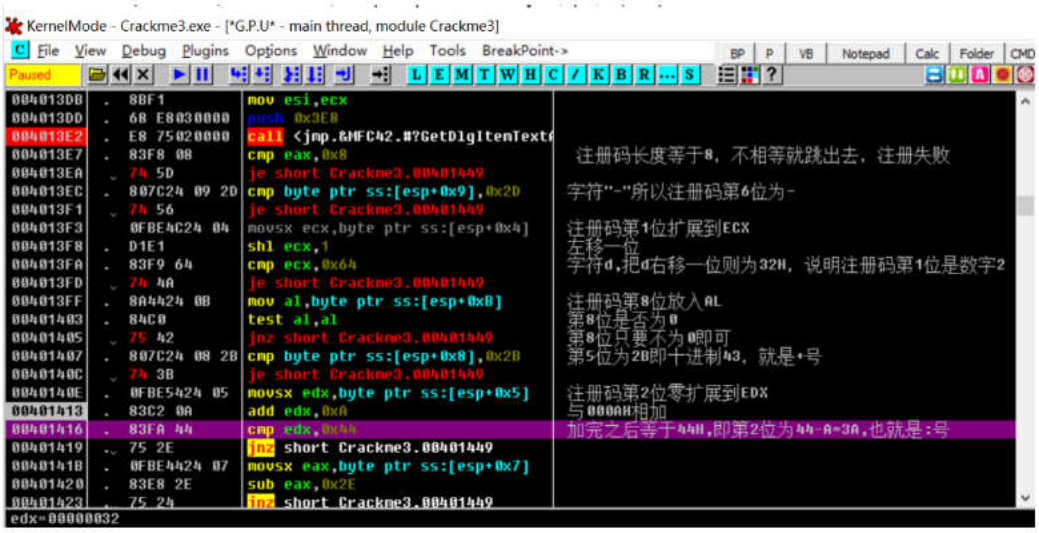
① 某些壳在处理 OEP 代码的时候，将某些正常的控制流代码转移到壳中，而且常伴随着花指令。

② 使起始代码从壳空间开始执行，然后再 JMP 回原空间。如果脱掉壳，代码就会遗失，也就达到了反脱壳的目的。

若 dump 后修复导入表 IAT，这里 OEP 依旧是错误的，程序无法运行。因为壳中代码不会被转储，得不到执行。

解决方法：寻找真正的 OEP，并找到缺失的代码（stolen code）

10. 一个简单的 crackme 实例计算题目；



eax存注册码长度，ecx存注册码字符。
movsx ecx,byte ptr ss:[esp+0x4] 字符扩展ecx，即注册码第一位为esp+4
shl ecx,1 第一位字符二进制左移1位
cmp ecx,0x64 二进制右移回去1位，即chr(0x32)='2'
je 如果相等就跳转（jump if equal），即根据上面的cmp判断
mov al,byte ptr ss:[sep+0xB] al存入注册码第八位
test al,al 第八位为零时，此处为0。第八位非零时，此处为1。
jnz 如果为零就跳转（jump if not zero），即根据上面的test判断
cmp byte ptr ss:[esp+0x8],0x2B 第五位为chr(0x2B)='+'
movsx edx,byte ptr ss:[esp+0x5] 字符扩展edx，即注册码第二位为esp+5
add edx,0xA 第二位字符加0xA
cmp edx,0x44 减回去0xA，即chr(0x44-0xA)=':'

11. 目标程序呈现出各种典型特点，给出对应的分析及破解思路；
考察大家对常见应用程序的分析逻辑，通过工具的查看，发现他有一些特征，比如说字符串不可见（程序加壳），
或者存在一定的调试异常（stolen code）之类的，这种情况之下，需要给出对应的分析和破解的思路。

12.反调试技术的分类及各类的具体技术名称及原理。
① 调试器检测：检测调试器是否存在；
原理：Windows API，手动检测数据结构，系统痕迹检测。
② 识别调试器：识别是否正在调试中；
原理：检测软件/硬件断点，时钟检测，父进程判断。
③ 干扰调试器：令调试失败；
原理：TLS 回调，利用中断，陷阱标志位。