# CodeSafe

- 我的解题顺序是 2、3、4、1，因为不太理解判题方式，而且第一题的第一版代码有多处漏洞，被坑了一下
- 尝试次数有限制，所以本地编译调试，调好了再打远程
- 挑子函数种类最多，结构体名字带 stc 的 rpc_function 准没错

# CodeSafe 1

- rpc_function_1 中 temp.mtt == 0 堆溢出，第二版修复；mtl 变量类型为 unsigned short，整形溢出，我选择 temp.mtt = 129 和 tl = 510
- rpc_function_2 传入非标准 base64 字符串可以绕过 strlen 的检查， 但是仍然受制于框架代码中的 MAX_REQUEST_DATA_SIZE，1024*6/8 = 768 < sizeof bin，因此无法利用

# CodeSafe 2

- rpc_function_2 结尾 sprintf 栈溢出
- pr.t 和 pr.v 都有长度限制，填空格
- function0 懒得仔细想，全 0 可以过

# CodeSafe 3

- rpc_fuction_2 中 sprintf 堆溢出
- t.ts 设为 0x7FFFFFFF 比较方便
- t.k 设为 "sdatsts-afu"，010 是 8 进制，太明显了
- t.url 走 "alibaba.com" 分支触发 sprintf

# CodeSafe 4

- rpc_function_1 中 system 命令执行
- temp.user 长度 127，szUser 长度 63，strncpy 截断用户名，走 "guest"

分支，function1 删掉结尾空格，满足 "admin" 用户名
- 密码 "urejhvg"

# Reverse 4

- trojan.exe 的 main 中很明显的许多 nop，开头的赋值是一段汇编，几个 flag 藏在各处功能里，总是无法满足 ecx == 8，藏着 call，没有反调试，OllyDbg 就可以搞定
- 本来真的是分分钟的事，整个文件传输的逻辑都被处理得妥妥的，只要调用了那个 call 就能拿到 Secret.rar，也是 Intended Solution，但我们就是拿不到，能连接上控制端但是 0x81 的 Token 发过去没有应答
- 于是开始怀疑可能有坑，鉴于我是 IDA 脑残粉，搭环境 VC6 + PSDK 2003.02，选 svchost-Console Profile，生成带符号的 exe，IDA 载入，idb2pat，sigmake，分别 apply 到 Ch4.exe 和 trojan.exe
- 结果真的没坑，改动很小，控制端增加了一个编号为 8 的 Dialog，2 个 Token 的处理代码分别有 2 份，只有一份在本题中用到，还藏了一段 16 位汇编，出题人真是蛋疼；受控端增加了几个 Command
- 换系统、换网络都试过了，80 端口就是挂，软磨硬泡要求出题人另开服务，88 端口一下子就拿到了 Secret.rar，简直凄惨，浪费了好多时间
- 后面的 php 代码套了几层 base64， 坑点是 \xxx 不是 8 进制，是 10 进制