1. 考察内容：倒序异或加密

* Flag输入长度为10
* 异或加密秘钥地址，byte\_425A30
* 密文地址，byte\_425A3C
* 加密过程，byte\_425A30与byte\_425A3C进行倒序异或
* Flag：Good\_Luck!

1. 考察内容：字节内部bit转置加密

* Flag输入长度为22
* 字节内部bit转置，将字节内部最低两位与字节内部最低三四位进行互换
* 密文地址，byte\_429A30
* Flag: KEEP\_YOUR\_DREAM\_ALIVE\_

1. 考察内容：RC4加密算法

* S表初始化等RC4算法实现过程的识别
* 秘钥识别，Qb3jdx+2=SD5lfz
* 密文地址，byte\_429B30
* 解密过程，密文与流密钥异或
* Flag: A\_GOOD\_REST\_OF\_LIFE

1. 考察内容：DES加密算法

* DES加密算法识别，包括初始置换、压缩置换、S盒置换、IP逆置换等内容
* 秘钥内容，Reserse\_
* 密文地址，byte\_42AA30
* Flag：FINBEGIN

1. 考察内容：程序花指令

* 第一花指令位置，401039
* 第二花指令位置，30104B
* 去除花指令方法，nop掉花指令，即将其改为机器码90

1. 考察内容：倒序异或加密+双重异或加密（20分）

* Flag输入长度为10
* 倒序异或加密的秘钥{0x01,0x02,0x03,0x04,0x05,0x06,0x07,0x08,0x09,0x0a}；
* 第二重异或加密秘钥为{0x11,0x12,0x13,0x14,0x15,0x16,0x17,0x18,0x19,0x1a};
* 输入字符串的加密结果为"0123456789";
* 通过编写解密脚本，得到flag
* Flag：GOOD\_LUCK!

1. 考察内容：字节内部bit转置+异或加密（20分）

* Flag输入长度为13
* 字节内部bit转置，将字节内部中间两个bit不变，高3个bit和低3个bit位置互换
* 异或加密的秘钥为{0x01,0x02,0x03,0x04,0x05,0x06,0x07,0x08,0x09,0x0a,0x0b,0x0c,0x0d};
* 密文为{0x5b,0x60,0x69,0x64,0x4e,0x7d,0x46,0x40,0x5b,0x6a,0x63,0x5e,0x01};
* 通过编写解密脚本，得到flag
* Flag: STUDY\_HARDER!

1. 考察内容：RC4加密算法（20分）

* S表初始化等RC4算法实现过程的识别
* 寻找秘钥，为"WORKER";
* 查找程序中隐藏的密文，为{0xd5,0x23,0xa5,0x22,0x75,0xd8,0xb7,0x80};
* 解密过程，密文与流密钥异或
* Flag: BE\_HAPPY

1. 考察内容：DES加密算法（20分）

* DES加密算法识别，包括初始置换、压缩置换、S盒置换、IP逆置换等内容
* 秘钥内容，"REVERSE!";
* 密文为{0x66,0xd3,0xd5,0xf4,0x1a,0xbf,0x81,0x28};
* Flag：NEWYEAR！

1. 考察内容：程序花指令+仿射加密算法逆向（20分）

* 去除JMP花指令，重新打开程序
* 输入字符的范围为a至z
* 仿射加密的参数分别为3和7
* 3的逆元为9
* 密文为"0x71,0x76,0x6c,0x64,0x78,0x74";
* 通过仿射加密的解密函数求得flag
* Flag：higuys

1. 考察内容：花指令+倒序异或加密（20分）

* 地址40117D处存在JMP型花指令
* 将花指令修改为NOP指令，即将其机器码改为90
* 要求输入字符串的长度为15
* 将输入与byte\_427A30处的进行倒序异或加密
* 逐位于byte\_427A40进行比较，相同则通过测试
* Flag：MoreThenFriends

1. 考察内容：异常处理函数+bit位置换（20分）

* 定位除零异常，关键代码逻辑在异常处理函数中
* 4011B6至4011C7处为SEH链表插入操作
* sub\_401005为关键函数入口点，定位关键函数地址sub\_401020
* 针对输入字符，高四位保持不动，第四位中，低两位和高两位进行置换
* 密文位置处于byte\_429A30处
* 编写解密脚本，获取flag
* Flag: SecurityIsPuzzle

1. 考察内容：MD5哈希算法逆向（20分）

* 输入字符串长度大于等于5
* 取前四位进行加密，其输出结果为128bits，猜测该过程可能为哈希函数
* 通过输出结果的长度判断，可能为MD5函数或者SHA-1函数
* 观察sub\_401420处函数，发现四个变量的初始化
* 可以发现4个变量的取值，与MD5中ABCD四个初始变量一致，推测该函数为MD5函数
* 通过暴力破解，求得flag
* Flag: loveX

1. 考察内容：仿射加密算法逆向（20分）

* 要求输入一串大写字母，对其进行仿射加密操作
* 仿射加密的参数分别为9和7
* 9的逆元为3
* 定位密文所处位置，429A30
* 通过仿射加密的解密算法，求取flag
* Flag：HANGON

1. 考察内容：DES加密算法的逆向（20分）

* 输入8字节数据进行加密，符合DES算法加密过程的特征
* 秘钥为TakeEasy
* 通过对加密过程中函数行为的分析，发现sub\_401046为PC1置换过程
* sub\_401014为ShiftLeft函数
* sub\_40100A为PC2置换等
* 推测该函数为DES算法
* 通过DES算法的解密程序，求取flag
* Flag: itiseasy
* 识别到main函数中调用的核心加密函数sub\_40100A。(2分)
* 沿sub\_40100A->sub\_4013D0->sub\_401014->sub\_401190可以进入到RC4的sbox初始化函数中。从这个函数的行为特征能够判断出加密算法为RC4，例如存在初始密钥为{0xfc,0xfb,0x3b,0x0d,0x73,0x6e}；生成值为0-255的S；循环256次复制密钥到临时向量T中；循环256次用T对S进行置换。（10分）
* 加密后的flag存储在byte\_42C700处。flag的前12字节使用RC4加密，flag后7字节使用异或加密，密钥存储在byte\_42C600处。（5分）
* 解密得到的flag为NetworkClassNotGood。（3分）
* 输入的flag要求长度大于等于5，且ascii码在大于等于65小于等于122。识别到main函数中调用的核心加密函数sub\_401005。（2分）
* 识别出哈希算法为SHA256，理由是：存在SHA256的初始特征值{0x67,0xe6,0x09,0x6a}等；sub\_401500对输入数据填充至512的整数倍；sub\_401760是SHA256进行逻辑运算的过程。（10分）
* Main函数中对哈希值的每字节进行了加1操作。加密后的flag哈希值存储在0x0428184处，将其每字节减1还原出flag的哈希值。（5分）
* 爆破得到原始明文为Seeyou。（3分）
* 输入的flag长度要求为8位，加密逻辑就在main函数中。(2分)
* 分析加密逻辑为将输入的每字节的最高位与最低位进行交换，然后与0x0429A38处的密钥进行异或加密。flag的加密结果存储在0x00429A30处。（10分）
* 解密方式为重新与密钥异或一遍，并将每字节的最高位与最低位交换。（5分）
* 得到flag为：TAKEeasy。（3分）
* 0x4010FB处存在jmp花指令，将机器码0xE9改为0x90来进行修复。(5分)
* 加密逻辑就在main函数中，byte\_429A30处存储了加密后的flag。（2分）
* 加密方式为逐字节取反，因此解密只需要对byte\_429A30处的数据取反即可拿到flag。（10分）
* 得到flag为：Beat\*0micron\*!（3分）
* 识别到main函数中主要调用的功能函数为sub\_401023，往里走一直到sub\_401470。该函数对输入进行了检查，要求输入的第10，13，17，20，22字节为’-’。(2分)
* 在sub\_401470中发现程序注册的SHE模块stru\_427078，根据其中的loc\_401530可知发生异常时会调用sub\_401014。(5分)
* 在0x0401509处的除法会引发除0异常，进入异常处理逻辑。（3分）
* 沿sub\_401014进入sub\_4012E0。其逻辑为对输入中的大小写字母进行参数a=3,b=7的仿射加密，而且加密过程中会将字母按字母表顺序映射为0~25再还原成字母。（6分）
* flag加密结果存储在dword\_429A44处。对其中的字母进行仿射密码的解密操作。（2分）
* 得到flag为flag{T1an-71-4a0-9e-a-!}。（2分）

1. 考察内容：hook技术及异或加密（15分）

* VirtualProtect函数推测考察hook技术
* sub\_40100A函数为hook回调函数
* WriteFile函数被hook
* 输入的flag需与一串固定值进行异或加密，加密结果需与realpwd字符串进行比较
* Flag：hookapi

1. 考察内容：字节内部bit转置加密（20分）

* 针对输入的每个字符，其1、2位与6、7位进行互换
* bit转置后，与一串秘钥进行加密
* 加密结果与一串硬编码进行比较，相同的话可以通过测试
* Flag: BUPTFJS

1. 考察内容：花指令+仿射加密算法逆向（20分）

* 004010BE地址处IDA无法处理，可发现花指令
* 将花指令改为NOP指令，即90
* 重新创建函数，并进行展示
* 输入字符的值需大于97，且小于122
* 仿射加密过程中的参数分别为5和5
* 5的逆元为21
* 通过放射加密的解密算法求得flag
* Flag: greatvalue

1. 考察内容：MD5算法逆向（20分）

* 检测输入字符的范围在48-57之间，且输入字符串的长度大于等于5
* 将前四个字节拷贝到新的变量
* 执行sub\_401014函数对变量进行处理
* 函数输出的结果为一串128bits的值，初步判断其为哈希函数
* 通过MD5函数的初始特征值，及FF，GG，HH，II四个专有函数，判断该加密过程为MD5加密过程
* 编写暴力破解程序，求得flag
* Flag：01234

1. 考察内容：变种RC4加密算法的逆向（20分）

* 定位密文位置，在byte\_429B30
* 关键函数sub\_401014循环256次，符合RC4算法特征
* 加密循环部分，通过异或对明文进行加密，符合RC4算法特征
* 值的注意的是，生成的秘钥填加了按位取反，取反后进行加密
* 通过RC4解密算法，求取flag
* Flag: seeyounextyear
* 识别到main函数中通过内联汇编方式添加异常处理函数。（3分)
* 识别程序中的除零异常，并找到异常处理函数，规避主函数中的迷惑函数。（3分）
* 分析异常处理函数中的字节处理和比特位移操作，并编写flag求解代码。（10分）
* 解密得到的flag为WeAllLoveReverse\*\*，其中\*\*为任意字符。（4分）
* 发现是仿射加密变种，由26的循环群转为25的循环群。（4分）
* 发现密钥分别为a=2，b=7，a的逆元为13，加密方式为y=2x+7。（4分）
* 发现输入字母的范围为A~Z。（4分）
* 提供解密逻辑x=（（（y-7）+25）\*13）mod 25。（4分）
* 爆破得到原始明文为ANEASYTEST。（4分）
* 能够判断是双层异或加密。(4分)
* 第一层异或加密为逆序异或加密，需要找到密钥。（4分）
* 第二层为顺序异或加密，需要找到密钥。（4分）
* 编写解密程序。（4分）
* 得到flag为：IsHackBad?。（4分）
* 通过IDA Pro 判断出flag长度为8。（2分）
* 判断加密算法为 DES，且只对明文的前 8 个字符进行加密（6分）
* 分析出密钥为 #$12\*&^)（4分）
* 分析出密文为 0x42, 0xAC, 0x43, 0xD3, 0xF1, 0x44, 0xB1, 0x36（4分）
* 写WP，解出 flag：1\_L0V5\_@（4分）
* 通过IDA Pro 判断出flag长度为6。（2分）
* 判断出对 flag 倒序翻转函数，且只截取翻转后 flag 的前4位。（4分）
* 判断加密算法为 sha256。（4分）
* 写WP，解出 flag：\*\*evol，其中，\*\*为任意值。

1. 考察内容：倒序异或加密+双重异或加密（20分）

* Flag输入长度为10
* 倒序异或加密的秘钥{0x01,0x02,0x03,0x04,0x05,0x06,0x07,0x08,0x09,0x0a}；
* 第二重异或加密秘钥为{0x11,0x12,0x13,0x14,0x15,0x16,0x17,0x18,0x19,0x1a};
* 输入字符串的加密结果为{0x4c,0x7e,0x50,0x7d,0x7c,0x64,0x5a,0x6f,0x54,0x70};
* 通过编写解密脚本，得到flag
* Flag：WeKnowItOk

1. 考察内容：字节内部bit转置加密

* Flag输入长度为10
* 字节内部bit转置，将字节内部最低两位与字节内部最低三四位进行互换
* 密文: {0x4e,0x45,0x45,0x50,0x5f,0x41,0x58,0x45,0x44,0x47}
* Flag: KEEP\_DREAM

1. 考察内容：仿射加密算法逆向（20分）

* 要求输入一串大写字母，对其进行仿射加密操作
* 仿射加密的参数分别为9和7
* 9的逆元为3
* 密文: QFMWRE
* 通过仿射加密的解密算法，求取flag
* Flag: BUPTER

1. 考察内容：MD5哈希算法逆向（20分）

* 输入字符串长度大于等于5
* 取前四位进行加密，其输出结果为128bits，猜测该过程可能为哈希函数
* 通过输出结果的长度判断，可能为MD5函数或者SHA-1函数
* 观察sub\_401420处函数，发现四个变量的初始化
* 可以发现4个变量的取值，与MD5中ABCD四个初始变量一致，推测该函数为MD5函数
* 通过暴力破解，求得flag
* Flag: SEPTEMBER

1. 考察内容：DES加密算法（20分）

* DES加密算法识别，包括初始置换、压缩置换、S盒置换、IP逆置换等内容
* 秘钥内容，"Mbuilder";
* 密文为{0x34,0x69,0xb1,0x78,0x3f,0x5c,0x9e,0x3f};
* Flag：achanceo
* 识别到flag长度为5，其中前4字节参与运算。（2分)
* 识别出程序中的SHA-256哈希函数，提供3-5个明显特征。（8分）
* 发现对hash值进行了+1操作，并考虑了循环机制，并编写flag求解代码。（6分）
* 得到flag为：Luck\*，其中\*为任意字符。（4分）
* 密钥生成涉及到读取内存中数据
* 密钥由异或产生
* 定位密文
* 密文与密钥异或得到明文
* Flag为：KaiJu\_YiZhi\_PEN
* RC4
* AREYOUOK?
* DES
* partplan
* 花指令
* IAMSORRYIAMGULITY

1. 考察内容：倒序异或加密

* Flag输入长度为10
* 异或加密秘钥地址，byte\_425A30
* 密文地址，byte\_425A3C
* 加密过程，byte\_425A30与byte\_425A3C进行倒序异或
* Flag：Good\_Bye！！

1. 考察内容：异常处理函数+bit位置换（20分）

* 定位除零异常，关键代码逻辑在异常处理函数中
* 4011B6至4011C7处为SEH链表插入操作
* sub\_401005为关键函数入口点，定位关键函数地址sub\_401020
* 针对输入字符，高四位保持不动，第四位中，低两位和高两位进行置换
* 密文位置处于byte\_429A30处
* 编写解密脚本，获取flag
* Flag: SecurityIsSoCool

1. 考察内容：花指令+仿射加密算法逆向（20分）

* 004010BE地址处IDA无法处理，可发现花指令
* 将花指令改为NOP指令，即90
* 重新创建函数，并进行展示
* 输入字符的值需大于97，且小于122
* 仿射加密过程中的参数分别为5和5
* 5的逆元为21
* 通过放射加密的解密算法求得flag
* Flag: buptnoone

1. 考察内容：MD5哈希算法逆向（20分）

* 输入字符串长度大于等于5
* 取前四位进行加密，其输出结果为128bits，猜测该过程可能为哈希函数
* 通过输出结果的长度判断，可能为MD5函数或者SHA-1函数
* 观察sub\_401420处函数，发现四个变量的初始化
* 可以发现4个变量的取值，与MD5中ABCD四个初始变量一致，推测该函数为MD5函数
* 通过暴力破解，求得flag
* Flag: SEPTEMBER

1. 考察内容：变种RC4加密算法的逆向（20分）

* 定位密文位置，在byte\_429B30
* 关键函数sub\_401014循环256次，符合RC4算法特征
* 加密循环部分，通过异或对明文进行加密，符合RC4算法特征
* 值的注意的是，生成的秘钥填加了按位取反，取反后进行加密
* 通过RC4解密算法，求取flag
* Flag: SeeYou