服务器端请求伪造漏洞

讲师：空白

学院介绍

学院宗旨：专注网安人才实战技能培养

学院官网：https://edu.hetianlab.com/

合天网安实验室：https://www.hetianlab.com/

主打课程：

《web安全》：OWASP TOP 10漏洞原理及测试

《渗透测试》：渗透测试流程及工具的使用

《安全开发》：用python写一个综合的扫描器

《CTF-PWN》：CTF中的PWN相关

目录
CONTENTS

/01 服务端请求伪造漏洞概述

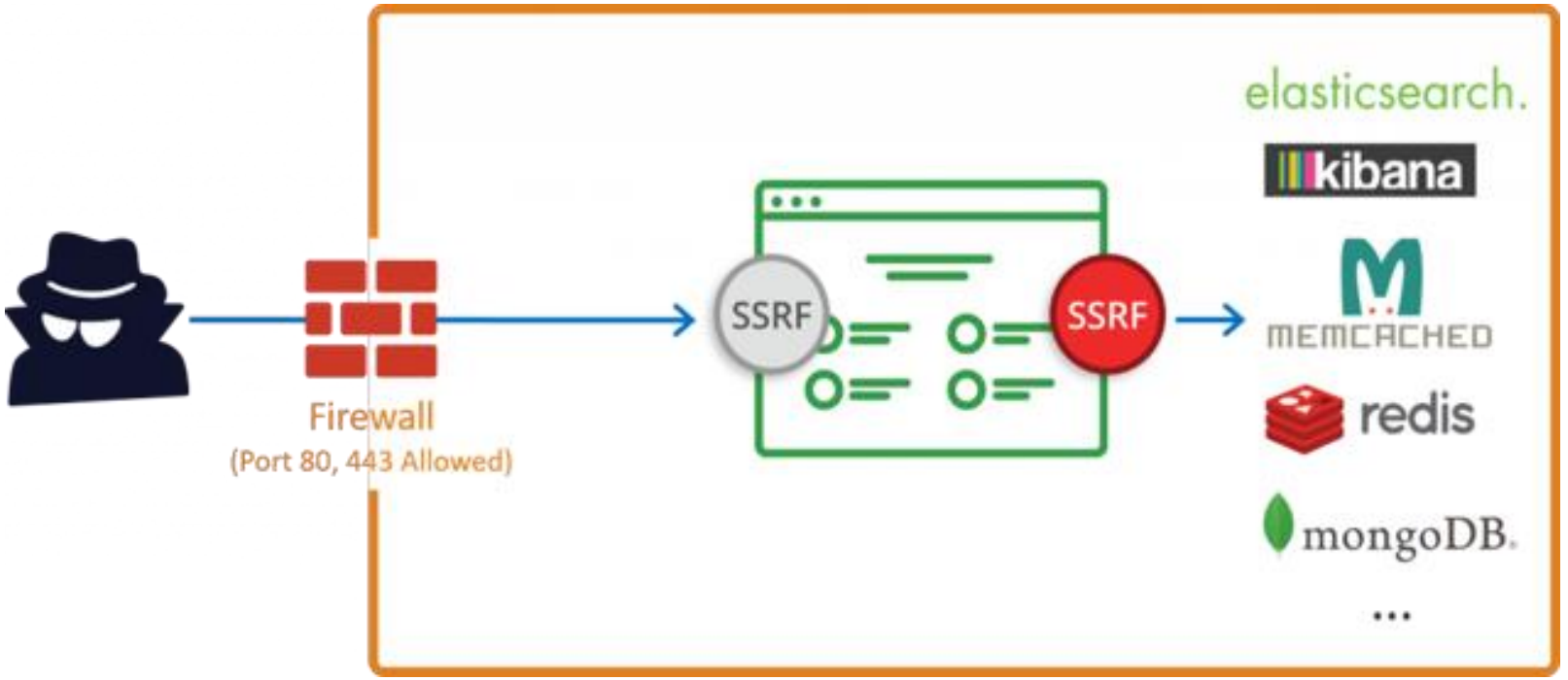
1.1 服务器端请求

是从客户端发起一个请求到服务端，服务端再向另外的服务端发起请求的过程称之为服务器端请求。

以寄快递为例子：我们自己（客户端）首先需要把物品交给快递员（服务端），快递员再把这个物品交到对方手里（服务器）。

1.2 服务器请求伪造

SSRF（server-side request forgery，服务端请求伪造），是一种由攻击者构造形成由服务器端发起请求的一个漏洞。让服务器去请求你通常请求不到的东西。一般用来在外网探测或攻击内网服务。



1.3 形成原因

一般情况下，SSRF 攻击的目标是从外网无法访问的内部系统。漏洞形成的原因大都是由于服务端提供了从其他服务器应用获取数据的功能，但又没有对目标地址做严格过滤与限制，导致攻击者可以传入任意的地址来让后端服务器对其发送请求，并返回对该目标地址请求的数据。

最常见的例子：攻击者传入一个未经验证的 URL，后端代码直接请求这个 URL，就会造成 SSRF 漏洞。

1.4 形成的危害

获取 web 应用可达服务器服务的 banner 信息，以及收集内网 web 应用的指纹识别，根据这些信息进行下一步的渗透

攻击运行在内网的系统或应用程序，获取内网系统弱口令进行内网漫游，对有漏洞的内网 web 应用实施攻击

1.4.1 内网网段

10.0.0.0--10.255.255.255

172.16.0.0--172.31.255.255

192.168.0.0--192.168.255.255

/02 服务端请求伪造漏洞场景

2.1 WEB功能

通过 URL 地址分享网页内容：获取超链接的标题进行显示

文件处理、编码处理、转码等服务：通过 URL 地址把原地址的网页内容调优使其适合手机屏幕浏览

在线翻译：给网址翻译对应网页的内容

通过 URL 地址加载与下载图片：例如富文本编辑器中的点击下载图片到本地；通过 URL 地址加载或下载图片

图片、文章收藏功能：主要网站会取 URL 地址中 title 以及文本的内容作为显示以求好的用户体验

未公开的 api 实现及其他调用 URL 的功能

2.2 url关键字寻找

share、wap、url、link、src、source、target、u、3g、display、sourceURl、imageURL、domain …

如果利用 google 语法（inurl:url=）加上这些关键字去寻找 SSRF 漏洞，耐心的验证，现在还是可以找到存在的 SSRF 漏洞。

/03 服务端请求伪造漏洞分析

3.1 相关函数

PHP 中的函数：

curl_exec() 、file_get_contents()、fsockopen()

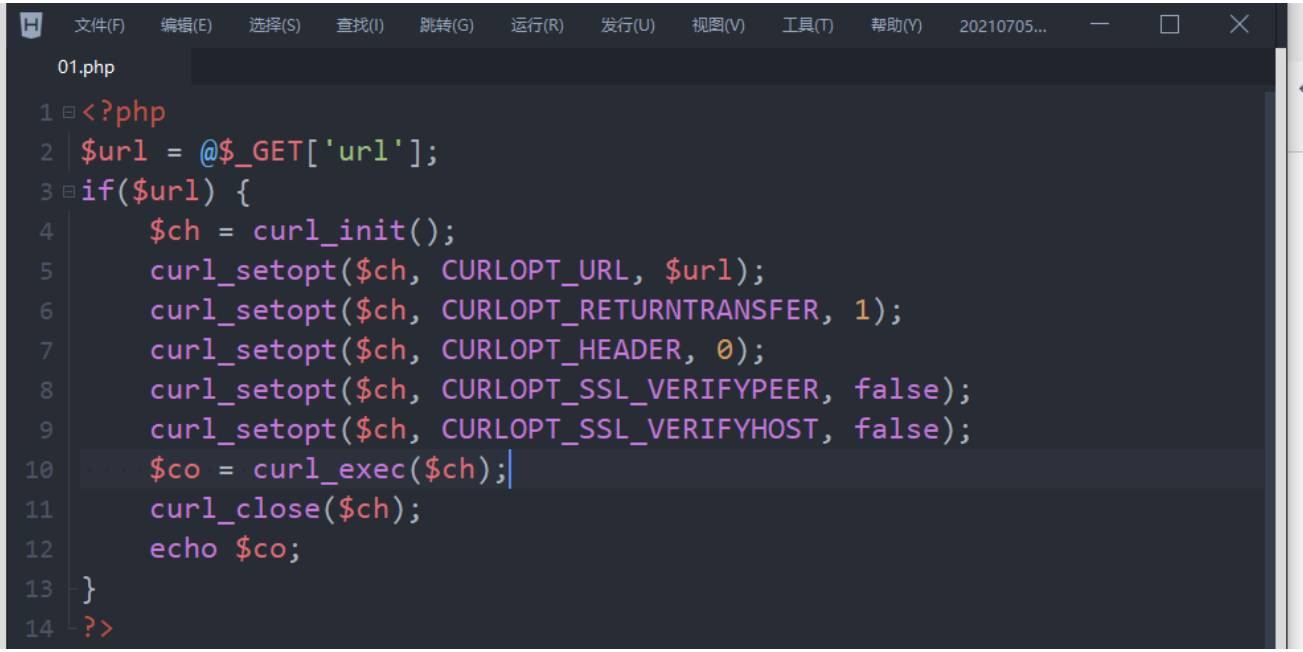
Java 中相关类：

仅支持 HTTP/HTTPS 协议的类：HttpClient 类、HttpURLConnection 类、 OkHttp 类、 Request 类

支持 sun.net.www.protocol 所有协议的类：URLConnection 类、URL 类、ImageIO 类

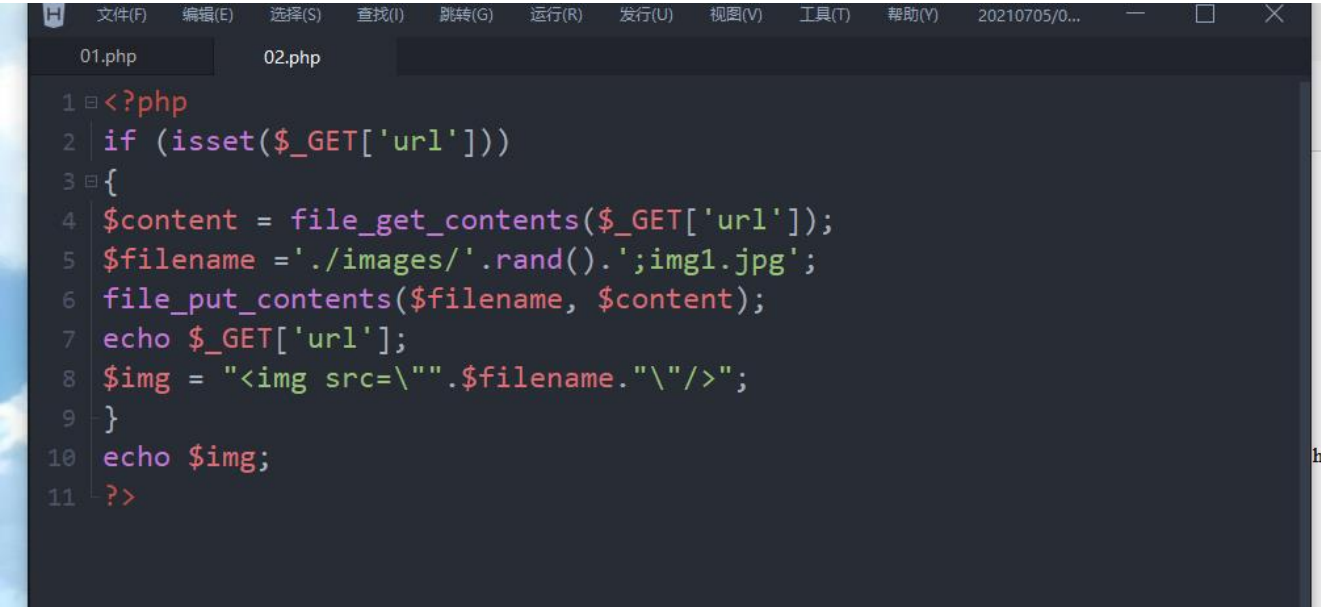
3.1.1 curl_exec()

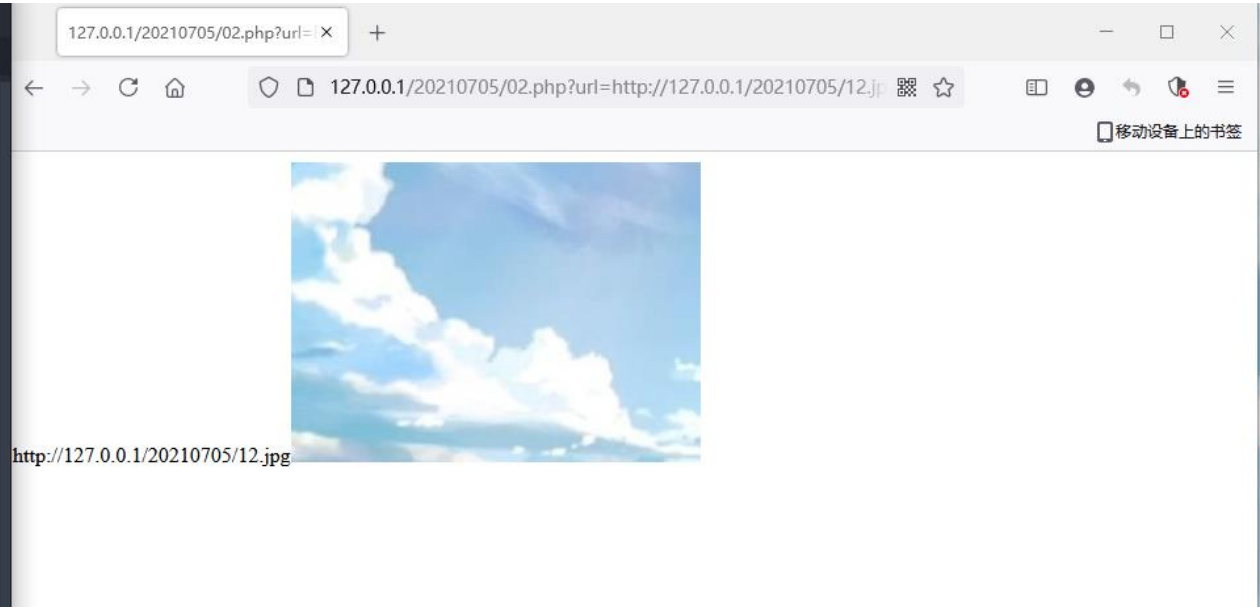
前端传进来的 url 被后台使用 curl_exec() 进行了请求，然后将请求的结果又返回给了前端。



3.1.2 file_get_contents()

这段代码使用file_get_contents函数从用户指定的url获取图片。然后把它用一个随即文件名保存在硬盘上，并展示给用户。





3.1.3 fsockopen()

fsockopen 函数实现获取用户制定 url 的数据（文件或者 html ）。这个函数会使用 socket 跟服务器建立 tcp 连接，传输原始数据

03.php

```php
<?php
        $host=@$_GET['url'];
        $fp = fsockopen("$host", 80, $errno, $errstr, 30);
        if (!$fp) {
        echo "$errstr ($errno)<br />\n";
        } else {
        $out = "GET / HTTP/1.1\r\n";
        $out .= "Host: $host\r\n";
        $out .= "Connection: Close\r\n\r\n";
        fwrite($fp, $out);
        while (!feof($fp)) {
                echo fgets($fp, 128);
        }
        fclose($fp);
        }
?>
```

/04 服务端请求伪造漏洞类型

4.1 分类

有回显：从页面中可以看到返回内容。

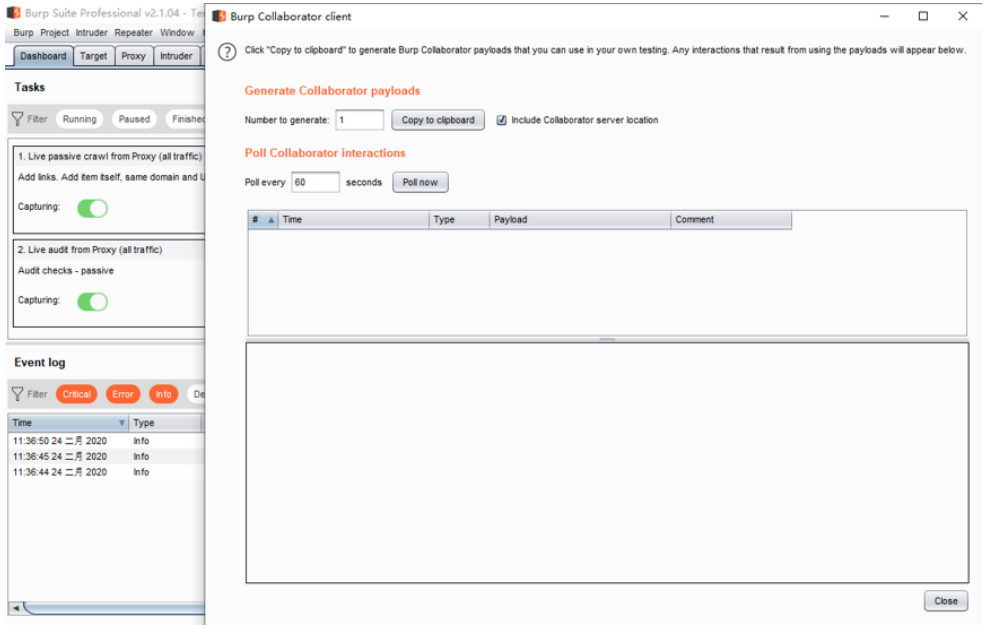
无回显：无法从页面中看到返回内容。

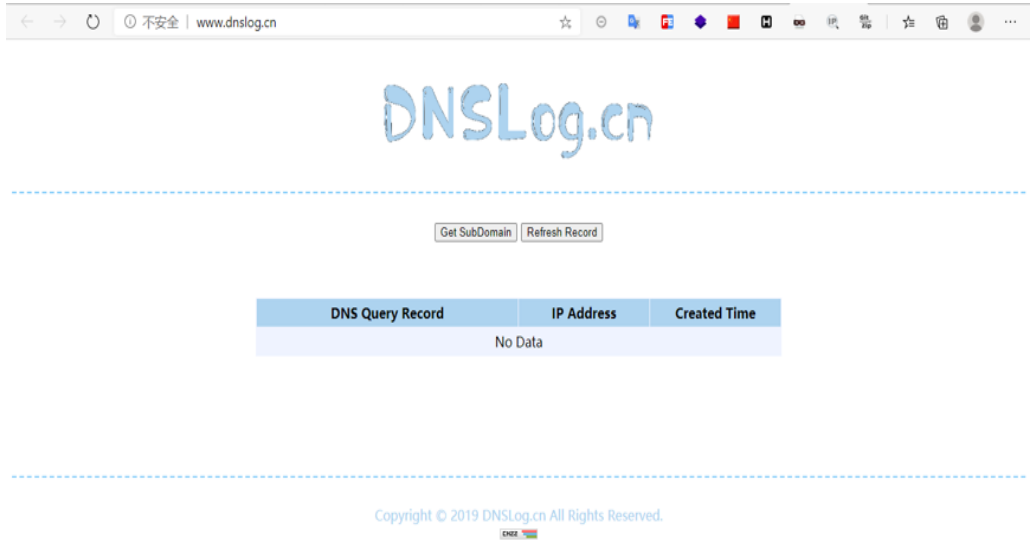
4.2 有回显判断存在

通过目标服务器去请求 url，如果返回了该网站的信息，则说明存在服务端请求伪造。

4.3 无回显判断存在

dnslog





感谢您的聆听

学习工具、资料及课程回放

扫码免费领取