chrome反蜜罐插件-Armor

蜜罐溯源技术原理

蜜罐溯源这部分用到的技术主要是jsonp
jsonp全称是 JSON with Padding ， 是基于 JSON 格式的为解决跨域请求资源而产生的解决方案。

如果某些站点存在jsonp劫持漏洞，如web1这个站点有个jsonp接口暴露者，功能就是返回用户的姓名：

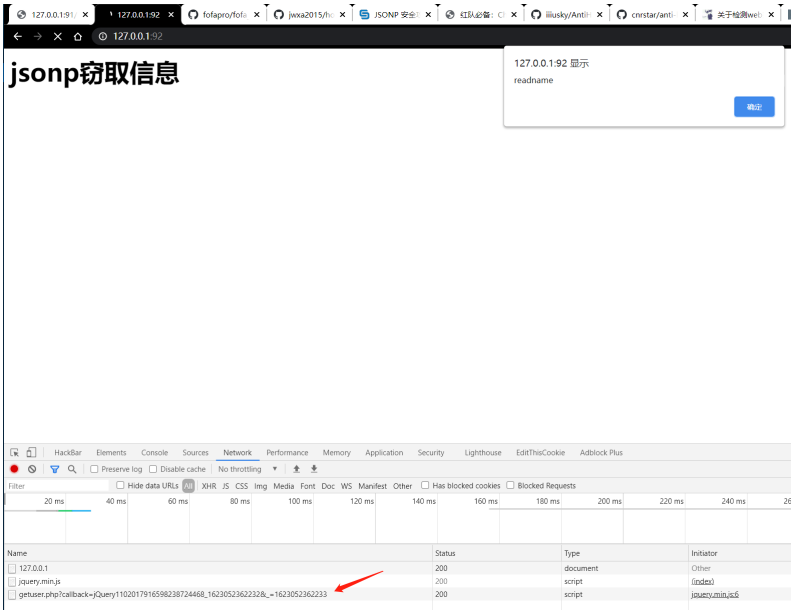
PHP 复制代码

```php
<?php
$callback = $_GET['callback'];
print $callback.'({"id" : "1","name" : "readname"});';
?>
```

正常的请求是这样的：

这时候就可以在蜜罐中插入一个script脚本来跨域获取攻击者登陆的web1站点的姓名信息了

```HTML
<html>
    <h1>jsonp窃取信息</h1>
    <script type="text/javascript"
src="https://cdn.staticfile.org/jquery/1.10.2/jquery.min.js"></script>
    <script type="text/javascript">
        $.getJSON("http://127.0.0.1:91/getuser.php?callback=?",
function(getUsers){
                alert(getUsers.name);
        });
    </script>
</html>
```



实际的蜜罐产品则是收集了非常多常用站点的jsonp漏洞，用来窃取攻击者的个人信息。
本文中的反蜜罐主要指的是，识别蜜罐、并且阻止蜜罐通过jsonp接口拿到攻击者的个人信息。反蜜罐插件更加强调的是后者，识别不到蜜罐问题不大，但是一旦信息被蜜罐窃取走，后果就很严重了。

反蜜罐浏览器插件Armor

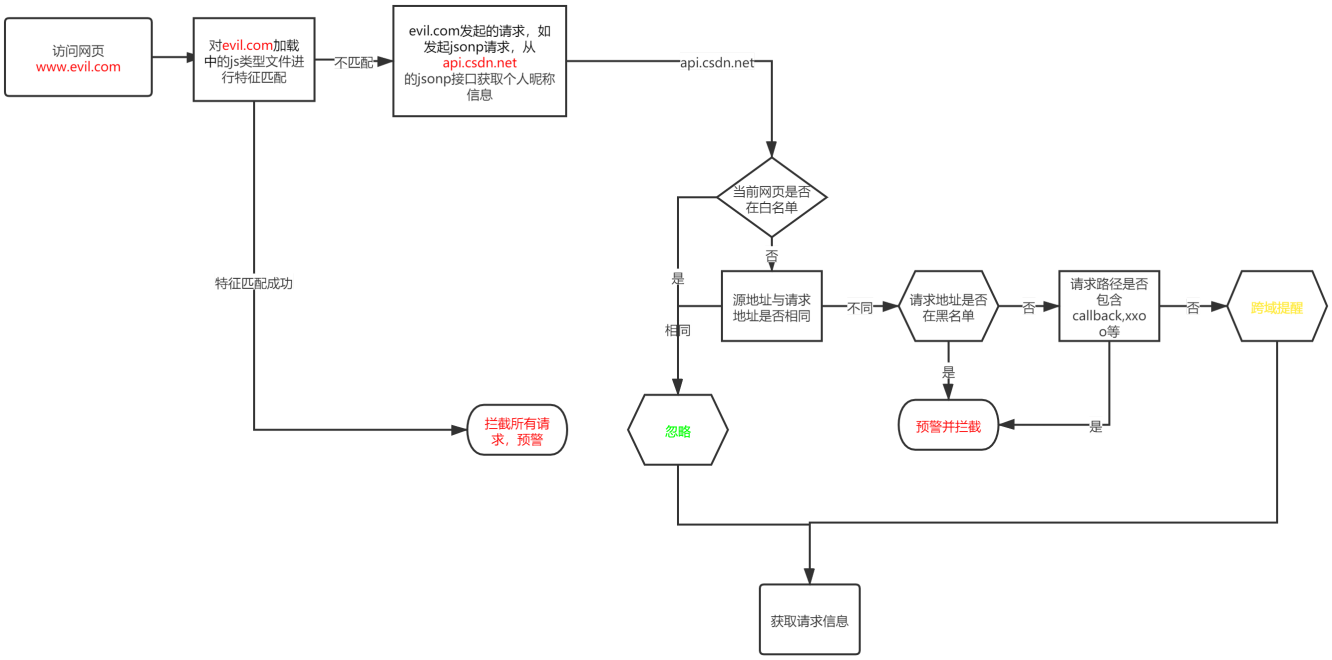
被动探测主要是浏览器插件的形式，在访问的时候插件检测当前是否为蜜罐
目前github上比较好用的两款反蜜罐插件主要是：

- https://github.com/cnrstar/anti-honeypot
  判断当前网页是否有跨域请求，且有黑白名单，黑名单主要就是蜜罐常用的jsonp劫持接口，匹配到则报蜜罐
- https://github.com/iiiusky/AntiHoneypot-Chrome-simple
  主要通过匹配js名称及内容来检测常见的蜜罐

这两个第一个存在漏报的可能，如果黑名单不全的话，还是会被蜜罐拿到信息
第二个也是一样的，不同蜜罐使用的js名称不一样，js内容不一样，比较难很全面的识别拦截
二者都很依赖规则库的完备性。

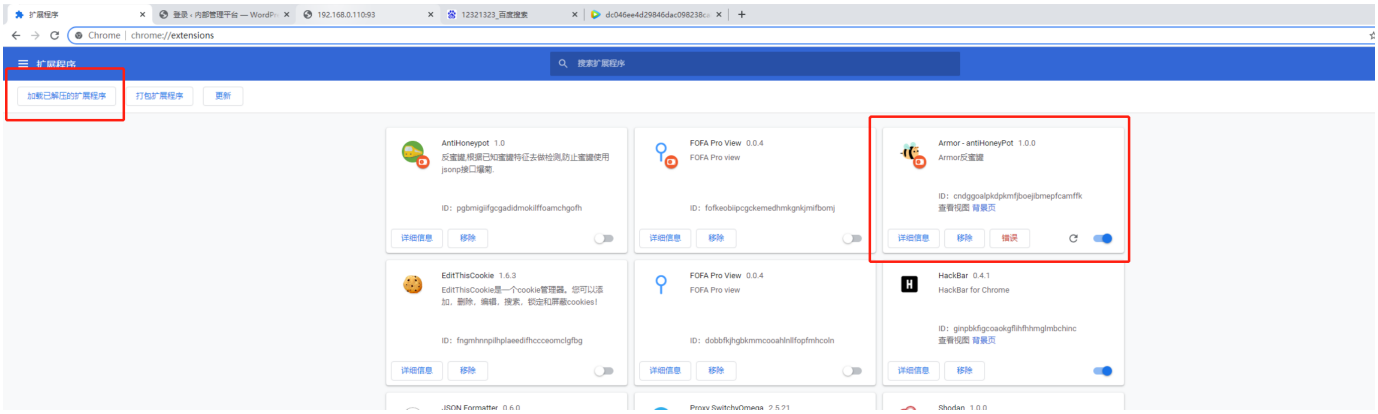
honpot-Armor里将二者结合起来进行改进，整体流程如下：
主要分三个阶段，分别是

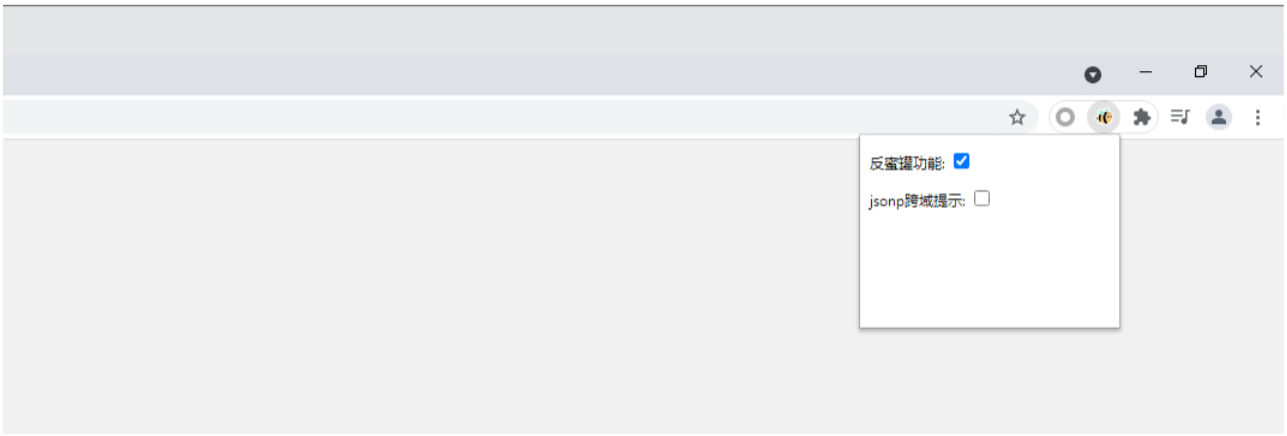
- 探测白名单，由于检测的jsonp接口较多，在正常站点为了不影响使用，添加了白名单
- 对页面加载的js文件进行文件名和文件内容匹配，如Hfish蜜罐，特征就是**x.js**，且里面包含**sec_key**字段
- 对于页面发出的所有请求，对于源地址与目的请求地址，如果相同则忽略，如果不同则进入下一步
- 对于源地址，请求地址不同的请求，黑名单进行匹配，匹配到则直接拦截
- 对于黑名单未匹配到的，则检测请求路径中是否包含callback，cb等字符串，这是由jsonp的特性决定的，还有一些包含ooxx字段(默安的蜜罐)，将这些请求进行拦截，标记为可疑请求。
- 为了减少规则匹配不完全导致漏报的危险，增加了jsonp跨域提醒，跨域请求仍然放行(数量比较多，如百度统计服务，很多网站都有，是正常的jsonp请求)，但是会有通知提醒，用户可以自己选择开启提醒功能或关闭。



使用

开发者模式，选择项目文件夹即可

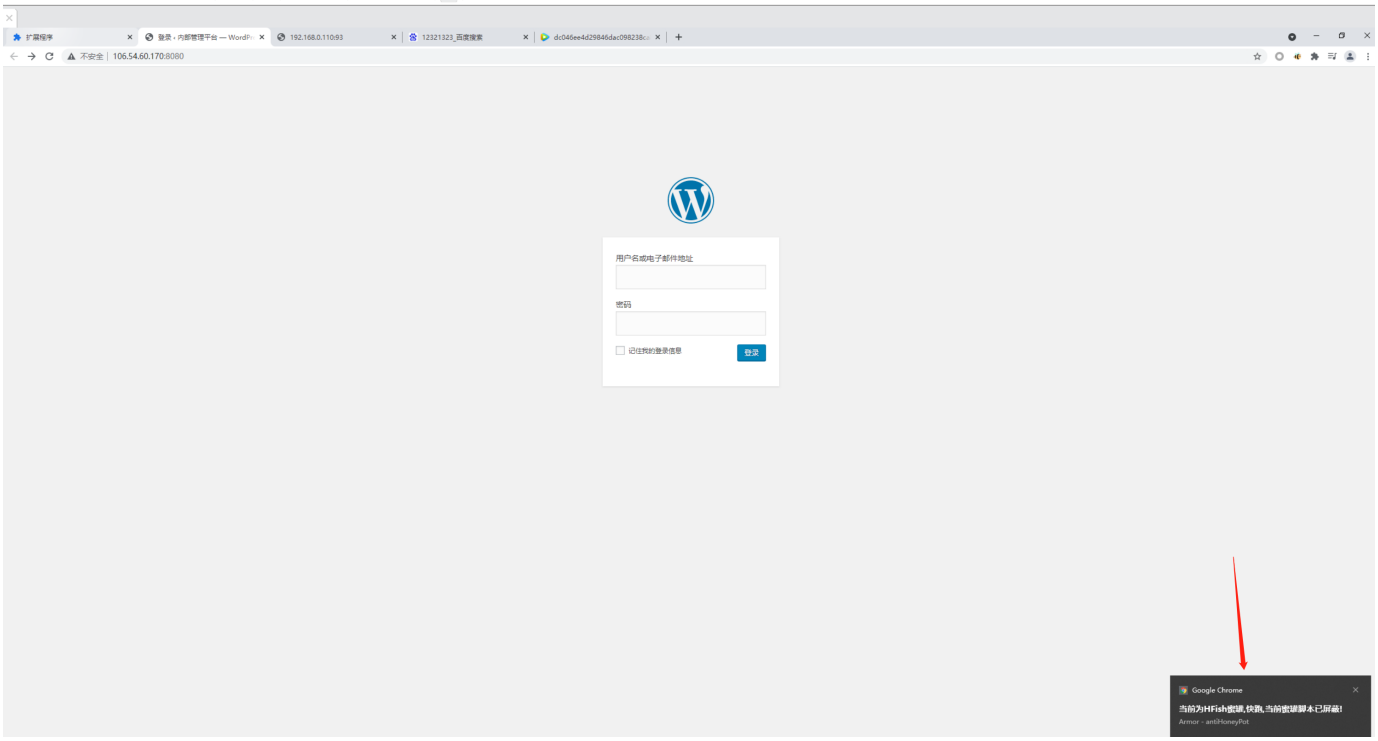


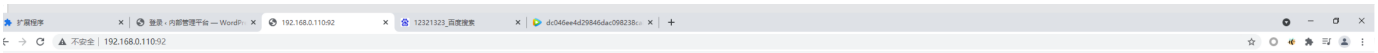


在一些正常网站上，黑名单导致一些正常功能不能使用，可以直接点击关闭反蜜罐功能

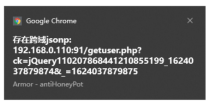
jsonp跨域提醒误报太多时，可以关闭jsonp跨域提示

如果识别到会以chrome通知的形式在右下角弹窗，自己会消失





sonp attack



todo

- 当前通知使用的是Notification函数，通知相对于检测延迟有点多，会导致开关切换之后，之前的通知还在弹
- 增加右键加入白名单之类的更多的交互
- 增加更多规则

参考

https://juejin.cn/post/6844904127932137485
https://github.com/sxei/chrome-plugin-demo
http://www.ptbird.cn/chrome-extensions-storage.html