<u>s33xy</u> / 2017-08-22 05:25:17 / 浏览数 3809 安全工具 工具 顶(0) 踩(0)

## 前言

早前写过一个基于代理的模式的自动化的安全扫描器。

当初是用python来写的一个代理,但是可能我实际安全测试的时候用burpsuite比较多一点,所以写个burpsuite的插件来被动的存储http请求到数据库里。这个也比较简单,所以大概跟同学们说下,代码会开源。

### 插件基础

基础环境的搭建可参考上篇文章,一些API的基础可参考官方文档。

还是同之前的插件一样,所有的burpsuite插件都必须实现IBurpExtender这个接口。由于我们需要被动的收集http的请求,所以还要实现IScannerCheck。

为了界面客观性,加了个UI面板,所以需要实现ITab接口。

### 代码实现

### burpsuite插件的注册声明

```
public void registerExtenderCallbacks(IBurpExtenderCallbacks callbacks) {
                                                                                                                                                                                                                                                                                                                                       &nbsp
```

这个方法每个插件都要实现,基本操作都是类似的。初始化一些变量并注册插件。

为了避免一些静态资源js和css之类的存到数据库,所以进行了后缀的检测。

同样的,避免一些不需要检测的请求入库,还进行了域名的黑名单过滤。预定义如下

```
                                                                                                                                                                                                                                                                                                                                                    &
```

# 可以根据需求来进行具体的修改。

### 检测方法如下

```
public static boolean isblackext(String url) {
                                                                                                                                                                                                                                                                                                                                            &n
```

```
anbsp; anbsp;
      anbsp;
   
    public static boolean isblackdomain(String url) {
        
                                                                                                                                                                                                                                                                                                                                                     
                                                                                                                                                                                                                                                                                                                                                     
anbsp;        
       
      return false;
   }
重复的URL地址将不会再次存储,每次入库前会进行查重,具体为
public int queryrepeat(String url, String body, Connection conn) {
         String sql_exec = "SELECT COUNT(*) as count FROM httplog WHERE url=? ANI
        int flag = 0;
       
            PreparedStatement ps = conn.prepareStatement(sql_exec)
           
            
            
            
                  
anbsp; anbsp;
           
        atch (SQLException e) {
            
       
       return 0;
   
然后就是根据被动的请求,来拿到http请求,进行拆分组装后存到MySQL数据库里。
String url = this.helpers.analyzeRequest(baseRequestResponse).getUrl().toString();
helpers是burpsuite提供的一个帮助类的插件,可由他来获取到http请求的URL, method, body参数等信息。
封装后入库。
public int insert(Map<String,String&gt; sql,Connection conn,PrintWriter stdout) {
                                                                                                                                                                                                                                                                                                                                                     
       
                                                                                                                                                                                                                                                                                                                                                     
                                                                                                                                                                                                                                                                                                                                                     
                                                                                                                                                                                                                                                                                                                                                     
            bnssp;ps.setString(3, sql.get("headers"));
                                                                                                                                                                                                                                                                                                                                                     
            
                                                                                                                                                                                                                                                                                                                                                     
          
       
                                                                                                                                                                                                                                                                                                                                                     
      
         anbsp;
      }
httplog表结构如下,需要先创建表
USE `scan`;
/*Table structure for table `httplog` */
DROP TABLE IF EXISTS `httplog`;
```

```
CREATE TABLE `httplog` (
   `id` int(10) NOT NULL AUTO_INCREMENT,
   `url` varchar(255) DEFAULT NULL,
   `method` varchar(50) DEFAULT NULL,
   `header` text,
   `body` text,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=55 DEFAULT CHARSET=utf8;
```

# 使用

## 直接导入插件后

修改响应的MySQL连接字符串,当Proxy被选中则存储到数据库,否则将不会处理。方便手工测试一些东西时,所以加个是否开启的开关。

# 控制台输出

然后浏览器开启burpsuite代理即可自动的存储到数据库中。

然后就可以根据具体的需要来进行检测漏洞了。

# 开源

# 项目地址

https://github.com/ibey0nd/NSTProxy

点击收藏 | 0 关注 | 0

<u>上一篇:D-Link-Dir-850L-远...</u> <u>下一篇:msdtc后门的实现思路</u>

# 1. 4条回复



我是bey0nd 2017-08-23 02:39:52

膜拜s3xy

0 回复Ta



simeon 2017-08-23 03:01:17

牛逼的帖子,先收藏,再学习!

0 回复Ta



<u>番茄123</u> 2017-09-11 03:10:52

大佬,请问mysql-connector.jar这个包要放在哪里,我的一直找不到这个jar。。。。

0 回复Ta



<u>cabx</u> 2017-10-24 07:02:33

收藏了,好好学习。

0 回复Ta

ᅏᆿ	一四十
⇔ऋ	

# 先知社区

现在登录

热门节点

技术文章

社区小黑板

目录

RSS <u>关于社区</u> 友情链接 社区小黑板