一种分布式的跨站脚本漏洞检测方法

黄细标[1]
HUANG Xibiao

摘 要 针对现有的跨站脚本检测技术尚未对漏洞注入点进行充分研究，且漏洞检测率相对较低的问题，为了提升检测效果，提出一种基于分布式系统的跨站脚本动态检测方法。通过综合考虑字符串长度、字符类型等因素对攻击向量进行了分类与变形，根据输入点、输出点类型自动生成攻击向量与合法向量。采用基于 PhantomJS 的网络爬虫对网站进行更为全面的漏洞注入点分析，并使用 Gearman 来实现分布式任务调度，提高检测效率。采用 PHP 语言设计并实现了分布式自动化跨站脚本检测系统，进一步通过实验和其他相关工具对比分析，表明所提方法能够有效挖掘出 Web 应用中的跨站脚本漏洞。

关键词 跨站脚本；动态检测；分布式系统；PhantomJS；攻击向量

doi：10.3969/j.issn.1672-9528.2024.03.034

0 引言

Web 技术具有开放性、易用性和共享性的特点，随着现代互联网技术的发展，越来越多的应用程序都基于 Web 环境来构建。Web 技术不仅受到越来越多开发者的青睐，还给用户带来了极佳的体验，但也给用户带来越来越多的网络安全威胁，如身份信息泄露、钓鱼欺骗、网站挂马等。基于 Web 技术开发的应用程序往往存在着一些漏洞，其中跨站脚本漏洞（cross-site scripting，XSS）是 Web 应用中最为常见的漏洞之一。由于跨站脚本攻击具有易操作、难以检测的特点，跨站脚本攻击一般将恶意代码植入特定的正常网页中，用户一旦浏览这些含有恶意代码的网页，便会触发代码执行，从而使用户遭受损失。

最早针对跨站脚本进行检测的方法是渗透测试，即从攻击者的角度对网站构造跨站脚本的攻击，来确定漏洞的存在情况。文献 [1] 基于 Web 应用程序提出从攻击作用位置对跨站脚本漏洞进行分类的方法，实现动态的漏洞检测，但该方法只能检测出反射型跨站脚本。文献 [2] 提出基于模拟浏览器行为的网络爬虫来解析 JavaScript 和加载 Ajax 以获取网页中隐藏式注入点，增加了对漏洞注入点的覆盖，但该方法只能在单线程情形下执行，检测效率较低。文献 [3] 通过分析 Web 应用对跨站脚本过滤方式，提出了反过滤规则集对脚本代码进行任何可能的变换，并采用自动爬虫技术进行自动注入和漏洞执行效果自动检测的方案以完成漏洞挖掘，但是在判断脚本是否攻击成功方面存在不足，易发生漏报现象。

通过对以上脚本漏洞检测方法分析，可知目前对跨站脚本漏洞检测的方法存在许多不足。因此，本文针对存储型与反射型跨站脚本提出一种基于分布式系统的动态检测方法。系统采用分布式架构，即由多个工作机同时进行工作，提高检测效率，采用基于 PhantomJS 的网络爬虫对网站进行爬取，对隐藏式的注入点进行分析，扩大漏洞注入点的覆盖面，在攻击过程中动态生成攻击向量与合法向量对漏洞注入点进行攻击，待攻击过程结束对网站进行重新遍历，检验未被发现的漏洞，减少漏报。

1 攻击向量与合法向量

攻击向量是指能够触发执行 JavaScript 脚本代码的字符串，合法向量是指不会对网页造成恶意攻击的有效数据。攻击向量与合法向量的设计同时影响着攻击的成功与否，本文针对网页数据的输入点输出点类型设计出较为全面的攻击向量与合法向量生成方法。

1.1 攻击向量

在实际应用中，很多网站会采用相关工具或设置对客户端输入的内容先进行一定程度的拦截、过滤与净化，要达到跨站脚本攻击的目的，需要解决的问题是如何设计出绕过服务器端对脚本代码的过滤。根据由 RSnake 总结分析的攻击向量库，收集了基于 HTML5 的攻击代码，通过输出点位置，创建了初始攻击向量库（如表 1 所示），针对每种输出点给出一个对应初始攻击向量。自动生成的攻击向量可以直接使用，但可能无法通过网站的过滤机制，导致攻击失败。本文对初始攻击向量进行优化和转换，使之能够绕过网站的

1. 龙岩开放大学 福建龙岩 364000

过滤机制，攻击向量的优化如表 2 所示。

表 1 初始攻击向量库

序号	输出点位置（类型）	攻击向量举例
1	在 HTML 常见标签之间，如 div、p、td 等等	<script>alert(1)</script>
2	在 HTML 特色标签之间，如 iframe、title、textarea 等无法执行脚本的标签	</title><script>console.log(1)</script>
3	在 JavaScript 标签之间	;alert(1)
4	在注释之间	--> xss
5	在 style 标签之间	;height:expression(alert(1))
6	在 URL 伪协议的属性之间，如 href、src 等	javascript:alert(1);
7	在标签的 style 属性中	widtht:expression(confirm(1))
8	普通标签属性中，如 input 标签的 value 属性中	"onmouseover=confirm(1)

表 2 攻击向量的优化

序号	绕过方法
1	将 ASCII 编码的字符转为 HEX/DEC 编码
2	随机对字符进行大小写转换
3	在 HTML 标签或 JavaScript 语句中插入注释、回车、换行、空格、Tab 键等
4	对字符进行全角字符、半角字符混合输入
5	使用 Unicode、UTF-8、UTF-16 等编码格式编码
6	添加相应的模糊前缀
7	混合使用以上方法

1.2 合法向量

根据输入点信息，可以设计出最佳的合法向量，配合着攻击向量，能够提高攻击的成功率。目前大部分文献如文献 [4]、文献 [5] 等，都只对攻击向量进行研究与设计，缺少对合法向量的设计，会降低攻击的成功率。

如图 1 所示为一个 form 表单的提交信息，应当对每个注入点变量进行分析。

```
--<html>--
  ▼<head>
    <meta charset="utf-8" >
    </head>
  ▼<body>
    ▼<form method="post" action="/register.php" >
       <input type="hidden" name="recomand_id" value="2" >
       <input name="name" required= "required" >
       <input name= "age" value>
     ▼<select name="area" >
        <option value="1" >beijing</option>
        <option value="2" >shanghai</option>
        <option value="3" >shenzhen</option>
       </select>
       <textarea name="summary" placeholder="个人简介" ></textarea>
       <input type="submit" value="register" >
     </form>
   </body>
</html>
```

图 1 网页输入点相关信息

当攻击变量 name 时，其它变量选取合法向量，例如：变量 recommand_id 的值设置成 2 或者其它数值类型的值，变量 age 从含义上来解释则是年龄，因此可以尝试优先输入数值类的值，同理，变量 area、summary 均根据相关信息自动生成最佳的合法向量，而并非纯粹的数字与字母的随机组合。

2 分布式检测系统

2.1 分布式系统架构

本文根据传统单线程和多线程检测执行效率的不足，提出一种分布式跨站脚本检测系统，系统设计思想基于 Gearman 分布式任务调度原理，将原来单线程单 PC 检测系统和多线程单 PC 检测系统提升到分布式系统检测。分布式检测系统充分利用了现代网络优势，从而突破大数据跨站脚本检测的瓶颈，将检测中的部分流程并行工作，提升了系统检测效率。

该检测系统由 Job Server、Worker 及 Client 三部分组成。其中 Job Server 部分是该检测系统的核心，Job Server 接收由 Client 发出的请求，根据请求数量分派给相应的 Worker。Worker 在接收到 Job Server 的任务分派后，主要完成注入点分析、攻击与分析、二次遍历这三个模块任务，这三个模块之间相互配合，各 Worker 完成后将相应的结果返回，各 Worker 的并行工作有效提升了检测效率。图 2 为本文的分布式系统架构。

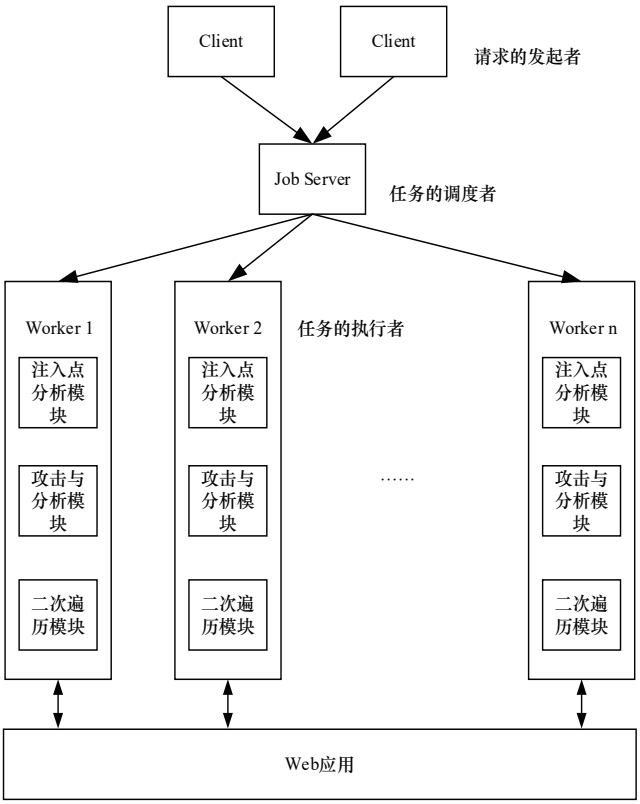

图 2 基于 Gearmand 的分布式系统架构

2.2 注入点分析模块

注入点分析模块主要完成检测的爬取阶段，该阶段从 Client 端中输入起始检测 URL 开始，由 Job Server 将任务分派到各个 Worker 的注入点分析模块中执行。本文采用基于无头浏览器 PhantomJS 的网络爬虫来获取网站页面信息，可以获取到由 JavaScript 执行后加载的数据，抓取网页内容更加全面。为覆盖尽可能多的网页，本文爬虫采用广度优先策略并行化方法爬取网站的页面，在爬取页面过程中同时分析漏洞注入点。

爬虫从某一个或一组链接开始爬取，使用正则表达式找到页面中的其它链接，例如：（1）a、link 标签的 href；form 标签的 action 等；（2）frame、iframe、img 的 src 等；（3）meta 标签的 content 等；（4）JavaScript 的 location.href、windows.open 等等中可能出现 URL 的位置，将原始链接处理成绝对路径，经过 URL 判重处理后，通过这些链接继续搜索下一个页面，这样循环下去，直到待爬取链接为空则停止搜索。在搜索每一个页面的过程中，匹配到注入点，如：（1）Form 表单；（2）含参数 URL；（3）由 JavaScript 或 Ajax 异步提交数据，则将其去重后并以同一格式保存，将爬取到的 URL 存储到 Client 端的 Redis 数据库服务器中。

2.3 攻击与分析模块

攻击与分析模块主要是通过构造数据包向服务器发送 GET 请求与 POST 请求，然后分析响应的数据，从而判定漏洞是否存在。在第二阶段攻击与分析阶段中，Client 端发起攻击任务请求，Job Server 端将注入点分派给 Worker 端去一一完成。因此，系统在同一个时刻对若干个注入点进行攻击，且相互独立，互不干扰。攻击与分析过程如图 3 所示。

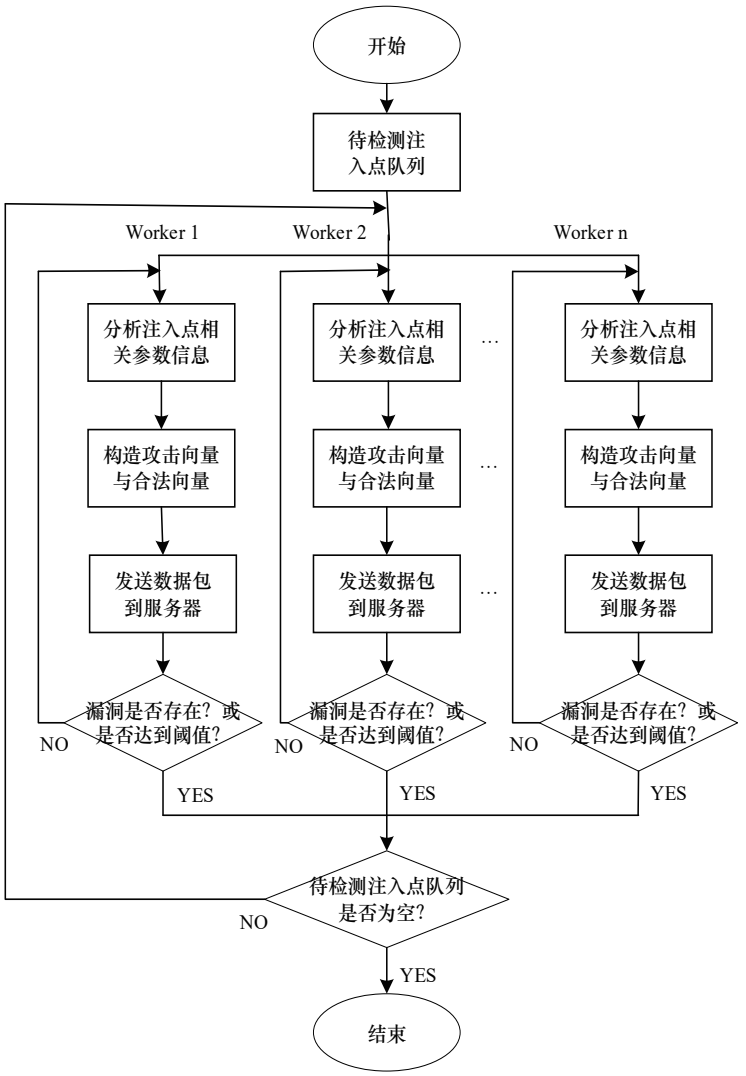

图 3 攻击与分析过程

在攻击过程中为了避免出现误报现象，生成的攻击向量是唯一的，且能够标识出所攻击的注入变量。相较而言，传统的攻击是对每个注入点的所有注入变量进行攻击，而本文系统在一次攻击中只对注入点的某一注入变量进行攻击，其它注入变量自动生成最佳的合法向量，当攻击次数达到所设置的阈值或者该注入变量被确认存在跨站脚本漏洞时，则攻击下一个注入变量，直到所有注入点的每个注入变量都被一一攻击后则停止。

在判断漏洞是否攻击成功中，依旧采用基于 PhantomJS 的网络爬虫来爬取相关的反馈页面以减少漏报，根据服务器响应的数据、所攻击的 URL 返回的数据、所攻击的 URL 在原始页面 URL 返回的数据中是否出现对应的攻击向量，如果出现攻击向量且所在输出点位置与攻击向量的输出点类型一致，即能够触发 JavaScript 代码的执行，则判断该注入变量存在跨站脚本漏洞，若仅仅根据攻击向量的出现，则有可能会造成误报。

在本阶段还需要分析网页中出现新的 URL 链接，能够扩大下一阶段搜索的范围与覆盖面，提高检测率，发现更多跨站脚本漏洞。

2.4 二次遍历模块

二次遍历是在检测结束后进行，由于在对所有页面检测过程中，可能出现用户提交了新的数据，而该数据可能在任意输出点的位置或者在新页面中出现，导致可能出现漏报现象。为了防止出现漏报现象，有必要在所有页面检测完成后，对整个网站进行二次遍历，检查在此过程中是否有新数据产

生，但对攻击前和攻击后没有发生变化的页面，进行二次遍历则会浪费时间，降低检测效率。因此，本文系统使用 MD5 函数对攻击前和攻击后的页面进行编码，然后进行匹配，若相同，则表明没有发现任何变化，无需进行遍历检测。

3 实验与结果分析

为和传统的单线程及多线程检测系统相比较，采用 PHP 语言编程设计并实现 PXS 检测系统、WXS 检测系统、GPXS 检测系统。其中 PXS（PhantomJS-based XSS scanner）是使用基于 PhantomJS 的单线程网络爬虫，是单 PC 的检测系统，但不是分布式架构；WXS（Web XSS scanner）是基于传统多线程网络爬虫的单 PC 检测系统；而 GPXS（gearman and PhantomJS based XSS scanner）是一款基于分布式的检测系统。为验证基于分布式检测系统 GPXS 的优势，将设计的 PXS 检测系统、WXS 检测系统、GPXS 检测系统以及著名的商业漏洞扫描工具 AWVX（acunetix Web vulnerability scanner，version 10.5）[6] 这四种检测系统在相同环境下对两个自建的测试站点进行相关对比测试，结果表明基于分布式的检测系统自动化程度及执行效率更高。

3.1 实验环境

实验需要有一台控制机（Client 部分）与三台工作机（Worker 部分），操作系统均为 CentOS 6.3，内存均为 4 GB，编写语言为 PHP，数据库为 Redis。测试站点操作系统为 Windows 7，内存为 4 GB，Web 服务器为 Apache，开发语言为 PHP，数据库为 MySQL。

3.2 实验结果

实验结果如表 3 ～ 5 所示。实验结果表明，本文设计的系统 GPXS 与其他工具相比较，能够检测出更多页面与注入点，挖掘出漏洞数量较多，且误报数少，提高了检测效率。

表 3 站点一检测结果

系统	URL/ 个	注入点 / 个	XSS/ 个	误报 / 个	HTTP/ 次
GPXS	215	2	4	0	721
PXS	213	2	3	0	720
WXS	212	1	3	0	514
AWVS	185	2	2	0	840

表 4 站点二检测结果

系统	URL/ 个	注入点 / 个	XSS/ 个	误报 / 个	HTTP/ 次
GPXS	439	23	21	0	2130
PXS	434	22	19	0	1920
WXS	430	22	18	0	1738
AWVS	420	21	15	1	2512

表 5 耗时比较

站点	GPXS	PXS	WXS	AWVS
站点一	7 min 24 s	18 min 29 s	3 min 21 s	10 min 27 s
站点二	17 min 18 s	42 min 16 s	14 min 51 s	22 min 3 s

4 结语

本文提出了基于 Gearman 的分布式任务调度系统，采用基于 PhantomJS 内核的无头浏览器对页面中由 JavaScript 执行后加载的数据进行捕获，并在爬取页面过程中分析注入点，在攻击过程中采用适应性随机测试方法自动调整攻击向量的优先级，自动生成最佳攻击向量与合法向量对网站发起攻击，待攻击完毕则对网站进行重新遍历，减少漏报现象。通过与其它同类扫描工具进行实验分析，本文系统 GPXS 发现的漏洞数较多，并且检测效率较高，下一步的主要工作是对基于 DOM 型跨站脚本检测方法的研究。

参考文献：

[1] 陈建青 , 张玉清 .Web 跨站脚本漏洞检测工具的设计与实现 [J]. 计算机工程 , 2010,36(6):152-154.

[2] 王丹 , 刘源 , 赵文兵 , 等 . 一种基于模拟浏览器行为的 XSS 漏洞检测系统 :CN104881608A[P]. 2015-09-02.

[3] 吴子敬 , 张宪忠 , 管磊 , 等 . 基于反过滤规则集和自动爬虫的 XSS 漏洞深度挖掘技术 [J]. 北京理工大学学报 , 2012, 32(4): 395-401.

[4] 张慧琳 , 李冠成 , 丁羽 , 等 . 基于分隔符的跨站脚本攻击防御方法 [J]. 北京大学学报 (自然科学版), 2018, 54(2): 320-330.

[5] 黄娜娜 , 万良 . 一种基于支持向量机的跨站脚本漏洞检测技术 [J]. 计算机应用研究 ,2019,36(2):506-510.

[6] 王丹 , 刘源 , 赵文兵 , 等 . 基于用户行为模拟的 XSS 漏洞检测 [J]. 大连理工大学学报 ,2017,57(3):302-307.

【作者简介】

黄细标（1979—），男，福建上杭人，学士，讲师，研究方向：网络安全。

（收稿日期：2024-01-02）