网站漏洞扫描软件①

冀振燕, 李春元, 莫建杨

(北京交通大学 软件学院, 北京 100044)

摘 要: 论文首先分析了网站安全面临的严峻形势, 指出设计网站漏洞扫描工具的必要性. 论文对系统所采用的网络爬虫技术、网站结构树的构建方法、以及 SQL 注入、XXS 攻击、整数溢出、URL 重定向、格式化字符串等常见漏洞类型的检测方法进行了分析与研究, 为系统的成功实现提供了保证. 论文还描述了系统的架构和子模块的设计. 最后的测试结果表明该系统能够快速全面的检测常见类型漏洞, 目前软件已交付使用.

关键词: 网站漏洞扫描; Web 应用安全; 攻击检测

Web Vulnerability Scanning Software

JI Zhen-Yan, LI Chun-Yuan, MO Jian-Yang

(School of Software Engineering, Beijing Jiaotong University, Beijing 100044, China)

Abstract: The paper introduces the current severe situation of the Web security, and brings up the necessity to design Web vulnerability scanning software. Then it analyzes Web crawler, construction of website structure tree, and detection methods of SQL injection, XSS, integer overflow and URL redirection, which provides a guarantee for developing the system successfully. The paper also describes the software architecture and the design of modules. The final testing results prove that the software is able to detect the common types of vulnerabilities rapidly and comprehensively. The software has been published.

Key words: web vulnerability scanning; web security; attack detection

据全球著名 IT 研究与顾问咨询公司 Gartner 的最新调查表明, 目前 3/4 以上的攻击行为都针对 Web 应用层面而非网络层面, 而 2/3 的 Web 站点都十分脆弱, 容易受到各种各样的攻击[1][2]. 传统网络安全设备防火墙、IDP/IPS 等只能从网络层面上对 Web 应用系统进行保护, 但不能解决由于 Web 应用系统本身存在漏洞而引起的安全问题. 本文针对 Web 应用自身存在的漏洞问题提出一个完整的系统设计方案.

1 系统工作原理

网站漏洞扫描系统工作原理[2][3]: 首先, 客户端判断需要检测 URL 的有效性, 确定是否要进行下一步检测; 接着爬虫该网站, 对目标网站建立网站结构节点树, 在爬虫网站的同时, 对网站结构树的各个节点分别构造具有特定漏洞攻击特征的 HTTP 请求(分别对每个节点都进行常见漏洞类型检测), 向服务器发送构造的 HTTP 请求, 根据此种漏洞的响应特征确定服务器是否存在此种漏洞; 最后, 当检测完毕后再根据漏洞对网站的危害程度将漏洞分类统计输出, 最终在客户端生成一份详细的漏洞检测报告.

2 系统关键技术

网站漏洞扫描系统的成败由建立高效的网站结构树、网络爬虫的搜索策略、攻击检测技术决定的. 下面逐一介绍系统中关键技术的设计.

2.1 网站结构树的构建

网络爬虫抓取网页时须遵循一定的分析过滤策略, 如果不将提取的 URL 分析过滤, 直接添加到爬虫

① 基金项目:北京交通大学中央高校基本科研业务费项目(KRJB12002536)

收稿时间:2013-09-04;收到修改稿时间:2013-09-23

队列或者数据库, 这样就会出现“爬虫陷阱”. 构建网站结构树, 实现站内重复节点的过滤, 首先, 建立以主机为根节点的树形结构, 将提取的有效 URL 根据路径进行拆分, 根据 URL 路径不同分别建立分支节点; 然后将提取有效 URL 与网站结构树的各节点对比, 如果网站结构树中已经含有此节点, 则不将其添加到网站结构树, 如果没有此节点, 则将添加到网站结构树. 通过网站结构树可以过滤网站中重复节点, 提高了网络爬虫的效率.

2.2 网络爬虫技术

聚焦爬虫需要根据一定的网页分析算法过滤与主题无关的链接, 保留有用的链接, 并将其添加到等待抓取的 URL 队列, 再根据一定的搜索策略从队列中选择下一步要抓取的网页 URL, 并重复上述过程, 直到满足某一条件才停止. 聚焦爬虫需要遵循一定的搜索策略, 目前主流的网页抓取策略主要有三种, 即: 广度优先搜索策略、深度优先搜索策略、最佳优先搜索策略[4].

广度优先策略有利于多个爬虫并行抓取网页, 在爬取站内链接时, 如果遇到站外链接就停止爬虫, 抓取网页的封闭性很强. 深度优先策略的优点是能遍历一个 Web 站点或深层嵌套的文档集合, 缺点是当 Web 结构很深时, 就可能无限的爬虫此条链接. 最佳优先搜索策略的特点是可以大幅度减少抓取与主题无关的网页数量, 提高搜索效率, 但是其算法实现较为复杂. 本扫描软件针对特定目标网站上所有的网页及其目录进行爬虫, 爬虫网站规模较小, 不对与主题无关的网页进行爬虫抓取, 结合上述实际情况, 选用深度优先搜索策略实现网站爬虫.

2.3 攻击检测方法

攻击者能够利用 Web 应用程序存在的漏洞入侵 Web 应用系统, 对系统数据进行非法篡改; 利用漏洞破坏 Web 应用系统或者网站的正常运行, 导致用户无法正常使用 Web 服务; 利用漏洞窃取受保护的信息以达到非法目的. Web 应用常见漏洞中会能给用户或者企业造成严重损失的有: SQL 注入、XXS 攻击、整数溢出、URL 重定向和格式化字符串等.

(1) SQL 注入

SQL 注入就是通过把伪造的非法 SQL 命令插入到 Web 表单提交或者页面请求中, 最终达到欺骗服务器执行恶意 SQL 命令的行为. Web 系统开发人员在开发系统的过程中对用户提交数据的合法性没有进行适当的验证, 导致 SQL 注入漏洞的产生, SQL 注入根据注入点的不同可以分为数字型 SQL 注入和字符型 SQL 注入.

1) 数字型 SQL 注入漏洞检测方法

数字型 SQL 注入检测原理: 利用经过特殊构造的字符串验证 Web 应用是否充分过滤了用户提交的表单或者参数, 如果过滤不够充分, 则可以判定存在 SQL 注入漏洞. 即分别构造以下三个 URL 提交到服务器, 根据响应特征判断是否存在数字型 SQL 注入漏洞.

http://www.xxx.com/xxx/show.asp? id=100

http://www.xxx.com/xxx/show.asp? id=100-1

http://www.xxx.com/xxx/show.asp? id=100-0

判断方法: 如果 B 的响应正常且和 A 的响应不相同, 表明存在数字型 SQL 注入漏洞; 如果 C 的响应和 A 的响应相同, 但是 B 的响应错误, 则说明存在数字型 SQL 注入漏洞.

2) 字符型 SQL 注入漏洞检测方法

字符型 SQL 注入漏洞检测方法同数字型 SQL 注入检测原理基本相同, 即分别提交以下三个特殊构造的 URL, 并比较其响应.

http://www.xxx.com/xxx/show.asp? type=abc

http://www.xxx.com/xxx/show.asp? type=abc ‘%2B’

http://www.xxx.com/xxx/show.asp? type=abc ‘%2B’def

如果 B 的响应和 A 的响应相同且 C 和 B 响应不同, 表明存在字符型 SQL 注入漏洞. 如果 C 响应错误或者提示不存在此请求页面, 表明存在字符型 SQL 注入漏洞.

(2) XSS 攻击检测

XSS 即跨站脚本攻击, 是一种常见的 Web 应用漏洞, 它允许恶意攻击者在 Web 页面中插入恶意的 HTML 代码, 当用户浏览该网页之时, 嵌入的恶意代码会被执行, 从而达到非法目的[5]. 随着动态网页技术的发展, 客户端或者服务器需要用某些脚本执行用户提交表单验证或者提供动态元素, 这给 XSS 攻击提供了可能, 黑客可以利用 XSS 攻击盗取各类用户账号和密码, 篡改或者窃取企业数据或者控制受害者机器向其它网站发起攻击等.

XSS 漏洞检测方法: 在正常 URL 中添加可以检测 XSS 漏洞的特殊脚本字符串, 然后将请求发送到服务器, 如果响应中存在此脚本字符串, 则表明此站点存

在 XSS 漏洞. 通常检测的方法如下:

测试 URL:

http://www.xxx.com/xxx.jsp?fitle=news

测试脚本: <script>alert(‘XSS’)</script>

请求 URL:

http://www.xxx.com/xxx.jsp?fitle=news<script>alert(‘XSS’)</script>

为了考虑测试的准确性和高效性, 采取一些脚本混淆技术, 例如改变大小写字母, 对不可见字符的添加等, 将其测试脚本稍加改进[11]. 其测试脚本如下:

<script>alert(${random})< / script>

"%s-->\">'>'\"<WEBSCAN${random}v${random}>"

>”><ScRiPt%20%0a%0d>alert(${random})%3B< / ScRiPt>

>’><ScRiPt%20%0a%0d>alert(${random})%3B< / ScRiPt>

其中随机数由系统随机生成后存储在数据分析与管理系统, 以便后续的结果分析和漏洞确定.

(3) URL 重定向

URL 重定向即网址重定向, 是指把对一个域名、目录或者文件的访问者请求转发至另一个域名、目录或者其它服务器空间, 当用户发送相应的访问请求时将自动跳转到指定的位置[6]. URL 重定向可以有效的实现新旧域名或者目录的无缝对接, 因此无论是对普通用户还是对搜索引擎都是非常有利的, 但是攻击者利用这个漏洞可以诱使用户访问某个非法页面, 记录用户密码并发送到指定地址, 或者让用户下载病毒等.

检测 URL 重定向的基本原理: 在 URL 中添加如下特殊字符串, 然后检测其响应.

http://topsec-webcan.invalid/;?

//topsec-webscan.invalid/;?

topsec-webscan://invalid/;?

检测响应 HTTP 标头中标签 Location, Refresh 的值 (Location 表示重定向, Refresh 表示刷新), 如果字符串中包含 http://topsec-webscan.invalid/ 或者 //topsec-webscan.invalid/则说明其存在 URL 重定向漏洞.

(4) 整数溢出漏洞

计算机中一个整数所能表示的最大值是固定的, 当程序试图保存或者引用比最大值还大的数值时, 就会发生整数溢出[7]. 整数溢出漏洞不易察觉, 如果这个整数是用来计算缓冲区的大小或者计算数组索引排列距离, 那么程序可以直接读写或者修改内存, 使得程序崩溃或者保密数据被窃取.

整数溢出是由于未对提交参数进行充分的边界检测, 或者对整数进行错误的操作引起的, 检测程序是否严格检测或者过滤提交参数就可以判断程序是否存在整数溢出漏洞. 整数溢出漏洞检测方法: 将一个正常的整数参数-12345 构造的请求发送到服务器, 然后将其响应分别与以 2^31-1、-2^31 作为参数的响应进行比较, 如果响应不相同, 说明此网页存在整数溢出漏洞. 同理, 将 12345 作为参数分别与 2^31-1、2^31、2^32-1、2^32 的响应进行比较, 能够判断程序是否存在整数溢出漏洞.

(5) 格式化字符串漏洞

一般情况下, 调用格式化函数必须保证格式字符串与其他参数完全匹配, 但是 C 语言本身并没有一种检查机制去检查参数类型和参数个数的匹配性, 此为格式化字符漏洞攻击提供了可能. 格式化字符串攻击利用了堆栈生长方向和数据存储方向相反的特点, 用后来存储的数据覆盖先前入栈的数据, 改变程序的执行流程, 函数在返回时就跳转到攻击者指定的地址, 继而执行攻击者指定的操作.

格式化系列输出函数的缺点: 参数个数不固定造成的越界访问、提交字符串时添加任意函数的返回地址的地址、通过附加格式来控制向函数返回地址写值. 格式化系列输出函数的缺点说明: 不检查参数的类型或者个数匹配性是产生格式化字符串漏洞的根本原因[8]. 本文检测格式化字符串漏洞的方法如下:

1) 分别向服务器发送正常格化字符串“topsec%dn%dn%n”和特殊构造的字符串“topsec%dn%dn%dn%d%dn%dn%dn%dn%dn%dn”请求, 然后检测其响应是否相同, 如果其响应不同则说明其存在格式化字符串漏洞; 如果响应相同则说明其不存在格式化字符串漏洞.

2) 分别向服务器发送由格化字符串 topsec%dn%dn%dn%d%dn%dn%dn%nd 和 topsec%nd%nd%nd%nd%nd%nd%nd%nd 构造的请求, 然后检测其响应是否相同, 如果响应不同则说明其存在格式化字符串漏洞; 如果响应相同则说明其不存在格式化字符串漏洞.

3　系统设计与实现

Web 应用漏洞扫描工具系统架构设计如图 1 所示, 其主要由六个模块组成: 界面模块、HTTP 及 URL 分析处理模块、网络爬虫兼攻击模块、数据管理与分析模块、检测报告生成模块, 其中 HTTP 及 URL 分析处理模块、网络爬虫兼攻击模块以及数据管理与分析模块三个模块之间相互调用, 共同完成漏洞的检测功能.

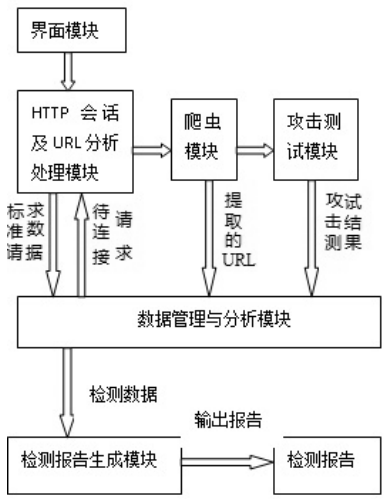

图 1　Web 应用漏洞扫描工具系统架构

(1) 界面模块. 利用 HTML 技术实现应用程序界面, 通过 CGI 接口实现应用程序和界面的信息交流, 界面模块可以通过参数配置控制扫描工具的扫描性能, 命令行解析使得程序能够识别界面模块的调用命令, 用户通过当前扫描信息能够快速判断网站扫描进度以及已经扫描出的漏洞个数.

(2) HTTP 会话及 URL 分析处理模块. 程序解析界面模块的命令行获取初始 URL, 判断 URL 的有效性, 只有有效的 URL 才能得到服务器的响应, 并将非标准格式的 URL 拆分转化为标准格式的请求数据. 在利用攻击测试模块构造请求数据后, 建立 HTTP 会话, 将待连接请求队列中的数据发送到服务器, 监听会话的状态, 对会话连接进行相应的读写处理, 待响应返回之后, 分析响应的有效性.

(3) 网络爬虫模块. 通过深度优先搜索策略爬虫整个网站的目录结构, 获取所有文件的 URL、文件类型等信息, 并根据这些信息确定是否要对网站节点攻击测试, 通过与网站结构树对比过滤站内重复链接或者站外链接, 实现完整的网络爬虫. 网站爬虫的关键步骤是网页解析提取含有链接 URL 的标签, 将链接分析过滤和标准格式化后作为网络节点添加到网站结构树, 网页解析需要分析网页中可能含有链接的所有标签, 解析标签的完整性直接影响网站爬虫的完整性.

(4) 攻击测试模块. 网站的节点类型包括目录节点和文件节点, 目录节点的攻击测试检测目录下的隐藏文件, 用常用文件名替换目录下的文件名, 当确定目录下存在某个文件时, 调用文件攻击检测接口对此文件实施文件攻击检测, 可以检测到无法通过爬虫发现的隐藏文件, 保证网站漏洞扫描的覆盖率; 文件攻击检测是攻击检测的核心任务, 程序开发人员在开发程序时, 可能会对同一个目录下的具有相似功能的文件采用相同的文件名, 但文件后缀各不相同, 因此在对文件进行攻击检测时, 用关键词列表的后缀关键词替换文件后缀检测隐藏文件, 如果发现隐藏文件再对其进行攻击检测. 攻击检测利用常见漏洞的特征, 将构造特殊字符串请求发送到服务器, 根据服务器返回的响应特征判断是否具有漏洞.

(5) 数据管理与分析模块. 构建网站节点结构树, 利用树型结构容易实现数据节点的添加、删除、查找等特点, 将爬虫或者构造的所有节点添加到树型结构, 以便对网站节点的重复性检测和检测报告的输出. 此外, 程序必须利用关键词检测隐藏文件, 关键词保存在文件中, 程序加载关键词后, 要从返回的响应中提取关键词, 将新关键词添加到关键词列表, 通过不断提取新的关键词, 提高程序扫描的精确度.

(6) 报告生成模块. 检测结果以树型结构保存在内存中, 为了方便查看检测结果, 将检测报告和统计信息以脚本的形式输出保存, 将检测数据以二进制形式保存, 利用 html 技术和脚本技术实现报告的动态读取, 用户直接用浏览器就可以查看扫描结果.

4　测试结果

漏洞检测软件检测架设在 IIS 服务器的网站, 以便验证系统总体架构设计的有效性. 检测漏洞类型和数目如表 1 所示, 用 37 分钟完成网站 http://192.168.94.187 的检测, 检测到网站存在整数溢出漏洞 6 个、SQL 注入漏洞 2 个、XSS 攻击漏洞 1 个、字符集丢失漏洞 2 个, 总共检测到该网站漏洞数 11 个. 网站漏洞软件不但能够检测到 SQL 注入、

+表 1 网站漏洞检测结果统计

漏洞类型	危害程度	个数	备注
整数溢出	高	6	
SQL 注入	高	2	数字型 SQL 注入
XXS 攻击	高	1	
丢失字符集	中	2	
总计		11	

XXS 攻击、整数溢出、URL 重定向和格式化字符串漏洞, 而且运行速度高且性能稳定, 其性能和功能都满足需求.

5 结语

论文分析设计了系统所用的关键技术, 描述了系统的设计与实现, 最后还对所实现漏洞检测软件进行集成测试, 测试结果表明漏洞检测软件可以快速高效地检测出 Web 应用中存在的常见类型漏洞, 基本满足用户需要, 目前已投入使用. 为了应对新型漏洞和攻击方式的不断出现, 提高软件的扩展性, 未来需要考虑采用插件技术来实现对漏洞的检测.

参考文献

1 范渊.Web 应用风险扫描的研究与应用.技术应用,2010, 21(11):23−24.

2 陆凯.Web 应用程序安全漏洞挖掘的研究[硕士学位论文]. 成都:电子科技大学, 2010.

3 丁妮.Web 应用安全研究[硕士学位论文].南京:南京信息工程大学, 2007.

4 Liu WT. Design and implement of common network security scanning system. 2009 International Symposium on Intelligent Ubiquitous Computing and Ducation. 2009. 148−151.

5 吴耀斌,王科.基于跨站脚本的网络漏洞攻击与防范.计算机系统应用,2008,(1):11−13.

6 张实睿,许蕾,徐宝文等.一种防止缓冲区溢出的整数溢出检测方法.东南大学学报,2009, 25(2): 219−223.

7 Zhang SM, Xu L, Xu BW. Method of integer overflow detection to avoid buffer overflow. Joumal of Southeast University(English Edition). 2009, (9): 19−20.

8 李鹏,王汝传,王绍棣等.格式化字符串攻击检测与防范研究. 南京邮电大学学报,2007,27(5):84−89.

(上接第 148 页)

3 黄超,齐英剑.SIFT 算法研究与应用.中国传媒大学学报自然科学版,2012(1):68−72.

4 Weickert J, Romeny BMTH, Viergever MA. Efficient and reliable schemes for non-linear diffusion filtering. IEEE Trans. on Image Processing, 1998, 7 (3): 398−410.

5 Perona P, Malik J. Scale-space and detection using anisotropic diffusion. IEEE Trans. Pattern Anal. Machine Intell. 1990, 12: 1651−1686.

6 Weickert J. Efficient image segmentation using partial differential equations and morphology. Pattern Recognition, 2001, 34 (9) :1813−1824.

7 阮宗才,许冠明.基于 AOS 的线性种子扩散图像分割方法研究.微电子学与计算机,2006, 23(12).

8 Lindeberg T. Feature detection with automatic scale selection. International Journal of Computer Vision, 1998, 30(2): 77−116.

9 Brown M, Lowe D. Invariant features from interest point groups. British Machine Vision Conf. Cardiff, UK. 2002.

10 Bay H, Ess A, Tuytelaars T, Gool LV. SURF: Speeded up robust features. Computer Vision and Image Understanding, 2008, 110: 346−359.

11 Alcantarilla PF, Bergasa LM, Davison AJ. Gauge-SURF Descriptors. Image and Vision Computing, 2013, 31(1): 103−116.

12 Agrawal M, Konolige K, Blas MR. Censure: Center surround extremas for realtime feature detection and matching. Computer Vision-ECCV 2008. 2008, 5305: 102−115.

13 马莉.MATALB 语言实用教程.北京:清华大学出版社, 2010:141−176.