Web 应用漏洞扫描系统①

王扬品, 程绍银, 蒋 凡

(中国科学技术大学 计算机科学与技术学院, 合肥 230027)

摘 要: 首先介绍了 Web 应用漏洞安全的严峻形式和对漏洞扫描系统的急切需求, 接着分析了扫描系统的实现原理, 研究和总结了 SQL 注入漏洞、XSS 漏洞、上传文件漏洞等常见漏洞的检测技术. 在此基础上, 设计了 Web 应用漏洞扫描系统的各个模块, 最后的通过实验结果表明该系统设计的可行性.

关键词: Web 应用; 漏洞扫描; SQL 注入; XSS

Web Application Vulnerabilities Scan System

WANG Yang-Pin, CHENG Shao-Yin, JIANG Fan

(School of Computer Science and Technology, University of Science and Technology of China, Hefei 230027, China)

Abstract: First the paper analyzes severe situations of Web application vulnerabilities security and urgent requirements of Web application vulnerability detection technology. Then it analyzes the working principle of the scanning system, studies and summarizes detection methods of SQL injection, XSS and uploads file vulnerability. Based on those achievements, a system infrastructure of Web application vulnerability scanner is designed. The final experimental testing results prove the feasibility of the system design.

Key words: Web application; vulnerabilities scan; SQL injection; XSS

如今, 基于浏览器/服务器模式的系统越来越普遍, 传统的网络安全设备和防护手段一般侧重于防止系统漏洞和网络漏洞, 却很难阻止 Web 应用漏洞, 这样将会给系统带来安全隐患. 根据 360 网站安全检测的统计[1], 2014 年全年, 存在安全漏洞的网站占扫描网站总数的 37.6%, 存在高危安全漏洞的网站占扫描网站总数的 17.0%, 在 360 网站安全检测所有扫出的漏洞中, 跨站脚本漏洞占 33.7%, SQL 注入漏洞占 14.5%, 二者之和接近网站所有漏洞检出总次数的一半, 被篡改的网站约占扫描网站总数的 10.8%. 根据 CNCERT 检测数据[2], 2015 年 1 月国内被篡改网站数量为 10455 个, 被植入后门的网站数量为 3967 个, 其中 GOV 类网站有 237 个, 针对国内网站的仿冒页面数量为 20103 个. 截至 2014 年 6 月, 我国网民规模达 6.32 亿, 网站却频频遭受黑客攻击, 使用户信息频遭泄露, 因为很多网站系统防护脆弱, 往往让攻击者得手, 损害了网站单位的形象, 更有甚者导致经济上的损失. 例如中国铁路总公司网上订票网站 12306.cn 前段时间也被曝出包括账号、明文密码、身份证邮箱在内的用户信息遭到泄露.

为了有效保证网站系统的安全和正常运行, 保障网站的可靠性、可用性和完整性, 企业在注重外部防护的同时, 也应注重对应用漏洞的排查. 这就需要在漏洞对系统的安全形成真正的危害之前, 查找到 Web 应用中存在的安全隐患, 继而采取补救措施消除漏洞, 降低被攻击的风险, Web 应用漏洞扫描就是其中最常用且成本低效果好的技术手段[3]. 市面上的扫描工具都比较昂贵, 比如 IBM 的 APPSCAN, 并且工具中许多功能在一般网站应用中也用不到, 这些工具都不能根据实际情况进行定制, 所以为了适应政府事业单位和中小企业网站的特点, 开发一款可定制和可扩展的扫描系统是很有必要的.

1 漏洞扫描系统工作原理

漏洞扫描一般有两种策略: 主动式和被动式, 本

① 收稿时间:2015-03-25;收到修改稿时间:2015-05-28

系统将采用被动式策略, 即采用模糊测试的方式. 通过模拟对系统的攻击行为, 向目标应用系统提供某种形式的输入, 然后观察目标服务器的响应来发现漏洞, 这种输入是经过精心构造的[4-5], 其过程如图 1.

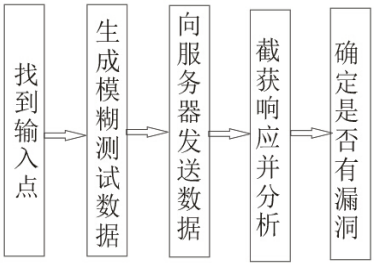

图 1 基于模糊测试的扫描过程

系统采用 B/S 模式, 用户在 Web 页面端验证用户名密码登陆, 验证用户身份后创建扫描任务, 配置扫描信息, 信息会被存储至数据库. 服务器后台将实时运行监控程序, 程序在固定间隔时间会去数据库中查看是否有新的扫描任务, 当发现有新的扫描任务, 后台马上启动扫描模块进行扫描. 扫描启动后, 首先进行爬虫, 然后构造特定漏洞特征的 URL 请求, 向目标网站服务器发送请求, 根据目标服务器返回的响应来判断该 URL 中是否存在漏洞 . 在扫描启动的同时, 后台开启将一个新的线程监控扫描进度, 如果后台监控程序检测到有扫描任务已经完成, 就把扫描结果保存进数据库中, 服务器从数据库中得到扫描结果信息并生产扫描报告[6], 流程如图 2 所示.

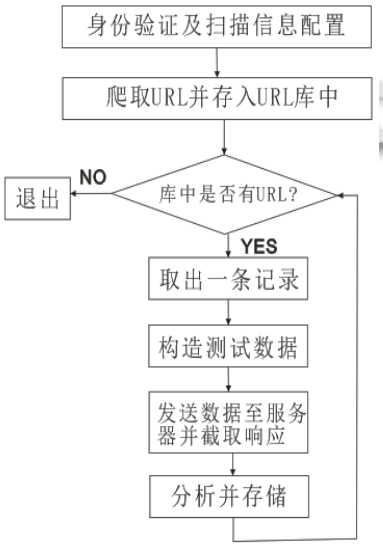

图 2 漏洞扫描系统工作流程

2 Web应用漏洞检测技术

漏洞检测技术是通过模拟黑客攻击的方法去测试应用漏洞是否存在, 对目标网站有可能出现的应用漏洞进行逐项检测. 下面将通过对 Web 应用漏洞进行研究, 分析每种类型漏洞特征, 形成 Web 应用漏洞检测方法, 结合模糊测试原理使扫描系统能够自动化检测.

2.1 XSS 检测

当应用程序收到含有不可信的数据, 在没有进行适当的验证和转义的情况下, 就将它提交给一个网页浏览器, 将会产生跨站脚本攻击(简称 XSS). XSS 允许攻击者在受害者的浏览器上执行脚本, 从而劫持用户会话、危害网站、或者将用户转向至恶意网站. XSS 漏洞的攻击点可分为 2 大类: 表单和含参数的 URL. XSS 漏洞目前一般可分为以下 3 类: DOM．based XSS 、Reflected XSS 、Stored XSS[7].

XSS 漏洞检测原理: 假如正常的请求链接如下:

http://www．example．com/search．asp?value=***

经过系统构造后的变体如下:

http://www.example.com/search．

asp?value=<script>alert(‘XSS’)< / script>

向服务器发送该变体链接请求, 如果在返回的 response 中能够捕获到特定的数据, 则该 URL 中存在 XSS 漏洞.

(1)反射型 XSS 挖掘. 这种 XSS 的输入点在 URL 上, URL 常见的组成模式在上面已经介绍过了, 入侵者可修改[path]、[query]、[fragment]三部分, 所以首先要把这类路径的参数识别出来. 把构造过的 URL 发送到服务器, 然后根据请求后的响应来检查是否有弹窗或是否引起了脚本错误. 假设出现了上述情况, 可以认定目标存在 XSS 漏洞

(2)存储型 XSS 挖掘. 这种 XSS 一般是由表单提交, 被存入服务器端中, 当用户浏览时在页面上输出. 表单提交后跳转的页面有可能是输出点, 表单所在的页面也有可能是输出点, 如果表单提交后没有了, 这就需要对网站的页面进行爬取分析, 比如通过页面缓存来判断目标的页面是否有变动.

(3)DOM XSS 挖掘. 通过完全的黑盒测试方式进行模糊测试出输入点, 然后判断 DOM 树中有无特定的值, 如 fuzzing 测试过程中向服务器提交的请求中有这样一行代码[11]:

Document.write(‘d0m’+’x55’)

当这个代码在服务器被执行了, DOM 树中会有 d0mx55 节点, 判断代码如下:

If(document.documentElement.innerHTML.indexOf (‘d0mx55’)!==-1){

Println(“found dom xss”);

};

漏洞检测执行过程:

(1)扫描简单的 XSS 漏洞

将所有的特殊字符(</->,/*,",',`,=)放在一起, 组成一个字符串

payload=['RANDOM</->','RANDOM/*','RANDOM"RANDOM',"RANDOMI'RANDOM","RANDOM`RANDOM","RANDOM ="]

RANDOM 是随机生成的一个字符串, 它的作用是帮助在响应报文中定位到特殊字符的位置. 把这个字符串构造到请求链接里, 发送到服务器, 返回一个 HTTP RESPONSE, 如果所有特殊字符没有被过滤或者没被编码就直接反射回来, 则存在 XSS 漏洞.

(2)扫描复杂的 XSS 漏洞

采取一些混淆技术, 构造复杂的变体, 比如变换大小写、16 进制变换、URL 编码等. 常用的变体内容如下:

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

<%00script>alert(${RANDOM})%3B</script>

%3Cimg%20dynsrc%3D%22JaVaScRiPt:alert%28${RANDOM}%29%3B%22%3E

2.2 SQL 注入

利用特意构造的SQL语句来欺骗服务器执行恶意的操作, 通过执行 SQL 语句中的恶意代码, 从而获得非法权限, 对数据库进行超越本身权限操作, 控制服务器系统, 获得重要的用户信息和数据[9]. 维基百科对 SQL 注入的定义如下: SQL injection is a code injection technique that exploits a security vulnerability occurring in the database layer of an application.

SQL 注入漏洞检测原理: 执行攻击者 SQL 命令的行为按照从简单到复杂的顺序, 选择测试脚本进行测试, 发现漏洞就返回.

(1)基于服务器响应信息中的特征. 构造两个测试 URL, 分别是在需要检测的 URL 最后加上单引号“’”和分号“;”, 提交链接到网站服务器, 假如返回错误信息, 则该网页存在 SQL 注入漏洞.

http://website/index.jsp?id=1'

http://website/index.jsp?id=1;

(2)基于常规的 1=1,1=2. 将需检测的 URL 改写为三种不同形式, 即 URL、URL and 1=l、URL and l=2, 分别把这个链接提交给服务器. 如果服务器对 URL 和 URL and l=1 的响应相同, 而服务器对 URL and l=2 的响应中提示 BOF 或者 EOF 或者找不到记录或者显示内容为空, 则认为该网页存在 SQL 注入漏洞[10].

http://website/index.jsp?id=1

http://website/index.jsp?id=1 and 1=1

http://website/index.jsp?id=1 and 1=2

SQL 注入漏洞检测执行过程:

①从 URL 库中选择带参数的 URL, 并解析出参数;

②读取一个参数, 用测试脚本替代它, 构造测试 URL;

③进行基于服务器响应信息中的特征检测, 如果检测出漏洞, 保存漏洞信息, 获取下一个参数进行测试;

④如果第三步没有检测出漏洞, 则进行基于 1=1,1=2 的漏洞检测, 如果检测出漏洞, 保存漏洞信息, 获取下一个参数进行测试;

⑤URL 中参数检测完后, 获取下一个 URL 测试.

2.3 上传文件漏洞

很多网站需要用户上传文件, 比如图片上传、附件上传等功能, 这些上传的地方如果验证方式不严格就会存在安全缺陷, 在黑客渗透中是一个非常重要的突破点. 攻击者通过这些漏洞绕过验证机制, 将会造成被攻击的网站被攻击者上传后门和木马. 上传文件漏洞攻击是一种出现概率大且安全隐患严重的攻击方式.

上传漏洞检测原理: 往往一个文件通过 HTTP 协议进行上传时, 将以 POST 请求发送至服务器, Web 服务器接收到 POST 请求后并且同意后, 用户客户端将与 Web 服务器将建立连接, 然后传输数据. 扫描系统通过在文件数据存储至服务器之前, 检测网站是否有对文件扩展名、content-Type、路径参数和文件内容进行验证来查找漏洞[13].

漏洞检测执行过程:

①文件扩展名检测. 上传一个.php 文件或者类似文件, 如果上传成功, 则系统没有对文件扩展名进行合法验证.

②MIME 类型检测. 构建一个上传文件的 request 包, 然后修改 Content-Type 的内容, 例如把

Content-Type=text/plain

修改为 Content-Type=image/gif.

如得到服务器上传成功的响应, 则系统没有对 MIME 类型进行合法验证

③路径参数检测. 目录路径参数检测就是检测路径是否合法, 上传一个构造过的路径参数, 如 D:/photo/image/test1.php .gif/test2.gif, 测试系统是否对参数进行验证.

④文件内容检测. 可能有些上传的文件, 通过修改了扩展名, 使它文件类型合法, 但文件内容和文件类型不一致, 通过进行文件头检测, 验证扩展名与内容是否相符.

3 漏洞扫描软件系统的设计和实现

该扫描系统将通过浏览器向用户提供友好的交互界面, 爬虫模块获取网站的链接结构树和表单等信息, 扫描模块实现对指定目标网站系统的扫描探测, 文件输出模块生产目标主机的漏洞报告. 根据爬虫模块所收集的信息, 分别对不同的链接调用合适的漏洞扫描插件, 每个插件负责检测一个特定的漏洞, 这也方便实现对系统的开发, 同时, 也提高了漏洞扫描的可扩展性. 如漏洞扫描系统结构图所示, 系统的主要组成部分是控制调度模块、界面模块、爬虫模块、扫描模块、文件输出模块等.

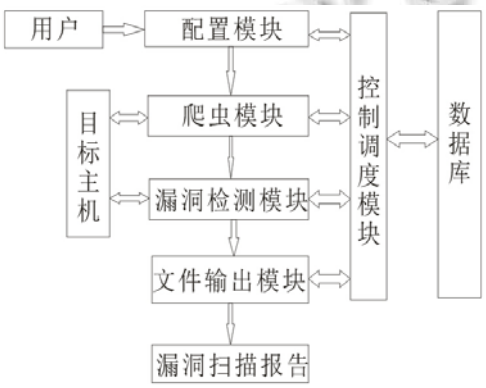

图 3 Web 应用漏洞扫描系统结构

3.1 控制调度模块

该模块是扫描系统中各模块的协调和控制中心, 它的作用贯穿一个扫描任务的全生命周期. 控制调度模块通过运行监测程序检查是否有新建的扫描任务, 发现新任务就立即根据配置信息进行初始化, 然后调用漏洞检测模块按照配置规则进行扫描, 当监测到扫描任务完成时, 调用文件输出模块对扫描结果进行处理[8]. 控制调度模块会为每个漏洞扫描任务开启一个扫描线程和一个监测线程, 扫描线程用于调度其他模块完成扫描任务, 监测线程通过实时监测了解扫描任务进度和状态. 控制调度模块中还含有数据库操作接口, 其他模块所采集的数据和所需要的数据都通过它与数据库交互.

3.2 界面模块

利用 JSP 技术开发一个页面作为系统前端, 用户在前端建立扫描任务, 界面模块接收来自用户的指令, 完成扫描任务的创建, 暂停, 删除, 任务管理及漏洞扫描结果的下载, 用户可以按照自己的需求在界面设定自己的扫描参数. 在任务执行过程中, 可以在界面模块查看任务的扫描进度. 界面模块最终会把用户信息和扫描配置信息存储至数据库, 供其他模块从数据库调用. 总之, 用户通过界面模块与服务器进行交互.

3.3 爬虫模块

爬虫是漏洞扫描系统中很重要的一部分, 通过网站爬虫来获取网站的 URL 结构树, 并搜寻表单和其它注入点, 爬虫的结果会很大程度影响漏洞检测的覆盖率和效率. 爬虫是通过一个网页的地址来寻找网站内其他网页, 它从一个给定的页面开始, 读取网页的内容, 从而找到在网页中的其他地址, 接着通过这些地址寻找下一个网页, 一直循环下去, 达到结束条件时停止抓取, 最后把这个网站的相关网页注入链接和表单等信息全都抓取完. 一般爬虫有 2 种策略: 深度优先策略和广度优先策略. 因为这个漏洞扫描器主要用于扫描政府事业单位和中小企业网站, 网站规模不大, 网页纵向维度不深, 为了提高速度和检测覆盖率, 所以选择广度优先策略. 配置模块工作结束后, 控制调度模块会启动爬虫模块, 爬虫工作内容如下:

①初始化, 从配置模块获得初始 URL 等信息, 并测试 URL 有效性;

②获得 web 服务器, 操作系统, 数据库等版本型号信息, 并经控制调度模块存入数据库;

③循环抓取网页链接和表单等信息, 并控制经调

度模块存入数据库;

④过滤所抓取的信息, 并分析是否要进行漏洞检测;

⑤把链接保存至由初始链接为首节点的结构树中;

3.4 漏洞检测模块

通过模拟黑客攻击的方式, 利用浏览器页面端和服务器交互的过程, 向目标网站服务器发送经过特殊构造的带有漏洞检测特征的请求, 然后从目标网站服务器的响应来检测该 URL 是否存在的 Web 应用安全漏洞. 由一个 URL 为输入, 然后得到一个或多个注入点, 各个漏洞检测插件获取相对应的注入点, 接着向可注入点发送特别设计的请求或数据. 具体过程如下: 1.HTTP 请求构造; 2.HTTP 请求发送; 3.HTTP 响应接收; 4.HTTP 响应分析和漏洞信息存储. 漏洞检测模块中包括 XSS、SQL 注入漏洞、上传文件漏洞和弱密码等漏洞检测插件. 当爬虫模块工作结束后, 控制调度模块会启动漏洞检测模块, 为每个漏洞检测检测插件开启一个线程, 通过多线程的方式检测爬虫模块采集的链接和表单中存在的漏洞. 链接和表单全部都检测完后, 关闭线程并把结果提交给文件输出模块.

3.5 文件输出模块

所有输出均由文件输出模块进行管理, 该模块会对扫描结果的处理, 然后生产扫描报告. 从数据库中调用扫描结果进行全面的分析处理, 得到目标网站系统的操作系统服务器版本型号信息、漏洞列表, 漏洞风险等级、漏洞详细信息、Web 漏洞风险统计和安全建议, 生成扫描漏洞报告供用户下载. 报告格式有 PDF、HTML 及 WORD 三个形式, 系统默认将所有报告信息写入扫描任务名称+时间.pdf 文件.

4 实验结果与分析

为了检测该漏洞扫描系统设计的有效性, 实现了原型系统. 在 Web 页面上新建了两个扫描任务, 一个是某高校 BBS 漏洞扫描, 一个是某地级市官方主页网站漏洞扫描, 对高校 BBS 总共抓取了 146 条 URL, 对政府网站总共抓取了 694 条 URL, 从扫描开始到结束总共花了 70 多分钟, 具体漏洞数据如表 1 所示.

表 1 漏洞扫描结果

	SQL 注入	XSS	弱密码	上传漏洞
某高校 BBS	3	2	5	1
某政府网站	1	3	0	0

以 XSS 漏洞为例介绍系统检测漏洞过程, 向服务器发送以下链接

http://bbs.****.edu.cn/cgi/sw?m=0';</script><script>alert(11071)</script>

服务器响应如下:

Content-Type: text/html;
charset=gb2312<script>document.cookie='m=0';
</script><script>alert(11071)</script>'</script>
<meta http-equiv='Refresh' content='0; url=/index.html'>

从服务器响应中发现没有对该 get 请求中的特殊字符过滤或转义, 就把构造过的链接提交到服务器后, 因此可以判定该链接中存在 XSS 漏洞.

通过上面的分析可以看出该 Web 应用漏洞扫描系统可以检测出 SQL 注入、跨站脚本(XSS)和上传文件等漏洞, 在运行过程中没有出现异常, 其功能可以满足设计的需求, 基本达到预期的设计目的.

5 结语

本文通过研究漏洞扫描系统的工作原理和分析常见应用漏洞的检测技术, 设计和实现了针对政府网站和中小企业网站的 Web 应用漏洞扫描系统, 该系统与同类型的软件相比, 更加有可扩展性, 可定制性, 不易遗漏漏洞, 而且可以同时对多个目标网站进行扫描. 下一步工作将在系统中加入对其他漏洞类型的检测, 并为系统增加攻击测试模块, 减少漏洞扫描中的误报率及提升漏洞覆盖率.

参考文献

1 360 互联网安全中心.2014 年中国网站安全报告. [2015-03-22]. http://webscan.360.cn, 2015-1-8.

2 CNCERT/CC. CNCERT 互联网安全威胁报告-2015 年 1 月. [2015-03-22]. http://www.cert.org.cn, 2015-3-6.

3 李必云,石俊萍.Web 攻击及安全防护技术研究.电脑知识与技术:学术交流,2009,5(11):8647–8649.

4 Takanen A, Demott JD, Miller C. Fuzzing for software security testing and quality assurance. Artech House, 2008.

5 都娟.基于模糊测试方法的 Web 应用安全性测试技术的研究及其工具实现[学位论文].上海:华东师范大学,2011.

6 齐富民,谢晓尧,徐洋.基于模糊理论的 Web 安全评估模型.计算机应用研究,2014,31(6):1883–1888.

7 OWASP T. Top 10–2013. The Open Web Application Security Project, 2013.

8 肖士锋.基于网络的漏洞扫描系统的设计与实现[学位论文].北京:北京邮电大学,2010.

9 Kemalis K, Tzouramanis T. SQL-IDS: a pecification-based approach for SQL-injection detection. Proc. of the 2008 ACM Symposium on Applied Computing. ACM, 2008: 2153–2158.

10 彭赓,范明钰.基于改进网络爬虫技术的 SQL 注入漏洞检测倡.计算机应用研究,2010,27(7).

11 刘奇旭,温涛,闻观行.Flash 跨站脚本漏洞挖掘技术研究.计算机研究与发展,2014,51(7):1624–1632.

12 周开东,魏理豪,王甜,等.远程文件包含漏洞分级检测工具研究.计算机应用与软件,2014,31(2):21–23.