

基于渗透测试的跨站脚本漏洞检测方法研究

王 强,蔡皖东,姚 烨

(西北工业大学 计算机学院,陕西 西安 710072)

摘 要:目前,跨站脚本漏洞已经成为互联网上最为严重的安全漏洞之一,文中针对跨站脚本漏洞的自动化检测问题,提出了一种基于渗透测试的检测方法。在向 Web 服务器提交攻击向量之前,对检测点使用合法输入进行探测,通过与 Web 服务器的一次交互可以排除一部分不包含跨站脚本漏洞的检测点,从而大量减少在分析检测点阶段与 Web 服务器交互的次数。另外,通过对获取的检测点进行去重,可有效防止对不同页面中相同检测点的重复检测。实验结果表明,该方法可有效提高跨站脚本漏洞的检测效率。

关键词:跨站脚本漏洞;渗透测试;Web 安全;cookie

中图分类号:TP393

文献标识码:A

文章编号:1673-629X(2013)03-0147-05

doi:10.3969/j.issn.1673-629X.2013.03.037

Research on Cross-site Scripting Vulnerability Detection Method Based on Penetration Testing

WANG Qiang, CAI Wan-dong, YAO Ye

(College of Computer, Northwestern Polytechnical University, Xi'an 710072, China)

Abstract: Cross-site scripting vulnerability has become one of the most serious vulnerabilities in the Internet. In order to auto-detect XSS vulnerabilities, propose a XSS detection method based on penetration testing. Submit simple string that will not filtered by Web server before submit attack vectors, interact with Web server once and then exclude part of the detection points that will not contain cross-site scripting vulnerabilities. When detect the detection points, the times of interaction with Web server will be reduced by this method. Delete the repeated detection points, prevent detecting the same detection points extracted from the Web pages. The experiment result shows that this method can detect XSS vulnerabilities effectively.

Key words: XSS vulnerability; penetration testing; Web security; cookie

0 引 言

根据 OWASP (Open Web Application Security Project) 统计的十大最具威胁的 Web 应用安全漏洞 (OWASP Top 10) 显示:跨站脚本 (Cross-Site Scripting, XSS) 漏洞的排名已经由 2004 年的第四位跃升至 2007 年的第一位,在 2010 年发布的最新排名仍高居第二位^[1],由此可见 XSS 漏洞的严重性。其主要的危害有:盗取用户 cookie、XSS 钓鱼、XSS 挂马、XSS 蠕虫等^[2]。

传统对于 XSS 漏洞的防范方法主要是在 Web 服务器端对浏览器提交的内容进行过滤或者编码^[3],经过处理后,返回到浏览器的攻击代码就无法被执行。

但是这种方法是粗粒度的,XSS 攻击可以通过一些方法来绕过这些检查,从而使 Web 服务器端的过滤失效。XSS 攻击主要是对 Web 应用的用户造成安全威胁,具有很强的隐蔽性,往往难以被发现。目前,XSS 漏洞检测工具对 XSS 漏洞的检测效果并不理想,存在较高的误报率和漏报率。

常用的对 XSS 漏洞的检测方法是直接向 Web 服务器提交攻击向量,并分析 Web 服务器的响应页面中是否包含能触发 XSS 漏洞的攻击脚本。为了使检测结果尽量准确,测试攻击脚本的数量较大,一般在 30 条以上。使用这种方法进行检测时,对不含 XSS 漏洞的检测点,需要向 Web 服务器提交所有的测试攻击脚本,而对一个检测点每提交一次攻击脚本都对应与 Web 服务器的一次交互。当所有攻击脚本全部被使用过时,如果还没有发现可触发 XSS 漏洞的攻击脚本,才认为该页面中不含 XSS 漏洞。

上述检测方法的不足之处是:对所有最终被判定为不含 XSS 漏洞的检测点,都向 Web 服务器提交了全

收稿日期:2012-07-11;修回日期:2012-10-19

基金项目:西北工业大学基础研究基金(JC201149);西北工业大学研究生创业种子基金(Z2012141)

作者简介:王 强(1988-),男,硕士研究生,研究方向为网络与信息安全;蔡皖东,教授,博士生导师,研究方向为网络与信息安全。

部的测试攻击脚本,而包含 XSS 漏洞的检测点在所有检测点中占极少数,这样,上述方法需要对大多数的检测点向 Web 服务器提交全部测试攻击脚本,与 Web 服务器交互次数过多。

本检测方法针对这一问题,采用的方法是在向 Web 服务器提交攻击向量之前,对检测点使用合法输入进行探测,通过与 Web 服务器的一次交互可以排除一部分一定不包含跨站脚本漏洞的检测点,从而大量减少在分析检测点阶段与 Web 服务器交互的次数,提高检测效率;另外,不同于一些使用网络爬虫对 XSS 漏洞进行检测的方法中只对爬虫抓取的 url 链接进行去重,而忽略了对获取的检测点进行去重。本方法通过对获取的检测点进行去重,可有效防止对相同检测点的重复检测。

1 相关概念

1.1 渗透测试

渗透测试作为一种渐进的并且逐步深入的测试方法,采用不影响业务系统正常运行的攻击方法进行的测试,不会对被测系统进行破坏活动。该测试方法通过模拟黑客的攻击方法和漏洞发现技术,对目标系统的安全状况进行逐步深入的探测,进而发现系统中存在的脆弱环节,是评估计算机网络系统安全的一种方法^[4]。渗透测试能够直观地让管理人员知道网络所面临的问题,这个过程包括对系统的弱点、技术缺陷或漏洞的主动分析,这种分析从攻击者可能存在的位置进行,并且从这个位置有条件主动利用安全漏洞。

1.2 XSS 漏洞

XSS 漏洞是指攻击者向 Web 页面插入恶意 HTML 代码,当正常用户浏览该页时,嵌入在 Web 页面里的恶意 HTML 代码被执行,从而达到攻击正常用户的目的^[5]。

XSS 漏洞可分为以下三类^[6]:

(1) 反射型 XSS 漏洞。

反射型 XSS 漏洞的恶意代码并不存在于 Web 服务器端,用户向 Web 服务器提交包含 XSS 攻击代码的信息之后,Web 服务器将恶意代码插入到返回的 HTML 代码中,恶意代码在客户端浏览器中被触发,导致反射型 XSS 漏洞。

(2) 存储型 XSS 漏洞。

用户向 Web 应用提交数据,提交的数据被存储在 Web 服务器端。如果这些数据中包含 XSS 代码,并且 Web 服务器没有对这些数据进行适当的过滤,当其他用户浏览该页面时,就会发生存储型 XSS 漏洞。

(3) 基于 DOM 的 XSS 漏洞。

对于基于 DOM 的 XSS 漏洞,恶意攻击者的 XSS

代码并不是由 Web 服务器嵌入到 HTML 页面中,而是在客户端接收到 HTML 代码之后,由浏览器将恶意代码嵌入到页面并执行^[7]。

2 检测方法

为了对 Web 站点进行自动化的 XSS 漏洞检测,需要对相应 Web 站点的全部页面进行分析,本检测方法在开源爬虫 larbin 的基础上实现。larbin 是著名的开源网络爬虫,可高效抓取网页,该爬虫只负责抓取网页,对网页的分析需要用户自己完成^[8]。检测方法主要包括对检测点的提取、调度和过滤,以及建立渗透策略和会话保持。

2.1 检测点提取

(1) 表单信息提取。

表单是用户与 Web 服务器交互的接口,表单获取用户的输入信息并向 Web 服务器提交。要模拟用户对表单的提交操作,需要提取表单信息。表单通过 POST 和 GET 方法向 Web 服务器提交数据,action 值记录了表单信息将向哪个页面提交。除此之外,还必须获得表单中各 input 标签里 type 值为空值、“text”、“textarea”、“hidden”、“password”和“submit”等所对应的 name 值和 value 值。

(2) url 检测点提取。

分析页面中的标签,将其中“<a>”、“<link>”、“<base>”、“<frame>”、“”、“<script>”等标签内所对应的 url 转化为绝对路径后,获取其中包含“? name=value”的 url 信息。

用获取的检测点信息构造 form_url 类的对象,form_url 类的数据成员如下:

```
struct InputElem
{
    char * type;
    char * name;
    char * value;
};

class form_url : public url
{
    char * parenturl;
    char * method;
    vector<struct InputElem > formInput;
};
```

InputElem 结构体对应检测点中的 type、name 和 value 值。url 类是 larbin 中的类,form_url 类是其子类。form_url 类中的 parenturl 是获取检测点时所对应的父页面的 url。检测点中的 action 信息由 url 类保存。获取的 url 检测点不同于表单,其本身不包含 type 和

method 信息,而其向 Web 服务器的提交方式为 GET,故在程序中将 url 检测点对应的 form_url 对象中的 type 值设置为“text”,method 值设为“GET”。

2.2 检测点调度

若不对获取的检测点信息进行合理有效的管理,庞大的检测点信息没有优先级别,将产生混乱。如果将其全部放入内存会消耗过多的内存资源,影响检测效率,本方法采用多级队列的方法对检测点进行调度。

检测点的调度流程如图 1 所示,检测点对象在检测点队列、检测点等待队列、NamedSiteList 和 IPSiteList 间进行调度。其中,检测点等待队列在外存中保存,NamedSiteList 中存放同域名的未经 dns 解析的检测点对象。IPSiteList 中存放经过 dns 解析后已经获得 IP 地址的检测点对象。由于 form_url 类继承于 url 类,可直接使用 larbin 中的 NamedSiteList 和 IPSiteList 模块。检测点对象经过去重后进入检测点队列中,如果 NamedSiteList 中对应站点的队列已满,则进入检测点等待队列。NamedSiteList 中的检测点经一次 dns 解析即可获取该站点的目标 IP 地址,实现批量 dns 解析。服务器请求队列中存放的是具备请求条件的检测点,可直接向 Web 服务器针对该检测点提出检测请求。

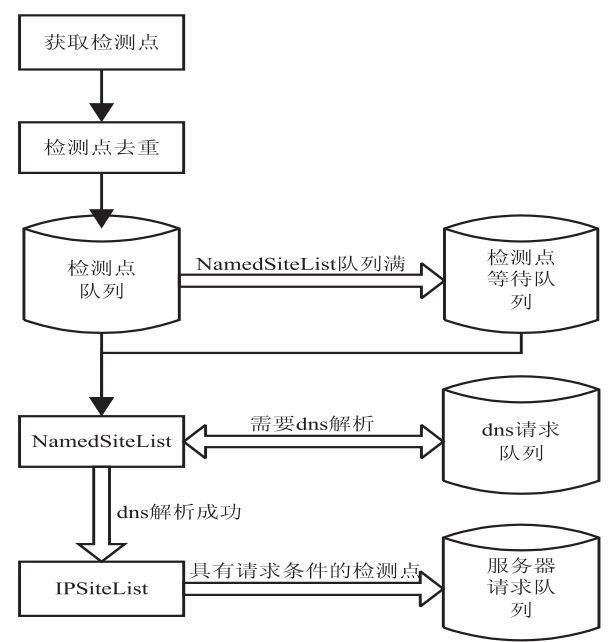


图 1 检测点调度流程

2.3 检测点过滤

同一个 Web 站点的不同页面中往往会包含大量相同的检测点,如一些站点会包含“站内搜索”功能,而该站点的大部分页面都会包含“站内搜索”功能所对应的表单。

在使用爬虫检测 XSS 漏洞的文献[9,10]中,只提到了对爬虫所提取的 url 链接去重,而没有涉及到对于检测点的去重。重复的检测点在所有获取的检测点

中占很大比重,如果对大量相同的检测点进行重复分析,会严重影响检测效率,为了避免这一问题,需要对重复的检测点信息进行过滤。

由于 action 值可唯一地标识一个 form_url 对象,且布隆过滤器^[11]具有高效性和占用内存资源少的优势,故使用该算法对检测点进行去重:对每个 action 值,使用不同的 hash 函数求出一组 hash 值,将 hash 值映射到 HashTable 中相应的比特位,如果 action 值的每一个 hash 值对应的比特位在 HashTable 中都被置为“1”,表示该检测点已经存在,则将其丢弃。否则,将 HashTable 中该 action 值所对应的 hash 值都置为“1”,并把检测点放入检测点队列。

2.4 渗透策略

2.4.1 检测原理

针对已有检测方法与 Web 服务器交互次数过多的问题,本方法在对检测点进行分析时,首先向 Web 服务器提交合法字符串,如“testxss”。该字符串需满足如下要求:不包含可用于构造 XSS 攻击代码的字符或字符串,如:“<”“/”“=”“>”“.”“”“script”“javascript”“alert”“iframe”等;为防止服务器端因长度过长而截断测试字符串,字符串长度也不应大于 8 个字符。然后分析响应的页面中是否原样返回了提交的字符串“testxss”。如果没有,则可以断定该检测点不包含 XSS 漏洞,程序可以直接放弃后续对该检测点的其他 XSS 检测工作。

之所以做这样的改进,是因为 Web 服务器端可能会对测试攻击脚本中的一些字符或字符串做编码或者丢弃的操作,使返回的页面无法与预期检测的字符串相匹配。但是服务器端的过滤方法不会对合法字符串进行过滤,如果合法字符串没有在响应页面中被返回,则提交的测试攻击脚本也一定不会被返回,再继续使用其他攻击脚本进行检测也不会与期望的检测字符串相匹配。即本方法可对一部分一定不包含 XSS 漏洞的检测点进行排除,可减少与服务交互的次数而不会对检测准确性产生影响。

2.4.2 渗透步骤

与 Web 服务器进行交互时,对检测点中 type 值为空值、“text”和“textarea”所对应的 value 值根据下述检测步骤填充,其他 type 值所对应的 value 值按从页面中获取的值填充。并根据 method 方法向 Web 服务器发送相应的请求。由于构造攻击向量的测试攻击脚本的质量直接决定了最终检测结果的准确性,本方法采用 OWASP2.0 Guide 附录部分所引用的 XSS 漏洞定位字符串^[12]作为测试攻击脚本。

渗透流程如图 2 所示。

本渗透策略对检测点的渗透步骤描述如下:

- (1)从服务器请求队列中取出一条检测点。
- (2)向检测点对应的字段提交合法字符串如“testxss”。

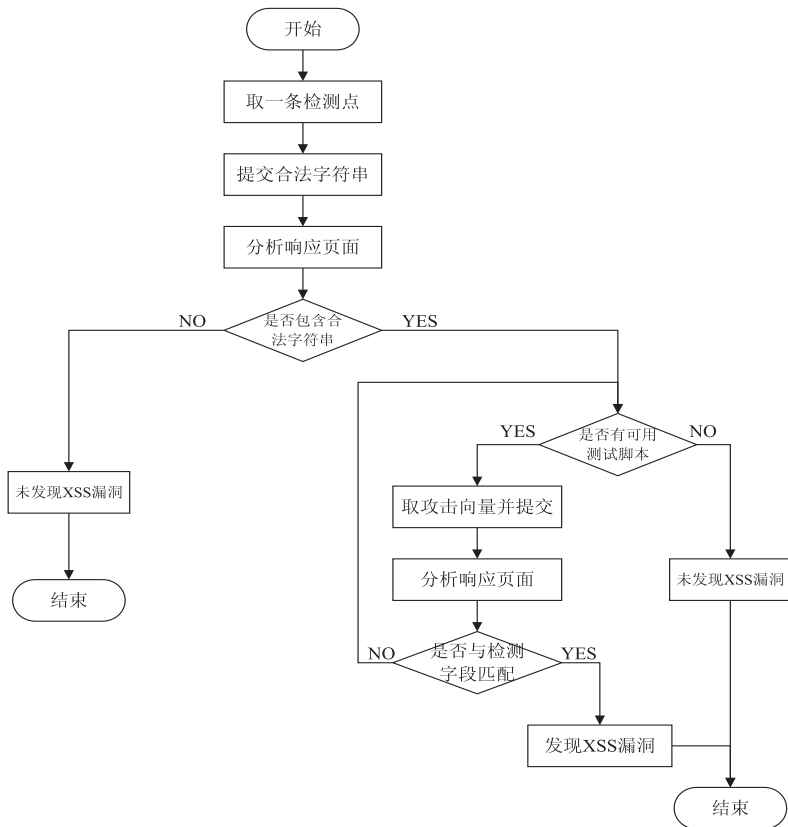


图2 渗透流程

(3)分析响应页面中是否包含“testxss”,如果包含,则继续执行步骤(4)。否则,本检测点分析结束,未发现 XSS 漏洞。

(4)从测试攻击脚本队列中取出一条攻击脚本,向检测点对应字段填充并提交。

(5)分析响应的页面是否与预期的检测字符串相匹配,如果匹配,则认为发现 XSS 漏洞,记录检测点信息。否则,如果还有攻击脚本未被分析,则继续执行步骤(4);如果所有攻击脚本都已经被分析,则本检测点分析结束,未发现 XSS 漏洞。

2.5 会话保持

会话是指用户在一定时间内发出的一系列请求和 Web 服务器返回的响应。HTTP 协议作为一种无状态协议,即两个 HTTP 请求/响应对之间是相互独立的。在会话的过程中需保持会话状态,以便让 Web 服务器对不同的会话进行区分。Web 服务器通过会话标识来区分不同的用户,而会话标识又需要借助 cookie 技术在 Web 服务器与浏览器之间来回传递^[13]。

在检测的过程中要与 Web 服务器进行多次交互,在与 Web 服务器交互时需要保持会话状态,尤其是对一些需要登录信息的站点的访问,否则在访问页面

时,Web 服务器会拒绝访问或对页面重定向。本方法中保存 Web 服务器返回的 cookie 信息,当再次请求时将相应的 cookie 信息添加到 HTTP 请求头中,以达到会话保持的目的。

3 实验结果与分析

3.1 实验环境

本实验运行在 Linux 操作系统,1.93G 内存,Intel Core2 2.53GHz 处理器的主机上,并接入互联网,测试本检测方法的效率和有效性。

3.2 实验结果

为了验证本方法的效率,用本方法对起始 url 为 www.nwpu.edu.cn 的网页进行检测。为使测得的数据更具一般性,将程序设置为允许抓取外部链接,即抓取的 url 链接与起始 url 可以不在同一个域名中。本实验的目的是验证原样返回合法字符串的检测点占有被测检测点的比重,进而说明本方法可提高检测效率。

经过 2 小时的检测,共获取 url 85554 个,成功获取页面 32111 个,每 8 秒钟采集一次 AnalysisNum 值和 ReturnedNum 值,共采集数据 900 组。其中,AnalysisNum 和 ReturnedNum 的含义如下:

AnalysisNum:所有被测试的检测点的数量。

ReturnedNum:对应的响应页面中包含合法字符串的检测点的数量。

每一组数据的采样值如图 3 所示。

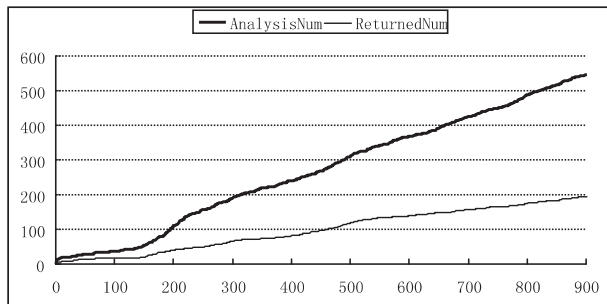


图3 采样值

在上述采样过程中,采样结果分析如图 4 所示。

由图 4 可知,在分析样本达到一定规模时,ReturnedNum 与 AnalysisNum 的比值趋于稳定,即能原样返回合法字符串的检测点约占所有已测检测点的 30%~40%,后续的检测工作只需对这不足 40%的检测点进行进一步的测试攻击脚本的提交和检测。采用本方法可对约 60%的检测点通过与 Web 服务器的一

次交互而直接排除其包含 XSS 漏洞的可能性,而之前的方法需要与 Web 服务器交互数次,所以此方法可大大减少在检测点分析环节与 Web 服务器交互的次数,提高了检测效率。

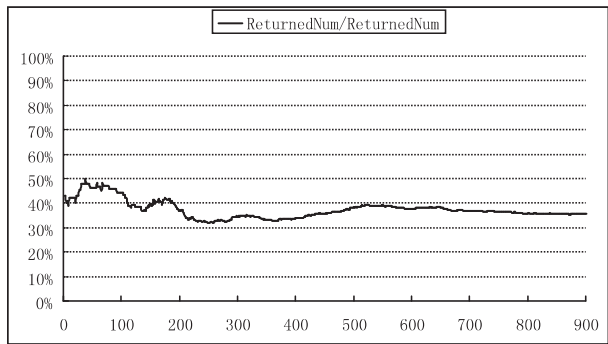


图 4 采样结果分析

为了测试本方法的有效性,对一些 Web 站点进行检测。将程序设置为不允许抓取外部链接,只针对起始 url 域名所对应的页面进行检测。并将结果与 Paros3.2.13 和 Acunetix Web Vulnerability Scanner 7 比较。XSS 漏洞检测结果如表 1 所示,对本方法所检测出的漏洞进一步分析发现这 9 个漏洞均为 XSS 漏洞。从检测结果可以发现,本方法可有效检测 XSS 漏洞。

表 1 XSS 漏洞检测结果

网站	本方法	Paros3.2.13	Acunetix Web Vulnerability Scanner 7
某高校网站	2	0	1
某政府网站	2	2	2
某研究生网站	4	2	4
某企业网站	1	0	0

通过使用 wireshark 软件抓包分析,发现 Paros3.2.13 中测试攻击脚本过少影响了其对 XSS 漏洞的查全率。Acunetix Web Vulnerability Scanner 7 对获取的检测点的父页面的 url 在规定的域名中而检测点的 action 值不在该域名的检测点不予分析,导致一些包含 XSS 漏洞的检测点无法被检测,造成漏报。

4 结束语

文中方法可有效检测 XSS 漏洞,在开源爬虫 larbin 的基础上实现了对 Web 站点的自动化检测。在对检测点的检测环节,通过与 Web 服务器的一次交互可直接排除一部分不包含 XSS 漏洞的检测点,实验数据表明这部分检测点约占全部被测检测点的 60%,提高了检测效率的同时不会对检测的准确性造成影响。

另外,本方法对获取的检测点也进行去重,而不只是对获取的 url 链接去重,避免了对相同检测点的重复分析。

下一步研究的重点是对需要登录信息的站点进行检测,一些站点如人人网、一些内部论坛以及基于 Web 的管理系统等,只有在登陆后才能进行进一步的访问,否则会重定向到登录页面。本方法已经通过对 cookie 信息的保存达到了保持会话状态的目的,但是除此之外还需要解决如登录时验证码的分析,以及对动态页面跳转的分析等问题。

参考文献:

[1] OWASP. Category:OWASP Top Ten Project[EB/OL]. [2012-01-18]. http://owasp.com/index.php/Category:OWASP_Top_Ten_Project.

[2] 邱永杰,姜建国. 跨站脚本攻击与防御技术研究[D]. 北京:北京交通大学,2010.

[3] 王夏莉,张玉清. 一种基于行为的 XSS 客户端防范方法[J]. 中国科学院研究生院学报,2011,28(5):668-675.

[4] 邢 斌,高 岭,孙 骞,等. 一种自动化的渗透测试系统的设计与实现[J]. 计算机应用研究,2010,27(4):1384-1387.

[5] 郝永清. 黑客 Web 脚本攻击与防御技术核心剖析[M]. 北京:科学出版社,2010.

[6] OWASP. Cross-site Scripting(XSS)[EB/OL]. [2011-11-17]. [https://www.owasp.org/index.php/Cross-site_Scripting_\(XSS\)](https://www.owasp.org/index.php/Cross-site_Scripting_(XSS)).

[7] OWASP. DOM Based XSS[EB/OL]. [2011-11-17]. http://www.owasp.com/index.php/DOM_Based_XSS.

[8] 罗 浩,魏祖宽. 基于 CLucene 和 Larkin 的企业搜索引擎的研究与实现[D]. 成都:电子科技大学,2010.

[9] 沈寿忠,张玉清. 基于爬虫的 XSS 漏洞检测工具设计与实现[J]. 计算机工程,2009,35(21):151-154.

[10] 彭 亮,卓新建,黄 玮,等. 基于网络爬虫的 XSS 漏洞扫描系统的设计与实现[C]//中国智慧城市论坛论文集. 天津:中国学术期刊电子出版社,2011.

[11] Dharmapurikar S, Krishnamurthy P, Sproull T, et al. Deep packet inspection using parallel bloom filters[J]. IEEE Micro,2004,24(1):52-61.

[12] RSnake. XSS (Cross Site Scripting) Cheat Sheet[EB/OL]. [2012-01-20]. <http://ha.ckers.org/xss.html>.

[13] Hope P, Waltber B. Web 安全测试[M]. 傅 鑫,译. 北京:清华大学出版社,2010.

基于渗透测试的跨站脚本漏洞检测方法研究

作者: [王强](#), [蔡皖东](#), [姚烨](#)
作者单位: [西北工业大学 计算机学院, 陕西 西安710072](#)
刊名: [计算机技术与发展](#)
英文刊名: [Computer Technology and Development](#)
年, 卷(期): 2013(3)

本文链接: http://d.g.wanfangdata.com.cn/Periodical_wjfz201303039.aspx