文章编号：2096-1618(2019)02-0131-07

基于业务逻辑思想的异常检测研究

蒋梦丹， 林宏刚， 曹鹤鸣
(成都信息工程大学网络空间安全学院，四川 成都，610225)

摘要：随着互联网的普及，Web站点承载的流量越来越多，为保证能够安全且高效地提供信息或服务，将业务逻辑思想与异常检测相结合，提出了一种以Web用户访问行为所产生的流量数据作为基础的异常检测方案。该方案为部署在重要节点上的Web服务器而设计，通过获取公开发布在网络上的Web站点的源代码来获取页面之间的链接关系，构建网站的拓扑结构，来学习符合正常业务逻辑跳转的用户访问轨迹集合，然后从访问产生的流量数据中提取出不同用户的行为轨迹，结合方案中提出的算法来判断用户的访问是否存在异常。最后在实际环境下对该算法的有效性进行了测试与验证，实验表明，该异常检测方法能够发现用户对站点无规律的大量异常访问、对站点进行的SQL注入尝试或语义URL攻击等。

关 键 词：业务逻辑；异常检测；流量分析；网站结构；访问信息
中图分类号：TP393.08 **文献标志码**：A
doi：10.16836/j.cnki.jcuit.2019.02.006

0 引言

随着Internet的迅速发展，网络在日常生活中被广泛应用。越来越多的网络攻击事件[1]也随之而来，威胁着信息安全。在过去几年的时间里，数以万计的网络终端感染了网络病毒，影响范围较大的网络事件也不断发生，且次数呈增长态势。早期为了应对不断出现的安全威胁，构建可信网络，提出了“可管、可控、可信”的网络安全需求[2]，入侵检测系统应运而生。通常，网络入侵检测系统分为误用检测和异常检测，能够提供实时的入侵检测，并对异常事件做出告警。入侵检测的提出使得许多网络威胁引发的网络故障和性能问题得到了解决，与此同时信息技术更大程度地融入社会生活中，网络站点的数量随着人们的需求呈增长趋势，根据2018年1月发布的第41次《中国互联网络发展状况统计报告》[3]中提供的信息，截至2017年12月，中国网站数量为533万个，年增长率为10.6%，互联网站点成为获取信息和服务的重要渠道之一。大量的Web应用程序利用浏览器作为服务器端，以HTTP协议或HTTPS协议作为应用的标准通信协议，与用户进行交互。统计表明，SQL注入、远程文件包含、XSS攻击等基于HTTP协议的网络攻击所占的比例曾占漏洞攻击总数的一半[4]，针对这类攻击的检测算法被大量提出[5]，大量的服务网站或信息网站的后台数据库成为被攻击的核心目标。为了保证它们能够安全且高效地运行，需要对站点进行监控和保护以发现异常的访问信息。

收稿日期：2018-09-27

用户在访问Web站点时，网络中会产生流量数据，这些网络流量数据描述了用户的网络行为，为网络异常检测提供了许多重要信息。谢逸等[6]提出一种基于Web用户访问行为的异常检测方案，首先根据Web页面的超文本链接特征和网络中各级Web代理对用户请求的响应作用，用隐马尔可夫模型来描述服务器端观测到的对Web页面进行访问的用户行为，然后将该行为模型进行特征提取，与正常用户访问行为特征进行比对，得出用户每次访问的异常程度。该方法需要对计算异常程度的参数进行推导和调整，可用于检测应用层分布式拒绝服务攻击。针对其他类型的Web攻击，温凯等[7]提出了一种自适应地建立正常行为模型对Web攻击进行异常检测的方法，利用Request-URL的结构特征来描述Web请求的类型，用隐马尔可夫模型识别URL结构特征的方法对样本集进行分类，搜集样本的各属性构造离散性函数，将样本子集的离散程度作为识别正常行为集的标准来识别Web攻击请求。之后有学者进一步对Web攻击检测进行了研究，包括对隐马尔可夫模型的训练过程进行改进，结合正常用户访问特征来检测基于URL的Web攻击[8]。以上所述的大多主流异常检测方法都需大量的数据集来对参数进行训练，以提高算法在实际环境中的适应性和正确率。在不存在环境干扰条件的情况下，国内外学者们提出的异常检测方法能够有效检测到网络异常情况。

为了使异常检测算法能够在抵抗噪声干扰的情况下保证运行效率和准确率，在这个方向上进行了研究。刘泽宇等[9]提出了基于用户行为轨迹的防御模型，把用

户的访问行为抽象为 Web 行为轨迹,根据攻击请求的生成方式与用户访问 Web 页面的行为特征,定义异常因素,然后计算用户正常访问网站时和攻击访问时产生的异常因素的偏离值,来检测针对 Web 网站的分布式拒绝服务攻击。闫伟等[10]提取网络流量的原始数据,并对原始数据进行小波阈值去噪处理,消除干扰因素的影响,然后采用时间序列分析法挖掘网络流量数据之间的变化关系,建立网络流量异常检测模型。廖鹏等[11]提出了一种基于用户序列的异常检测方法,对本地抓取的网络数据进行预处理后基于时间生成每个用户的访问序列,通过每个用户的行为序列计算用户之间的行为相似度和相关系数,比较相关系数进行异常行为检测,寻找用户异常行为。以上方法都是通过建立一种逻辑模式来达到异常检测的目的,在噪声干扰上具有明显的优势,使用资源少,运行效率高。

通常,在重要的主机设备或服务器中提供服务的业务系统都有清晰的业务逻辑,其中可能包括访问的发起方、应答方、使用的协议和端口等。一般地,业务被访问的时间、访问者、访问的数据量等都存在一定的规律性,这也为基于业务逻辑思想来进行攻击检测提供了基础[12-13]。将异常检测方案与业务逻辑思想相结合[14],构建了一种基于业务逻辑思想的异常检测方法,对重要的 Web 站点进行维护,用于发现站点访问信息中的异常情况。通过构建检测正常的业务逻辑,对搜集到的站点信息进行分析,可以发现部分基于 Web 的攻击。

1 基于业务逻辑思想的异常检测方案

Web 站点常用于展示信息或提供服务,在用户进入站点系统后,对站点页面的操作通过浏览器执行,各个页面之间的跳转与到达关系存在规律性和确定性,这种正常访问的规律性能够从网络站点的拓扑结构中体现。因此在浏览器界面中,各个页面的链接情况能够一定程度上反映系统的业务逻辑,网络站点拓扑图中的单向访问路径的所有子集构成了符合正常用户访问逻辑的路径集合。

基于上述,提出基于业务逻辑思想的异常检测方案:通常每个网站从搭建开始,就有属于自己的网站结构。一个完整可用的网络站点可能由成百上千个页面构成,且页面之间都有着直接或间接的链接关系。用户在使用浏览器访问站点的过程中,页面间的跳转通常通过鼠标的点击操作实现,站内页面之间存在的全部链接关系整合后可以反映出整个网站的拓扑结构,且这些链接信息可被用户以信息搜集的方式获取,将这类信息进行整理及构建,可形成网站地图,帮助描绘出正常用户的访问路径。再通过抓取网络流量数据包获取用户对站点的访问请求信息,从而建构出不同用户的访问序列,最后设定异常检测规则,从用户访问序列中找出违反正常业务访问逻辑的记录,即为可疑的访问。

1.1 用户访问行为的建模

在基于 Web 的业务系统中,要描述用户对业务系统的访问行为,需要对用户访问行为建立模型。建立的模型为一个三维的数据集,其中划分的 3 个维度分别是时间维度、空间维度和行为维度。时间维度,用户的某次访问行为产生的详细时间信息;空间维度,访问该 Web 业务系统的用户所使用的 IP 地址所构成的地址空间的集合;行为维度,即用户对 Web 站点的访问方式以及访问的 URL 地址等刻画用户行为的信息。

1.2 用户行为数据的提取

为获取全面的用户访问信息,以构建用户访问序列,在 Web 服务器的核心交换机设置镜像,通过镜像数据获取网络流量数据包。用户行为数据提取的完整过程如下:

(1)数据包的嗅探。从核心交换机数据中获取所有的原始数据包;

(2)数据包的解码。将原始数据包依次进行数据链路层、网络层、传输层的解码,得到应用层数据信息;

(3)数据包的过滤。在获取应用层数据后,过滤出使用 HTTP 协议的数据包,筛选出其中使用 GET 方法以及 POST 方法的数据包;

(4)行为数据的提取。分析过滤出的数据包的头部信息和负载信息,提取用户访问信息。

1.3 系统模块的构成

为实现这个异常检测系统,将系统划分为 3 个模块:网站信息搜集模块、访问信息获取模块、基于业务逻辑思想的异常发现模块。

3 个系统主要模块的关系如图 1 所示。网站信息搜集模块用于搜集站点信息以构建网站地图;访问信息获取模块抓取用户访问记录用于构建用户访问序列;基于业务逻辑的攻击检测模块对用户访问序列进行异常检测,判定访问是否存在异常情况。

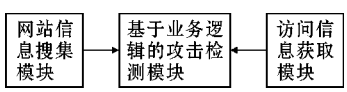

图 1 系统模块关系图

网站信息搜集模块和访问信息获取模块共同为攻击检测模块提供关键信息,支撑基于业务逻辑的攻击检测模块的正常运行,是该异常检测的两个基础信息

模块。而基于业务逻辑的攻击检测模块是该系统的主要功能模块。

该异常检测系统的主要处理流程如图 2 所示。

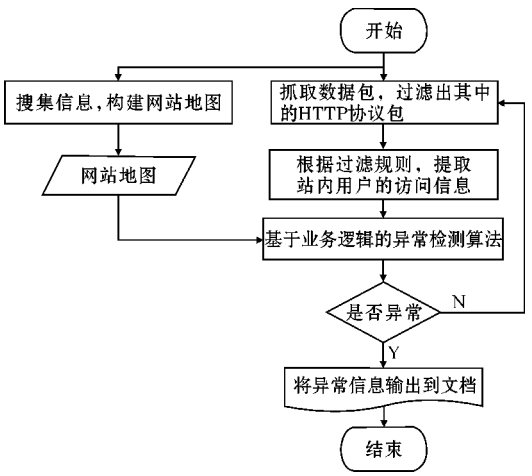

图 2　系统执行流程图

1.3.1　网络信息搜集模块

网站信息搜集模块主要用于获取指定站点的信息,由网络爬虫实现。网络爬虫又被称为网页蜘蛛、网络机器人,在 FOAF 社区中间,更常被称为网页追逐者,是一种按照一定规则自动抓取万维网信息的程序或者脚本,已被广泛应用于互联网领域。搜索引擎使用网络爬虫抓取 Web 网页、文档甚至图片、音频、视频等资源,通过相应的索引技术组织这些信息,提供给搜索用户进行查询。

此模块的网络爬虫在搜集整个网站包含的链接时,通过对指定页面发送请求来获取该页面的源代码,之后提取出在网页源码中包含的链接,做内链过滤和去重处理后,再对新获取到的链接做同样的递归处理,在获取新链接的同时,将网站包含的所有链接信息存入数据库中,并为其中每个链接赋予不同的 ID 号。在完成所有 URL 的获取和存储后,为数据库表中保存的每一个链接建立一个新的数据库表,用于存储每个链接能够直接跳转到的网页地址(此处不包含浏览时可以通过后退操作到达的链接,这种情况在构建网站地图的过程中再进行处理)。

1.3.2　访问信息获取模块

访问信息获取模块是抓取数据包并存储为 pcap 文件,通过处理该文件对数据包依次进行判断,数据包解析后若不属于 HTTP 协议的 GET 包或者 POST 包,则跳过该数据包,处理下一个;若属于,则保存该数据包信息,并且将这些信息以文本形式存入文件中,写入访问信息的文件放入指定文件夹,等待后续处理。在提取数据包有用信息时,保留了数据包中的时间信息、源 IP、源端口、目的 IP、目的端口、负载信息等。

访问信息获取模块除了处理 pcap 包外,还有一个独立的线程与系统界面进行通信,用户在更新网站结构后,通过该线程发送触发网站信息搜集模块的命令及更新所需的相关参数,用于重新获取网站结构信息。

1.3.3　基于业务逻辑思想的异常发现模块

基于业务逻辑思想的异常发现模块是由网站信息搜集模块和访问信息获取模块支撑的,是系统实现异常检测的主要模块。该模块有 3 大主要功能:生成指定站点的网站地图;从 HTTP 包中提取出有效访问信息,其中包括每次访问站点的用户 IP 地址、访问时间、访问的 URL 地址;实现基于业务逻辑的异常检测算法,处理访问信息获取模块输出的含 HTTP 数据包文件,提取其中有效信息并存入各个用户的数据库,用算法结合数据库存储的信息,检测用户访问信息是否符合定义的业务逻辑规则。

网站地图的生成。该模块通过查询网站信息搜集模块中获取的信息,生成网站地图。网站地图的生成逻辑如下:在网站信息搜集模块,将一些可用于生成网站地图的信息存储在本地数据库中,数据库包含了一个数据库表 allpages,保存站点下所有可达链接的 URL 地址以及每个地址的唯一 ID 号码,n 个(n 为表 allpages 中站点的个数)数据库表 pages1 ~ pages$n$,pages$m$ 保存了 ID 号码为 $m(1 \leqslant m \leqslant n)$ 的页面可直接跳转的页面的 URL 以及这些页面对应的 ID 号码。

假设要搜集信息的站点为 A,A 的完整地址为 http://www.xxx.com/,且这个站点下除了本页面还有 5 个其他链接的页面,分别为 B/C/D/E/F。

假设数据库中存储的信息如表 1 至表 7 所示。

表 1　allpages 表

ID	URL
1	A
2	B
3	C
4	D
5	E
6	F

表 2　pages1 表

URL	ID
A	1
B	2
C	3

表 3　pages2 表

URL	ID
B	2
A	1
E	3

表 4 pages3 表

URL	ID
C	3
A	1
B	2
D	4

表 5 pages4 表

URL	ID
D	4
F	6

表 6 pages5 表

URL	ID
E	3

表 7 pages6 表

URL	ID
F	6

在不考虑浏览器特性的情况下,根据以上信息可以构建出当前站点的网站地图(假定所有页面都处于一个平衡状态,即所有页面都属于同一个层次,没有主次之分),可视化的绘图结果如图 3 所示。

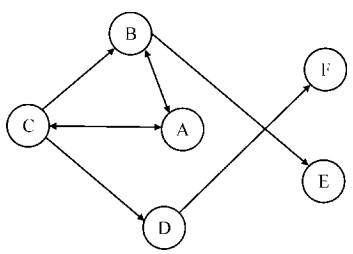

图 3 网站地图 1

以上所示的网站地图可以用一个 6×6 的矩阵描述。

矩阵当中为 0 的位置表示两个页面之间不存在直接链接关系,若矩阵中为 1 则表示两个页面之间存在直接链接关系。

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

但是由于浏览器存在一些特性:可由当前页面通过后退键返回到上一个页面;可以通过刷新键再次访问当前页面。因此在最终构建的网站地图中,在原有网站地图中具有任意链接的两个页面都是互相可达的,在以上例子中,构建的属于该站点的网站地图如图 4所示。

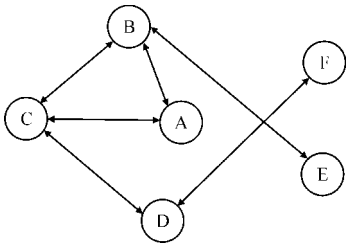

图 4 网站地图 2

同样也可以用一个 6×6 的矩阵描述该网站地图。

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \end{bmatrix}$$

用于描述网站地图的矩阵由于考虑了浏览器的操作特性,所以在此情况下生成的矩阵一定都为对称矩阵。

HTTP 包访问信息的提取。从 pcap 文件中提取 HTTP 协议的 GET 包和 POST 包后,将每个数据包的主要信息重新写入指定文件中,在异常检测模块对这些信息进行处理。

每一个 HTTP 包被提取出的信息是每个数据包的访问时间和访问者的 IP 以及负载信息。详细信息如图 5 所示。

```
#####Wed Mar 07 15:50:29 2018
#IP#192.168.1.102
GET /nyist327/article/details/44097199 HTTP/1.1
Host: blog.csdn.net
Connection: keep-alive
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like
Accept-Encoding: gzip, deflate, sdch
Accept-Language: zh-CN,zh;q=0.8
Cookie: uuid_tt_dd=10_19968155640-1517298158511-757808; __yadk_uid=ymooboORM9a5z
```

图 5 HTTP 包信息

在从 HTTP 包中提取用户对站点的访问序列时,不同用户是以不同 IP 地址来区分的。

在提取访问序列的过程中,系统在数据库中维护了一个 IP 地址的列表,这个表中存储的是已存在访问信息的 IP 地址的集合以及存储与之对应的访问信息的数据库表的编号。

在获取一条新访问信息时,系统从 HTTP 包信息中提取访问者的 IP 地址,查询该 IP 地址是否已存在于地址列表中;若不存在则用新的编号将此 IP 地址存入列表中,同时以此编号创建一个新的数据库表用于后续存储该 IP 地址产生的访问信息。

在 HTTP 包的负载信息中,从 Host 项中可获取 URL 地址中的前一部分,而完整的 URL 由 Host 项中的地址和 GET/POST 方法后的信息共同构成。

以图 5 为例,第三行的信息 GET /nyist327/article/details/44097199 HTTP/1.1 以及第四行的信息

Host:blog. csdn. net,结合这两行信息可以提取出本次访问的完整 URL 地址,在此例中完整的 URL 地址为 http:// blog. csdn. net/ nyist327/article/details/44097199。

判定异常访问的规则。针对基于业务逻辑思想的 Web 攻击,定义 Web 站点访问信息的异常检测规则如下。

在不同用户的访问信息数据库表中,每个访问信息是按照时间顺序排列的,以下规则针对同一用户产生的访问信息:

(1)如果提取出的用户访问的指定域名下的 URL 在网站链接的数据库中未出现,则认为此次访问存在异常;

(2)如果用户两次访问间隔的时间在一分钟内,则认为本次访问与前一次访问属于同一个序列,查询网站地图验证被访问的两个 URL 之间是否存在跳转关系,若存在则访问合法,反之认为本次访问存在异常;

(3) 如果用户两次访问间隔的时间超过一分钟,则认为本次访问独立于前一次访问。

异常检测过程的主要流程如图 6 所示。

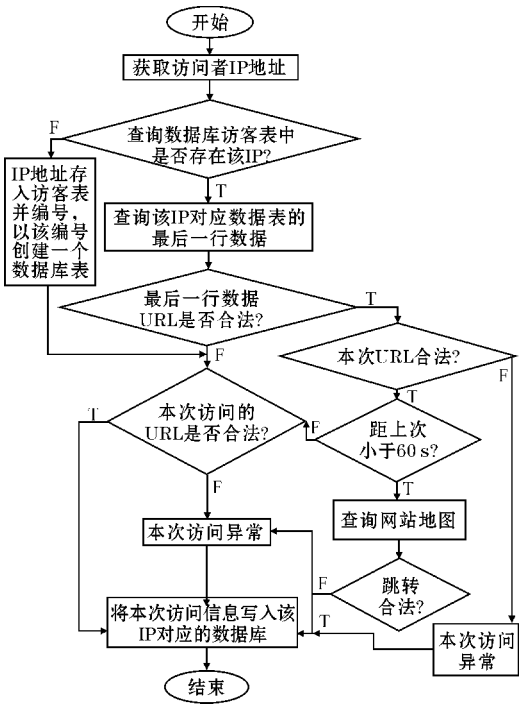

图 6　异常检测算法流程图

2　异常检测方案的测试

要对提出的异常检测方法进行验证,首先要获取实验数据。选择两个 Web 站点用于实际网络中的测试。

一个站点是无须认证就可以访问的站点,另一个是需要用户登录后才能够访问的站点。然后,在实验环境中模拟普通用户访问站点的操作,同时使用攻击工具尝试对实验站点进行后台扫描、SQL 注入操作等,同时抓取网络流量数据包,并保存为 pcap 文件作为实验数据。

在测试界面中,针对无须认证的 Web 站点,用户需要提供 4 个参数:基于业务逻辑思想的异常检测系统运行的 IP 地址、端口号、待监控站点的 URL 地址以及该站点下所有页面的 URL 共有的关键字(如主要的域名信息)。

而需要认证才能够访问完整信息的 Web 站点除了以上的 4 个参数外,还需要站点用于登录的用户名及其对应的密码信息、提交登录表单的网址、表单中填写用户名和密码的字段名称这 5 个参数。

2.1　针对无须认证的站点进行的测试

首先,在操作界面填写参数信息并启动系统,在界面中选择禁用用户名/密码的形式,然后根据测试方案填入所需的参数:服务器 IP 地址、服务器端口、目标网站 URL、关键字,如图 7、8 所示。

图 7　测试参数配置图 1

图 8　测试参数配置图 2

然后,在系统启动后,在真实环境中模拟攻击数据,并将抓取的数据包文件送入异常检测系统进行处理,一段时间后结果会显示在界面中,如图 9、10 所示。

图 9　测试结果 1

图 10 测试结果 2

2.2 针对需认证的站点进行的测试

与上一个测试类似,在操作界面填写参数信息并启动系统,如图 11、12 所示。

图 11 测试参数配置图 3

图 12 测试参数配置图 4

系统启动后,在真实环境中模拟攻击数据并将抓取的数据包文件送入异常检测系统进行处理。

一段时间后,结果会显示在界面中,如图 13 所示。

图 13 测试结果 3

2.3 测试结果说明

在实际环境中部署基于业务逻辑思想的异常检测系统后,对该异常检测算法的有效性进行了测试,结果显示该方法能够有效地保护使用 HTTP 协议的站点,在站点被未授权的网络爬虫搜集信息时、受到 SQL 注入攻击扫描时产生的访问信息都会被判定为异常的访问,从而在界面中被显示,由此得出结论,违反设定的正常用户访问轨迹的访问请求都能够被这种异常检测方法发现。

2.4 异常检测算法的效率

通过在实际环境中对基于业务逻辑的异常算法的各个模块进行测试,对该异常检测算法的效率进行了总结。在网站信息搜集模块,系统运行的时间复杂度为 $O(n)$,其中 n 为站点中页面的个数,该时间复杂度受到页面个数 n 和站点层次结构的影响;在访问信息获取模块,处理每个数据包的算法复杂度和时间复杂度都为 $O(1)$;而系统的核心模块即基于业务逻辑思想的攻击检测模块中,异常检测算法处理每个 HTTP 访问数据包的时间复杂度都也为 $O(1)$,得到了较好的效率。

3 结束语

提出的基于业务逻辑思想的异常检测算法,相比于之前的主流异常检测方法,优点在于不需要大量的训练数据且准确性不依赖于参数的设定,在受保护站点的结构更新后,只需要用户再次填写必要的信息,经过一段时间重新搜集信息并对网站地图进行更新后,就能够继续进行访问信息的异常检测。

但该算法仍存在一定的误报率,对于浏览器访问时产生的对界面控件或信息的请求若无法通过域名和关键字过滤掉,可能会被误判定为异常,除此之外用户产生的正常访问信息一般不会产生误报。另外一个问题是,因为 HTTPS 协议的信息是加密处理的,无法通过抓取流量数据获取访问该站点的请求信息,因此若该算法要部署在使用 HTTPS 协议的站点中,Web 站点必须自己设置并维护日志信息并通过日志信息还原来自同一 IP 地址的用户访问轨迹。

系统无法在使用动态验证码进行交互式用户登录的网络站点下使用,可在未使用验证码进行交互式登录的站点下达到良好的异常检测效果;在算法中使用站点提供的可用的用户名和密码信息获取登录状态下的 cookies 用于保持登录状态获取站点结构,因此在禁

用 cookies 的站点下也会受到限制。

提出异常检测算法有较好的抗噪性,能够在真实环境测试中体现出较高的检测率,但是目前不可避免地存在着误报率、检测条件限制等技术缺陷,为了解决这些问题,未来将对此进行进一步的研究。

参考文献:

[1] 熊芳芳. 浅谈计算机网络安全问题及其对策[J]. 电子世界,2012,8(22):139-140.

[2] 周光涛,王志军. 新一代高可信网络架构研究[C]. 2009 北京青年通信科技论坛,2011.

[3] 中国互联网信息中心. 第 41 次中国互联网络发展状况统计报告[OL]. URL:[2018-03-01]. http://cnnic. cn/hlwfzyj/hlwxzbg/hlwtjbg/201803/P020180305409870339136. pdf,2018.

[4] Christey S. Vulnerability type distributions in cve [EB/OL]. http://cwe. mitre. org/documents/vuln-trends. html,2007,05,22.

[5] Kar D,Panigrahi S, Sundararajan S. SQLiGoT:Detecting SQL injection attacks using graph of tokens and SVM[J]. Computers& Security,2016,60:206-225.

[6] 谢逸,余顺争. 基于 Web 用户浏览行为的统计异常检测[J]. 软件学报,2007,18(4):967-977.

[7] 温凯,郭帆,余敏. 自适应的 Web 攻击异常检测方法[J]. 计算机应用,2012,32(7):2003-2006.

[8] 彭思源. 基于 URL 的 Web 攻击异常检测方法[D]. 重庆:重庆邮电大学,2017.

[9] 刘泽宇,夏阳,张义龙,等. 基于 Web 行为轨迹的应用层 DDoS 攻击防御模型[J]. 计算机应用,2017,37(1):128-133.

[10] 闫伟,张军. 基于时间序列分析的网络流量异常检测[J]. 吉林大学学报(理学版),2017,55(5):1249-1254.

[11] 廖鹏,夏元轶,郭靓,等. 基于用户访问序列的异常行为检测方法[P]. 中国专利:CN106657410A,2017-05-10.

[12] 杨大路,范维,南淑君,等. 一种基于可信业务流的未知威胁检测方法[J]. 电子测试,2015(9):21-23.

[13] 姚伟. 业务系统异常行为检测[J]. 邮电设计技术,2016(1):70-73.

[14] 石波,王红艳,郭旭东. 基于业务白名单的异常违规行为监测研究[J]. 信息网络安全,2015(9):144-148.

Research on Abnormal Detection based on Business Logic

JIANG Mengdan, LIN Honggang, CAO Heming

(College of Cyberspace Security, Chengdu University of Information Technology, Chengdu 610225, China)

Abstract: With the popularity of the Internet, web sites are carrying more and more traffic. In order to ensure that they can provide information or services safely and efficiently, this article combines business logic and anomaly detection to proposes an anomaly detection scheme based on traffic data generated by web users' access behavior. The scheme is designed for web sites deployed on important nodes. Firstly, The scheme obtains the link relationship between all pages by obtaining the source code of the web site published on the network, and the topology of the web site is constructed to learn the user access path set that is in line with the normal business logic path. Then the behavior trajectories of different users are extracted from the traffic data generated by the access, and the algorithm proposed in the scheme is used to determine whether the user's access is abnormal. Finally, the effectiveness of the algorithm is tested and verified in the actual environment. The simulation results show that the anomaly detection method can find irregular access to the site, SQL injection attempt or semantic URL attacks toward the site.

Keywords: business logic; abnormal detection; traffic analysis; website structure; access information