基于信息熵的DNS拒绝服务攻击的检测研究

严 芬 丁 超 殷新春
(扬州大学信息工程学院 扬州 225127)

摘 要 DNS服务器在Internet中具有至关重要的作用,对它进行攻击会影响网络向用户提供正常的服务。DNS Query Flood攻击是最为常见的一种攻击方式,它向DNS服务器发送大量伪造的域名解析请求,消耗DNS服务器的资源,造成拒绝服务。及时检测到此类攻击的存在至关重要。在研究DNS解析过程的基础上,总结DNS Query Flood攻击的特点;根据攻击的特点,结合信息熵来判断网络是否出现异常;利用滑动窗口机制来确定是否存在攻击。

关键词 DNS Query Flood,拒绝服务,域名解析成功率,信息熵,滑动窗口
中图法分类号 TP393 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2015.3.029

Research on Exploiting DoS Attack Against DNS Based on Information Entropy

YAN Fen DING Chao YIN Xin-chun
(College of Information Engineering, Yangzhou University, Yangzhou 225127, China)

Abstract DNS server has a vital role in the Internet, and it will affect the network to provide normal services to users if DNS is attacked. DNS Query Flood attack sends a lot of fake DNS request to the DNS server, consumes the DNS server resources and causes denial of service. So it is very important to detect timely the attack. Based on the study of the DNS resolution process, we summed up the characteristics of the DNS Query Flood attack. According to the characteristics of attack, we combined the information entropy to determine whether a network abnormalities, and then used sliding window mechanism to determine whether there is any attack.

Keywords DNS query flood, Denial of service, Domain name resolution success rate, Information entropy, Sliding window

1 引言

随着网络的快速发展,网络安全问题越来越严峻。近几年,针对网络层的攻击问题已经得到很好的解决,致使攻击者因对网络层实施攻击相对较难而转向对应用层发动攻击,典型的应用层攻击是针对(域名系统 Domain Name System, DNS)服务器的攻击。DNS服务器在设计之初就因其自身的特点存在安全缺陷[1]。目前众多研究专家针对DNS服务器的安全缺陷提出了一些解决方案,例如IETF提出的DNS系统的安全协议DNSSEC[2],但该协议需要更多的系统开销及网络资源,后期还需要对相关的数据库和软件进行升级,因此该协议也不能很好地解决DNS安全缺陷问题。由于DNS服务器受到攻击的事件频发,研究者针对DNS攻击也提出了很多检测方法。宗兆伟等人提出了基于统计分析与流量控制的DNS攻击检测方法[3]。黄宸等人针对DNS攻击提出基于应用层DDoS攻击检测方法[4]。李锦玲也提出了应用层拒绝服务攻击的检测算法[5]。本文在研究DNS协议和DNS攻击方式的基础上,结合目前的研究现状,针对DNS Query Flood攻击提出基于信息熵的检测方法。

2 DNS服务及攻击方式

2.1 DNS服务

域名系统是Internet上最为关键的基础设施,其主要作用是提供域名与IP地址之间的转换,保障其他网络应用(如网页浏览、电子邮件等)的顺利执行。

DNS协议支持客户/服务器模式,当客户端要求解析某一域名的IP地址或者某一IP地址的域名时,向DNS服务器发送查询报文;本地服务器先搜索本地高速缓存,若缓存中存在IP地址和域名的资源记录,则直接返回给客户端;若不存在则进行递归查询获得相应的资源记录,把新的资源记录保存在高速缓存中并向客户端发送应答报文。

但是,DNS服务器本身存在诸多的安全缺陷[4],例如无正确认证性、无连接性、无状态性,而DNS服务器又是一个开放的体系,这些安全缺陷在某些时候可被攻击者利用而对DNS服务器发动拒绝服务攻击。

2.2 DNS攻击的方式

针对DNS服务器的攻击有多种形式,常见的攻击有以下几种。

(1)DNS Query Flood攻击

DNS Query Flood攻击采用的方法是向被攻击的服务器发送大量的域名请求。通常,请求解析的域名是随机生成的,或者是网络中不存在的域名。被攻击的DNS服务器在接收到域名解析请求后,首先会在服务器缓存上查找是否存有对应的域名,如果未查找到或该域名无法直接由当前服务器解析,DNS服务器则向其上层DNS服务器递归查询域名信息。在该情况下,域名解析的过程给服务器带来了很大的负载,当

到稿日期:2014-08-20 返修日期:2014-09-27

严 芬(1978—),女,博士,副教授,主要研究方向为网络与信息安全,E-mail:yanfen@yzu.edu.cn;丁 超(1990—),男,硕士生,主要研究方向为网络与信息安全;殷新春(1962—),男,博士,教授,主要研究方向为密码学与信息安全。

域名解析请求超过每秒钟某一既定数量时就会造成 DNS 服务器解析域名超时。

通常攻击者使用的手段为：

①利用发包程序向 DNS 服务器发送不带任何数据的空包。该类数据包本身并不符合协议规定，服务器收到报文后将直接丢弃。因此，这种攻击方式在攻击流量很小的情况下不会有明显的效果。

②利用程序构造 DNS 请求固定的域名。该种方法不停地向 DNS 服务器发送请求，最终造成 DNS 服务器的瘫痪。

③利用伪造程序向 DNS 服务器发送随机伪造的域名解析请求。这样，DNS 服务器就会不停地进行字符串匹配，造成大量 DNS 服务器资源的消耗。

(2)欺骗式攻击[6]

攻击者先执行利用型攻击(如缓冲区溢出、特洛伊木马等)侵入 DNS 服务器的高速缓存，并诱导其存储虚假的信息，或是获得 Root 权限改变服务器的转换表，使不同的域名映射到同一 IP 地址，即被攻击目标的 IP 地址。这时，用户发出一个域名解析请求后，收到的应答报文中关于相应的 IP 数据是被攻击目标的 IP 而非用户所要查询的 IP。

3 DNS 拒绝服务攻击的检测

3.1 检测原理

通过分析比较正常情况时的 DNS 数据和攻击发生时的 DNS 数据，在拒绝服务攻击发生时，可以得到如下特征。

(1)域名解析的成功率会降低：由于攻击使用的都是伪造的域名或者随机伪造的 IP 地址，此类域名或 IP 地址最终会解析失败，在这个过程中会大量消耗 DNS 服务器的资源，造成 DNS 服务器瘫痪。

(2)网络流量迅速增大：在正常情况下，DNS 服务器的工作流量在特定的范围内浮动，表现出相对稳定的状态。在拒绝服务攻击发生时，攻击者不断向 DNS 服务器发送伪造的域名或者 IP 地址，快速消耗 DNS 服务器的资源，造成流量增大，引起流量突变。

根据上述两个 DNS 服务器数据的变化特征，本文针对 DNS Query Flood 攻击提出了一种基于信息熵的检测方法。

3.2 检测方法

1. 参数的定义

(1)时间窗 T。时间窗 T 是指一个时间段，通过分析 T 时间段内的 DNS 数据流量，获得该时间段内 DNS 数据流量的变化情况。

(2)域名解析成功率 P_s。统计上述时间窗 T 内的 DNS 请求总数 N、域名成功解析的数目 N_s，从而计算出时间窗 T 内域名解析的成功率：

$$P_s=N_s/N \tag{1}$$

(3)域名解析成功率信息熵 H。利用(2)中的域名解析成功率计算信息熵值，从而判断网络中是否出现异常。信息熵值公式如下：

$$H=-\sum_{i=1} p_i \log(p_i) \tag{2}$$

(4)滑动窗口 m。将待检测的 DNS 请求包的源 IP 地址序列引入滑动窗口，窗口大小定位 m，利用滑动窗口算法计算源 IP 地址的信息熵值。

2. 检测步骤

(1)利用 Wireshark 软件抓取正常情况下 DNS 数据包，分析得到正常情况下域名解析成功率信息熵值的变化情况。对每隔时间窗 T 抓取的数据包进行分析，每次处理分析特定数量的数据包，则每个时间窗 T 内计算得到一系列域名成功解析的概率值，记为 $P=\{P_i, i=1,2,\cdots,n\}$。然后利用式(2)按特定的周期计算概率的信息熵值，记为 $H=\{H_i, i=1,2,\cdots,k\}$。

(2)对于攻击者来说，为了安全，每次按照一定的时间间隔发送攻击包实施攻击，每次攻击的时间较短，根据这一特点抓取待检测数据包，能够减少 DNS 服务器的开销。为了尽量准确地分析网络中的异常流量，在攻击检测时，定义时间窗 $T=0.5$ 小时，在 DNS 服务器上抓取数据包进行分析，每次间隔 5 分钟。这样做的优点是时间窗口 T 的大小可以根据实际网络环境进行配置，当网络繁忙时可缩短抓包间隔，增加抓包时间，这样可增加灵活性，提高抓包准确性。

(3)采用步骤(1)的算法计算待检测数据流中域名解析成功率的熵值。

(4)利用(3)中得到的熵值序列与正常情况下信息熵值进行比较，得出信息熵值的变化情况，即是否发生信息熵值突变，从而判断 DNS 服务器是否出现异常。

但是，并不是所有的熵值突变都是 DNS 拒绝服务攻击造成的。在正常网络环境下服务器由闲到忙，或者在同一时间段内大量用户访问同一站点，同样也会造成信息熵值的突变。为了提高检测方法的准确率，降低误判率，必须要将正常网络引起的熵值突变和攻击发生时引起的网络突变区分开来。

实际上，造成上述现象的主要原因就是在很短的时间内，存在大量的域名解析请求，而这些域名请求在正常情况下和攻击情况下是不同的。对 DNS 服务器的大多数攻击，源 IP 地址的分布都会存在异常，反映在信息熵上则是源 IP 地址的熵值与正常情况下的熵值产生较大的偏差。

基于这一特点采用滑动窗口机制[5]统计域名请求数据包的源 IP 地址的信息熵值，即采用一个固定大小的滑动窗口，在此窗口内计算每个 IP 地址出现的概率。窗口的大小可以进行调整，具体步骤如下：

①对源 IP 地址序列引入滑动窗口，每个窗口取 m 个源 IP 地址。m 为最大窗口容量。

②计算滑动窗口内每个源 IP 地址出现的概率，记为 $P=\{P_i, i=1,2,\cdots,l\}$，$l$ 表示一个窗口内不同源 IP 地址的个数($1\leqslant l\leqslant m$)，然后利用式(2)计算窗口内源 IP 地址的熵值。

③取出当前滑动窗口中第一个源 IP 地址在滑窗中的出现次数和相应概率，概率记为 P_1。

④滑动窗口向后推移一项，原滑动窗口中第一项移出，原第 $m+1$ 项移入滑动窗口中。重新计算 P_1，也就是移出的 IP 地址在当前窗口中的概率，即 $P_1=P_1+1/m$；并计算移入的源 IP 地址在当前窗口中的概率，若移入的源 IP 地址在上一个窗口中出现过，假定其概率为 P_j($1\leqslant j\leqslant l$)，则更新 P_j 的值，$P_j=P_j+1/m$；若未出现，则计算该源 IP 地址的概率，$P_{l+1}=1/m$。

⑤根据以上步骤，对于每个滑动窗口都能计算出窗口内当前源 IP 的熵值。这样，可以计算得到一系列的熵值，记为 $H=\{H_i, i=1,2,\cdots,t\}$，$t$ 为滑动窗口总数。

在正常情况下，源 IP 地址分布比较稳定，因此信息熵值

也在一定的范围内波动。一般来说，由于攻击源数据包的源IP地址是随机生成的，随机性较大，对源IP地址进行统计分析得到的信息熵值会大于正常情况。而在网络拥塞的状况下，数据包的源IP都是真实的且经常出现的，随机性较小，这样统计出来的熵值与正常情况下比较会变小。

由此根据源IP地址的熵值变化情况来区分是网络拥塞情况下造成的域名解析失败率增大，还是攻击情况下造成的域名解析成功率变小。

4 实验方案与结果分析

为更好地验证该方法的有效性，在windows server 2003环境下搭建DNS服务器A、B、C，DNS服务器A、B、C之间可以递归查询，并设置了Web服务器、客户端和攻击端，如图1所示。在搭建好的环境上利用Wireshark软件抓取DNS服务器A的数据包并对其进行分析。实验中根据常见的DNS攻击方法和原理，采用C语言实现了DNS Query Flood攻击程序，攻击端每隔1毫秒向搭建好的DNS服务器A发送一个源IP地址伪造的域名解析请求，让DNS服务器A不断进行域名解析，耗尽其资源，并在DNS服务器端A抓取DNS数据包进行检测。

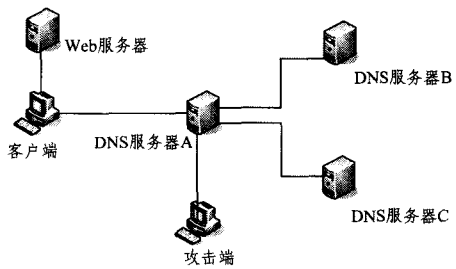

图1 实验环境结构

主要步骤如下：

首先，在客户端访问Web服务器，利用Wireshark软件在DNS服务器A上抓取正常情况下的DNS数据包，统计在正常情况下域名解析成功率的变化情况，并计算出正常情况下域名解析成功率信息熵值。如图2所示，在正常情况下，域名解析成功率信息熵值在0.150.25之间波动，相对平稳，未出现突变。

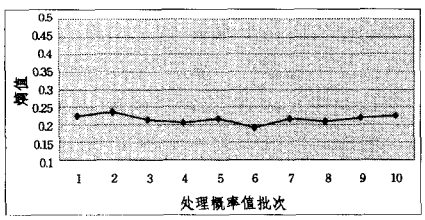

图2 正常情况下域名解析成功率熵值变化

其次，向DNS服务器A发送攻击程序，并继续抓包，我们抓取半小时的数据包进行统计分析。利用程序统计每20个DNS数据包中请求数据包数目和成功应答数据包数目，计算域名解析的成功率，从而得到一系列域名解析成功率。在此基础上，每3个概率计算一个信息熵，得到域名解析成功率信息熵值序列，通过信息熵值的变化情况来判断DNS服务器是否出现异常。

由图3可见，在DNS服务器没有受到攻击时，信息熵值在0.1～0.5之间波动（如图中的1—2批次）；当向DNS服务器实施DNS攻击时，信息熵值会突变到很大的范围内，因为在DNS服务器是否受到攻击时，域名解析的成功率会大大降低（如图中的3—19批次）。但随着攻击的结束，DNS服务器慢慢恢复正常，相应的域名解析成功率会升高，信息熵值又会降低到很小的范围内。通过图中熵值的突变情况来判断DNS服务器是否存在异常。

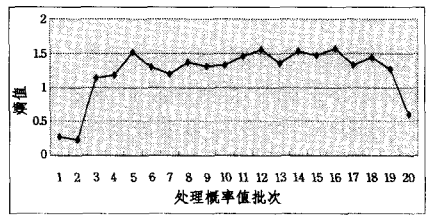

图3 待检测数据包域名解析成功率信息熵值变化情况

然后，结合滑动窗口机制进一步确定是否发生DNS攻击。仅通过以上的方法并不能准确判断DNS服务器受到攻击，因为网络拥塞也会造成域名解析失败，其信息熵值也发生变化。针对这一情况，结合采用滑动窗口分析请求报文的源IP地址的变化情况。在保证精确度的情况下，滑动窗口的大小由上文中每次处理的数据包而定，这里滑动窗口大小m取55。一般情况下，大多数针对DNS服务器的攻击数据包源IP都是随机生成的，反映在信息熵上，熵值会大于正常情况下的熵值。而在网络拥塞的情况下，数据包的源IP地址是真实的，并且都是经常出现的，随机性较小，计算出来的熵值应该小于正常情况下的熵值，如图4所示。

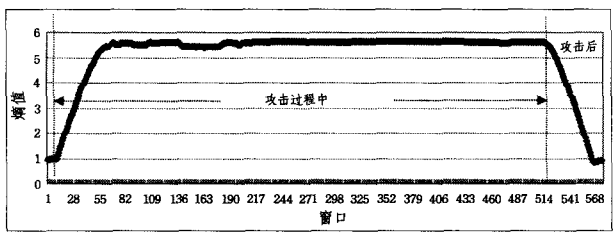

图4 待检测数据包源IP信息熵

考虑从图4中可以看出，攻击前，源IP地址的信息熵值保持在较小的范围内，随着DNS攻击的实施，即图3中域名成功率信息熵值发生突变后，图4中源IP地址的信息熵值也会相应地呈现逐渐变大的趋势。当滑动窗口内都充满攻击数据包的源IP地址时，且窗口内每个IP地址出现的概率都相同时（即窗口内每个IP地址出现的概率为$1/m$），信息熵达到最大值。当对DNS服务器攻击停止后，源IP地址的信息熵值呈现逐渐变小的趋势。当窗口中不存在攻击数据的IP地址时，源IP地址的熵值趋于稳定。

根据源IP地址的熵值变化情况来看，结合上述域名解析成功率的熵值变化情况可以准确判断DNS服务器是否存在攻击。利用域名解析成功率计算熵值，如果熵值有突然变大的情况，说明DNS服务器存在异常；为了区分这种异常是攻击造成的还是网络拥塞情况下造成的，采用了滑动窗口机制的方法对源IP地址进行统计分析计算其信息熵值，针对源IP地址是随机生成的攻击来说，其源IP地址的熵值与正常情况下相比会变大，而在网络拥塞的情况下，其源IP地址都是真实的，随机性较小，熵值与正常情况下相比会变小。因此将两种方法结合能够准确地判断DNS服务器中是否存在攻击。

结束语 本文提出了一种针对 DNS Query Flood 攻击的检测方法，其利用域名解析的成功率计算出信息熵值，根据信息熵值的变化情况来判断 DNS 服务器是否出现异常。在判断 DNS 服务器出现异常的情况下，利用了滑动窗口进一步判断源 IP 地址的信息熵的变化情况，从而判断 DNS 服务器是否受到了攻击，并通过实验验证了该检测方法的有效性。这两种基于信息熵的方法结合起来使用可以有效准确地检测到攻击，提高检测的准确率，在某种程度上降低了漏报率；同时这种方法不需要设置阈值，这就避免了因经验不足而设置错误阈值的情况。

参 考 文 献

[1] Mockapetris P. Domain Names-Concepts and Facilities [S]. RFC1034. 1987

[2] Eastlake D. Domain Name System Security Extensions [S]. RFC2535. 1999

[3] 宗兆伟，黎峰，翟征德. 基于统计分析和流量控制的 DNS 分布式拒绝服务攻击的检测及防御[C]//2009 年计算机网络与通信学术会议论文集. 2009：206-213

[4] 黄宸，郑康峰，卢天亮，等. 基于信息熵的应用层 DDoS 攻击检测方法[C]//第十七届全国青年通信学术年会论文集. 第二卷，2012：467-472

[5] 李锦玲. 应用层分布式拒绝服务攻击的异常检测算法研究[D]. 郑州：解放军信息工程大学，2013

[6] 张小妹，赵荣彩，单征，等. 基于 DNS 的拒绝服务攻击研究与防范[J]. 计算机工程与设计，2008，29(1)：21-23

[7] 王佳佳. DDoS 攻击检测技术的研究[D]. 扬州：扬州大学，2008

[8] 刘永杰. 异常流量识别系统及其关进技术研究[D]. 南京：南京邮电大学，2013

[9] 徐川. 应用层 DDoS 攻击检测算法研究及实现[D]. 重庆：重庆大学，2012

[10] 尚波涛，祝跃飞，陈嘉勇. 一种应用层分布式拒绝服务攻击快速检测方法[J]. 信息工程大学学报，2012(5)：601-607

(上接第 127 页)

(1)识别资源的功能块/功能块要素。这里功能块的定义为资源(硬件、软件)中能够影响到资源功能安全的最小元素，一个功能块可由许多功能块组成(功能综合)，组成功能块的部分成为功能要素。例如，一个用于控制容器燃油容量的安全阀门可以分解为图 8 所示的功能块。

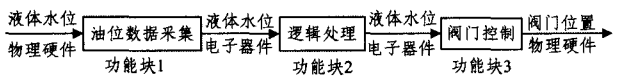

图 8 安全阀门-功能块分解示例

(2)将(1)中的功能块/功能块要素映射到子系统/子系统要素。子系统/子系统要素是一个物理实体的概念，是功能块/功能块要素的实现。一个子系统可以包括多个其它子系统，构成子系统的部分称为子系统要素，其失效会影响整个资源的失效，例如上例中构成安全阀门的每个子系统(水位传感器、可编程逻辑电路和阀门)失效都会影响到安全阀门的功能失效。对于子系统/子系统要素，其安全性分析需要考虑架构方面的约束和失效概率方面的要求。

(3)通过对资源的功能块、子系统分解，我们可以采用基于 Event-B 和 AltaRica 进行建模分析，这里不再赘述。

结束语 本文针对综合化航空电子系统安全性分析存在的失效模式完备性和动态失效问题以及数据一致性问题，提出了基于模型驱动的安全性分析方法，分别从应用操作层、功能层和资源层对航电系统建模，借助形式化工具能够实现 3 个层次上的语义一致性，从而从源头解决系统设计与安全性工作的相互分离的现象。本文的工作主要有如下 3 点：

(1)对综合化航空电子系统特征和安全性分析流程进行了分析，提出了现有工程实践中存在的问题；

(2)提出了基于模型的安全性分析流程和方法，实现了系统设计与安全性设计的集成；

(3)提出应用操作、功能和资源层分别建模的方法，并以航电系统安全性分析实例对提出的方法进行了验证。

参 考 文 献

[1] Society of Automotive Engineers. ARP-4761：Aerospace Recommended Practice：Guidelines and Methods for Conducting the Safety Assessment[C]//Process on Civil Airborne Systems and Equipment. 1996

[2] Papadopoulos Y, McDermid J A. Hierarchically Performed Hazard Origin and Propagation Studies[C]//Proceedings of SAFECOMP '99, 18th International Conference on Computer Safety, Reliability and Security. 1999

[3] Joshi A, Miller S P, Heimdahl M P E. Mode Confusion Analysis of a Flight Guidance System Using Formal Methods[C]//Proceedings of the 22st Digital Avionics Systems Conference (DASC'03). Indianapolis, Indiana, Oct. 2003：12-16

[4] Description A[OL]. [2012-01-19]. http://www.lix.polytechnique.fr/rauzy/

[5] IEC 61508 Functional Safety of Electrical/Electronic/Programmable Electronic Safety-Related Systems[S]. http://zh.wikipedia.org/wiki/IEC-61508, 1998

[6] Adeline R, et al. Toward a Methodology for The AltaRica Modelling of Multi-Physical Systems[C]//European Conference on Safety and Reliability (ESREL). Taylor & Francis：Rhodes, Greece, 2010

[7] Liu S, McDermid J A. A Model-Oriented Approach to Safety Analysis Using Fault Trees and a Support System[J]. Journal of Systems and Software, 1996, 35(2)：151-164

[8] Dotti F L, Iliasov A, Ribeiro L, et al. Modal Systems：Specification, Refinement and Realization[C]//Proceedings of the 11th International Conference on Formal Engineering Methods：Formal Methods and Software Engineering(ICFEM'09). 2009：601-619

[9] Chaudemar J-C, Bensana E, Castel C. Christel Seguin AltaRica and Event-B Models for Operational Safety Analysis：Unmanned Aerial Vehicle Case Study[OL]. [2014-03-19]. http://www.lix.polytechnique.fr/rauzy/altarica/AltaRica.html/

[10] Troubitsyna E, Laibinis L. Fault Tolerance in a Layered Architecture：a General Specification Pattern in B[C]//Proc. of the 2nd Int. Conference on SEFM. Beijing, IEEE, 2004：346-355

[11] Abrial J R. The B-book：Assigning Program to Meanings[M]. CUP, 1996

[12] Gallier J H. Logic for Computer Science：Foundations of Automatic Theorem Proving[M]. Publications Dover, 1986