防 TTL 值欺骗的数据包标记算法研究

刘 渊[1,2],李秀珍[2],朱晓键[2]
LIU Yuan[1,2],LI Xiu-zhen[2],ZHU Xiao-jian[2]

1.南京理工大学 计算机学院,南京 210094
2.江南大学 信息工程学院,江苏 无锡 214122
1.School of Computer,Nanjing University of Science & Technology,Nanjing 210094,China
2.College of Information Engineering of Southern Yangtze University,Wuxi,Jiangsu 214122,China
E-mial:mblxz@yahoo.com.cn

LIU Yuan,LI Xiu-zhen,ZHU Xiao-jian.Research on packet marking algorithm resisted spoofed TTL value.Computer Engineering and Applications,2008,44(23):127-129.

Abstract: Distributed Denial of Service(DDoS) attack is among the hardest network problems.To reply it,many kinds of schemes of countermeasures are proposed,these schemes all respectively have the good and bad points.But among these,an Adaptive Probabilistic Packet Making(APPM) is promising and using.In this paper,based on the snooped initial TTL value by the attacker,an adaptive marking scheme is improved,which is advantageous to resist spoofed TTL value,to reduce the router burden and save the IP packet's space.

Key words: Distributed Denial of Service(DDoS);packet marking;Adaptive Probabilistic Packet Making(APPM);IP traceback

摘 要:分布式拒绝服务攻击是目前最难处理的网络难题之一,针对分布式拒绝服务攻击提出了多种应对方案,这些方案都各有优缺点,但其中自适应概率包标记受到了广泛地重视和运用。针对攻击者对 TTL 初始值的伪造提出了一种自适应策略,有利于防止 TTL 值的伪造,减少路由器处理器的负担,节省了 IP 包头的空间。

关键词:分布式拒绝服务;包标记;自适应概率包标记;IP 回溯
DOI:10.3778/j.issn.1002-8331.2008.23.039 **文章编号**:1002-8331(2008)23-0127-03 **文献标识码**:A **中图分类号**:TP393

拒绝服务攻击(DoS)这种攻击行动使网站服务器充斥大量要求回复的信息,消耗网络带宽或系统资源,导致网络或系统不胜负荷以至于瘫痪而停止提供正常的网络服务。分布式拒绝服务攻击(DDoS)是指大量的攻击主机同时对受害者发起拒绝服务攻击,其攻击强度比单攻击者情形要大得多,而且由于攻击者的分布式特性,使得预防和消除这类攻击更加困难。针对 DDoS 攻击的特点,研究人员提出了很多研究方法,但解决的根本途径是跟踪攻击流,找出攻击路径并发现攻击源,在近攻击源端就彻底阻止攻击。Savage[1]等人提出的概率包标记(PPM)方案,通过路由器对 IP 包的标记,使受害者可以分析一定数目的攻击包中包含的标记信息,来完成对攻击路径的部分重构,并达到对风暴型拒绝服务攻击的路由进行追踪的目的,这一方案以其实现技术简单、不增加网络负载、效率高等优点,近年来得到了广泛研究和探讨。研究者们在此基础上,提出了多种改进包标记方案。在此基础上 T.Peng[2]又提出了自适应概率包标记(APPM)策略。本文在 APPM 算法的基础上进行改进,可以达到防止因攻击者伪造 TTL 值而导致对攻击路径的部分重构失效。

1 包标记算法

1.1 PPM 算法

概率包标记的思想[1,3]是路由器以固定的概率 p(一般 p=0.04)标记数据包,将路径信息加入到包头中的三元组(*start*,*end*,*distance*)中,路由器将自己的 IP 地址填入到 *start*,同时 *distance* 赋值为 0;否则如果 *distance* 为已经被标记为 0,表明上一个路由器对包进行了标记,则将自己的 IP 地址填入到 *end* 中;如果路由器以概率 1-p 不对包标记,则将 *distance* 值加 1。受害者从这些数据包中提出路径信息,重构出攻击路径。当一个路由器标记一个数据包以后,该数据包可能被后续的路由器重新标记,使原有的标记信息被覆盖。设攻击路径为(a,r_1,r_2,…,r_D,v),a 为攻击者,v 为受害者。

定义 1 α_i 表示为它最后被 r_i 路由器标记而不被随后的其它路由器标记的概率。

$$\alpha_i=\begin{cases}P_i\prod_{j=i+1}^{D}(1-p_j), & 1\leqslant i<D\\ p_D, & i=D\end{cases} \quad (1)$$

基金项目:国家部委预研基金资助项目(the Pre-Research Foundation of China Ministries and Commissions)。
作者简介:刘渊(1967-),男,教授,硕士生导师,主要研究方向为网络安全及网络信息系统及网络信息系统;李秀珍,(1982-),女,硕士生,主要研究方向为网络信息安全;朱晓建(1984-)男,硕士,主要研究方向为网络安全。
收稿日期:2007-10-17 **修回日期**:2008-01-11

对于 PPM 算法则：

$$\alpha_i=P(1-P)^{D-i},1\leqslant i\leqslant D \tag{2}$$

则即 $\alpha_1<\alpha_2<\cdots\alpha_D$ 被路由器 r_1 标记而没有被下游的路由器标记的概率最小，则包没有被标记的概率为：

$$\alpha_0=1-\sum_{i=1}^{D}\alpha_i=(1-p)^D \tag{3}$$

当 p=0.04，D=20 时 $\alpha_0\approx0.44$，则为攻击者伪造路由信息提供了很大的机会。

文献[4-5]的重构路径需要的数据包数为：

$$N_{ppm}=\frac{\ln D}{p(1-p)^{D-1}} \tag{4}$$

由此可知随着路径的增长需要的数据包数成指数增长，在受害者端不能很快的重构出攻击路径。

1.2 自适应概率包标记

为了减少路径重构所需要的数据包数，T.Peng[2]等人在此基础上提出了 APPM 算法（自适应概率包标记）。在 APPM 算法中，每个路由器标记包的概率是 $p=1/i$，i 是从第一个路由器到现在所在路由器之间的距离。假设从攻击者到受害者的距离是 $D+1$，即数据包从攻击者出发到受害者之间有 D 个路由器，由 $p=1/i$ 可知当路由器距离受害者最远时，包标记的概率是最大的，随着离受害者距离的越来越近，标记数据包的概率也越来越小。则：

$$\alpha_i=P_i\cdot\prod_{j=i+1}^{d}(1-p_j)=p_i(1-p_{i+1})\cdot(1-p_{i+2})\cdots(1-p_D)=$$
$$\frac{1}{l}\cdot(1-\frac{1}{i+1})\cdot(1-\frac{1}{i+2})\cdots(1-\frac{1}{D})=\frac{1}{D}(1\leqslant i\leqslant D) \tag{5}$$

包没有被标记的概率：

$$\alpha_0=1-\sum_{i=1}^{D}\alpha_i=1-D\cdot\frac{1}{D}=0 \tag{6}$$

即攻击者的数据包都会被标记。

由不平均概率 CouponCollection[6]重构路径需要的数据包数最少平均为：

$$N_{appm}=D\cdot(\frac{1}{1}+\frac{1}{2}+\frac{1}{3}\cdots+\frac{1}{D})\approx D\cdot\ln D \tag{7}$$

假设有 N 个数据包，每个路由器的消耗量就为 $O_{appm}=N\cdot p_i$ 那么总的消耗为：

$$O_{appm}=N\cdot(\frac{1}{1}+\frac{1}{2}+\frac{1}{3}\cdots+\frac{1}{D})\approx N\cdot H_D \tag{8}$$

但是 T.Peng[2]等人提出的 APPM 算法中，为了计算 i 的值，就需要在 IP 的选项域（option）中增加一个额外的标记域，每经过一个路由器，i 的值就加 1，同时计算出路由器标记包的概率 p，然而这个标记域极有可能被攻击者伪造，那么标记的概率就不能遵守 APPM，而且在数据包传输过程中，路由器往 IP 选项域中写入数据是很费时的操作，这个标记域要被途中每个路由器进行读取，修改操作，这样会占用路由器的资源，影响路由器的处理器效率。在 IP 包头中的 TTL 域的值每经过一个路由器就会减少 1，可以利用 IP 包的 TTL 域，这样就可以记录数据包经过的路由器数目。由于不同系统 TTL 的初始值也不一样，攻击者很容易伪造 TTL 初始值，基于这种情况下面给出一种算法。

1.3 防止 TTL 欺骗的包标记方法

不同的系统 TTL 的初始值有 X={32,64,128,255}，路由器识别 TTL 的初始值是通过判断到达该路由器时候的值，设到达路由器时候 TTL 的值为 t 则路由器的初始值 $T0\geqslant t$，为了精确计算 TTL 的初始值，设一个从小到大排列的集合 $X=\{v_i|1\leqslant i\leqslant s,v_i<v_j,1\leqslant i<j\leqslant s\}$，$s$ 是 X 集合中元素的个数，那么当 TTL 的值为 t 时候 $v_{i-1}<t\leqslant v_i$，就可以判别 TTL 的初始值 $T0=v_i$。由于这样不同系统默认的 TTL 初始值不同，攻击者很容易通过更改 TTL 的初始值来达到欺骗目的。针对这样 TTL 欺骗，采取的办法是统一 TTL 的初始值设为 $T0$，当到达路由器的数据包的 TTL 值要是大于 $T0$ 时就重新写入 TTL 值为 $T0$，并以概率 $p=1$ 标记该数据包。如果所有的系统都遵守这个规定并且路由器都是诚实的，只有攻击者会把 TTL 初始值设置为大于 $T0$，那么攻击者的发送的数据包将都会被标记，在受害端根据标记的包就会很快重构出攻击路径找到攻击源。经研究数据包传输很少超过 25 跳的，所以 TTL 初始值最好是设置为 32。

但是攻击者会把 TTL 的初始值设置为比 $T0$ 小的数，假设 $z(0\leqslant z<T0)$，那么它的初始值就是 $TTL=T0-z$，在这种情况下没有一个路由器可以识别它，路由器 r_1 则以为它的距离是 $1+z$ 跳，并以概率 $p'=\frac{1}{1+z}$ 标记，同样 r_i 以概率 $p'=\frac{1}{i+z}$ 标记，由式(5)可得：

$$\alpha_i'=\frac{1}{D+z} \tag{9}$$

即在 TTL 欺骗的情况下每个路由器的 α_i' 也都是一样的，由公式(7)可得重构出攻击路径需要的包数是：

$$N'_{appm}\approx D\cdot\ln(D+z) \tag{10}$$

没有标记包的概率：

$$\alpha_0'=1-\sum_{i=1}^{D}\alpha_i'=\frac{z}{z+D} \tag{11}$$

攻击者的数据包不能完全被标记，由式(8)可得路由器总的消耗为：

$$O'_{appm}=N\cdot(H_{D+z}-H_z) \tag{12}$$

由于 TTL 的值每经过一个路由器其值就减小 1，当 TTL 的值为 0 时，数据包就被路由器抛弃，统一 $T0$=32，攻击者为了不让数据包在途中丢弃，会尽可能选择小的 z。由式(11)：$\alpha_0'=\frac{z}{z+D}=1-\frac{D}{D+z}$ 随着 z 的增大 α_0' 的值也就增大，重构路径所需要的包数 N'_{appm} 也就增大。当 D=20，z 取最大值 12 时 $\alpha_0'\approx0.375$，显然比 PPM 算法要小，但是若 TTL 的值不统一，当攻击者实施 TTL 欺骗的时，可以把 TTL 设置很大的值，假设系统的 TTL 值是 64，攻击者设置的 TTL 值是 66，那么到达 r_1 时 TTL 的值是 65，路由器就认为 TTL 的初始值是 128 并以概率 $P=\frac{1}{128-65}=\frac{1}{63}$ 的概率标记它，显然没有标记的概率是很大，后面的路由器标记的概率都要小于 $\frac{1}{63}$，这样受害者需要收到大量的数据包才能重构出攻击路径，这样会浪费很多的时间，不利于很快找到攻击源。

2 模拟实验

图 1 是对 D=20、z=0、z=6 和 z=12 的情况下，选取攻击路径从 1~30，进行 10 000 次实验取平均值得到的结果，为检验真实环境的可行性实验数据来源于 CAIDA[7]（Cooperative Association for Internet Data）提供的跟踪路由仿真实验。可以看出随着 z

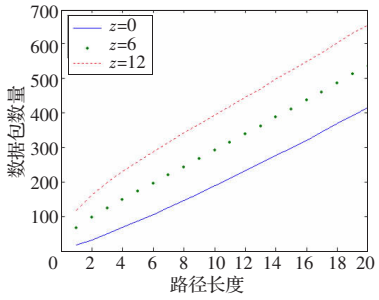

图 1　z 值不同所需的数据包数

值的增大所需要的数据包数量是增大的。当 z=6、z=12 的时候所需要的数据包平均约是 z=0 时候需要数据包的 1.5 倍和 1.9 倍,当 z 的值大于 12 时候需要的数据包数量是很大的,这对于受害者来说是很不利的,但是统一 TTL 值为 32 的情况下攻击者伪造的 TTL 值的变化范围是很小的, 因为攻击者要考虑数据包被丢弃的情况会尽可能的选择比较小的 z 值。

3　结束语

本文是分析了在 IP 包头增加额外的标志域来计算距离,这个标志域极容易被攻击者伪造,而且会给路由器增加很大的负担的情况下,提出了利用 TTL 域来计算距离,针对 TTL 的欺骗提出了一种算法,若所有的系统都规定统一的 TTL 初始值,这种算法会比原来的利用 IP 包头增加额外的标志域有更好的优势。(1)它可以减少攻击者伪造数据,即使攻击者伪造 TTL 值,当伪造的值大于初始值时它会完全标记出攻击包;(2)它可以减轻路由器的负担,不需要向 IP 包头的 option 域中写入数据,有利于提高路由器的处理速度;(3)不需要增加额外的标记域来表示距离这样会节省 IP 包头的空间,减少数据包的分片,在受害端需要很少的数据包就能重构出攻击路径。对于攻击者伪造的 TTL 初始值比统一值小的时候的处理方案有待于进一步研究。

参考文献:

[1] Savage S,Wetherall D,Karlin A,et al.Practical network support for IP traceback[C]//Proceedings of the 2000 ACM SIGCOMM Conference,Stockholm,Sweden,2000:295-306.

[2] Peng T,Leckie C,Kotagiri R.Adjusted probabilistic packet marking for IP traceback[C]//Proceedings of the 2nd IFIP Networking Conference(Networking 2002),Pisa,Italy,2002:697-708.

[3] Li D Q,Su P R,Feng D G.Notes on packet marking for IP trace back[J].Journal of Software,2004,15(2):250-258.

[4] Savage S,Wetherall D,Karlin A,et al.Network support for IP traceback[J].IEEE/ACM Transactions on Networking,2001,20(2):226-237.

[5] Savage S,Wetherall D,Karlin A,et al.Practical network support for IP traceback[C]//ACM SIGCOMN,2000:295-306.

[6] Adler I,Ross S M.The coupon subset collection problem[J].Journal of App lied Probability,2001(3):737-746.

[7] Cai D A.Cooperative association for internet data analysis[EB/OL].(2006-05).http://www.caida.org.

(上接 119 页)

进行两次 Montgomery 模乘,但文献[7]提出的模逆算法在硬件实现上也只设计了一个模乘器,通过2 次或 3 次循环调用该模乘器来完成模乘运算,而本文算法中增加了两步计算,一是预计算 $c=2^k$ 和 $c=2^{2k}\pmod p$这两个值并存储于控制单元的寄存器中,二是采用移位寄存器来计算以 2 为底的整数方幂,这在一定程度上增加了硬件资源的开销, 但相对于模乘计算而言,计算以 2 为底的整数方幂在硬件实现上要简单高效得多。因此本文设计相对于文献[7]提出的两种模逆运算的硬件实现所消耗的资源要稍大一些。本文设计的主要创新点在于用一套硬件资源实现 Montgomery 模逆运算和一般整数模逆运算这两种公钥密码系统中主要的求逆运算,而若使用两种单独的运算结构来进行一般整数模逆运算和 Montgomery 模逆运算, 所消耗的硬件资源将接近本设计的 2 倍[8]。

5　结束语

本文设计的模逆运算结构是一种很基本的结构,最大的特点在于可以基于这样的结构设计一套硬件资源实现两种类型的模逆运算。并且如果采用一种具有可伸缩功能的模乘器,比如采用 FIOS 算法的模乘器[9],该结构就能完成素数域上 Montgomery 模逆域内和整数域内的任意长度数据的求逆元运算,这对于提高整个椭圆曲线加密系统的可配置性有很重要的意义。

参考文献:

[1] Miller V S.Use of elliptic curves in cryptography[C]//CRYPTO'85,1986:417-426.

[2] Koblitz N.Elleptic curve cryptosystems[J].Mathematics of computation,1987,48(4):203-209.

[3] Standard Specifications for Public-key Cryptography.IEEE Standard P1363[S/OL].(2000).http://grouper.ie-ee.org/groups/1363.

[4] Hankerson D,Menezes A,Vanston S.椭圆曲线密码学导论[M].张焕国,译.北京:电子工业出版社,2005.

[5] Mcivor C J,Mcloone M,Mccanny J V.Improved montgomery modular inverse algorithm[J].IEEE Electronics Letters,2004,40(18).

[6] Kaliski B S.The montgomery inverse and its applications[J].IEEE Transactions on Computers,1995,44(8):1064-1065.

[7] Savas E,Koc C K.The montgomery modular inverse-revisited[J].IEEE Transactions on Computers,2000,49(7):763-766.

[8] Mcivor C J,Mcloone M.Hardware elliptic curve cryptographic processor over $GF(p)$[J].IEEE Transactions on Circuits and Systems,2006,53(9):1946-1957.

[9] Koc C K,Acar T,Kaliski B S.Analyzing and comparing montgomery multiplication algorithms[J].IEEE Micro,1996,16(3):26-33.