基于 IPv6 拒绝服务攻击和改进确定包算法研究*

赵树枫
(上海理工大学 信息化办公室, 上海 200093)

摘 要: 对 IPv6 下拒绝服务攻击进行了研究,并根据 IPv6 协议的特点,提出一种基于 IPv6 的 MAC 认证改进确定包标记(ADPM-v6)算法。ADPM-v6 利用 IPv6 新特性,即逐跳选项和改进的 MAC 认证方法,有效解决了受控路由器修改标记的问题,能直接快速地追踪攻击源。同时分析验证了 IPv6 真实攻击环境的数据包大小分布,使得算法有效且更具有较强的实用性。理论分析和仿真实验结果表明,该算法在 IPv6 下大大缩短了重构时间,减少了重构计算量和误报率。

关键词: IPv6 协议; IP 追踪; 分布式拒绝服务; 确定包标记; MAC 认证
中图分类号: TP393 **文献标志码**: A **文章编号**: 1001-3695(2012)05-1914-04
doi:10.3969/j.issn.1001-3695.2012.05.084

Study of denial of service attack & improved deterministic packet marking scheme based on IPv6

ZHAO Shu-feng
(*Information Office*, *University of Shanghai for Science & Technology*, *Shanghai* 200093, *China*)

Abstract: This paper studied the DoS attacks under IPv6, and according to the new features of IPv6 protocol, proposed the ADPM-v6 scheme based on improved MAC-authentication. By using a new IPv6 feature hop-by-hop options and the improved MAC authentication method, the scheme could quickly locate the source of attacker. The scheme could prevent compromised router which could forge the marking effectively. Simultaneous, analysis of a real attack environment verifying IPv6 packet size distribution, made the algorithm have a strong practicality and more effective. The implementation and evaluation shows that the ADPM-v6 scheme can greatly reduce the reconstruction time and reduce the amount of reconstruction computing and false positive.

Key words: IPv6 protocol; IP traceback; distributed denial of service(DDoS); deterministic packet marking (DPM); MAC authentication

0 引言

追踪攻击源是解决 DDoS 攻击最有效的方法。IPv4 目前有多种追踪技术[1],主要是包标记、日志记录、ICMP 消息追踪等。数据包标记主要分为两大类,一种是概率包标记算法,另一种是确定性包标记(DPM)算法[2]。DPM 算法由于只使用少量攻击包进行追踪,且具有重构简单及误报率低等优点,更具备一定的实用价值。

IPv6 协议虽通过设计 IPSec 提高了安全性,但由于 TCP/IP 协议的缺陷,也面临拒绝服务攻击的危险[3]。文献[4]依据概率包标记算法,提出记录路由器 hash 值进行回溯,但计算量较大;文献[5]用 IPv6 的扩展头标记实施概率包标记算法,但也需较大的计算量和较长的重构时间;文献[6]通过所有路由器签名认证来解决受控路由器的问题,但需要额外的开销进行签名认证。

以上都是基于 IPv6 下的追踪算法,主要是通过 IPv4 下的追踪技术并结合 IPv6 的新特性进行的研究,但对 DPM 算法的研究较少。因此本文对 IPv6 下的 DPM 算法进行了的研究,提出一种 IPv6 下基于 MAC 认证的改进确定包(ADPM-v6)算法,提高了标记效率,有效减少了重构路径所需包数,大大降低了重构计算量和收敛时间。

1 DPM 和 IPv6 简介

1.1 DPM 算法简介

DPM 基本思想[2]是在 IP 包从内网通过边界路由器时进行标记,将路由器内网口 IP 地址(入口地址)等分成两部分和 1 bit RF 域。受害者通过维护一个以源地址为索引的表,从表中得到对应同一个源地址的入口地址的两个分段,就可获得入口地址。

1.2 IPv6 简介

IPv6 协议将地址变为 128 bit,简化了包头结构。包头采用固定的 40 Byte,结构是基本包头加扩展包头的方式,用扩展包头的方式实现 IPv4 包头中可选字段的功用,以及 IPv6 协议的其他新特征,如图 1 所示。扩展头必须紧随在 IPv6 包头之后,它所携带的信息必须被传送路径上所经过的每一个节点检查和处理,每个路由器必须在处理 IPv6 包头的同时也处理其中的逐跳选项头。

收稿日期: 2011-09-26; **修回日期**: 2011-11-07 **基金项目**: 国家自然科学基金资助项目(61170277);上海市教委科研创新重点资助项目(12zz137);上海市重点学科建设资助项目(S30504)

作者简介:赵树枫(1976-),男,江苏淮安人,工程师,硕士,主要研究方向为下一代互联网、网络安全、网络管理(zhaosf@usst.edu.cn).

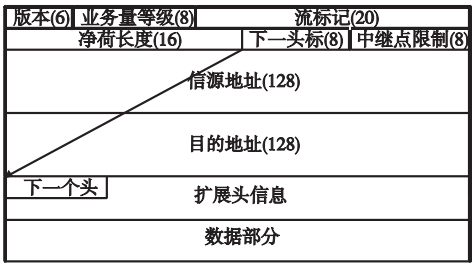

图1　IPv6包头结构图

2　基于 MAC 的 ADPM-v6 算法

2.1　IPv6 下 DPM 算法分析

根据 IPv6 包头的结构,一般有两块区域可被用来保存标记:流标签和逐跳选项扩展包头。流标签可以使用,但将 IPv6 的 128 bit 地址编码放进 20 bit 的流标签中,面临很大的计算开销。

本算法采用扩展包头中的逐跳选项头进行 DPM 入口地址记录。扩展头所携带的信息由路径上所有的节点进行检查和处理,且位于 IPv6 包头的后面。每个路由器处理 IPv6 包头的同时,也需处理逐跳选项头。如果 IPv6 包头中的下一个包头域的值为 0, 就表明此 IP 包中包含有逐跳选项头。最靠近源的路由器网络接口会把自己的 IPv6 地址标记到这个新选项中。如果对每个数据包记录完整的 IPv6 地址,那么一旦受害者知道自己正在被攻击,可直接从逐跳选项头中获得入口路由器的地址,从而实现实时追踪。IPv6 的 ADPM 实验环境原理如图 2 所示。

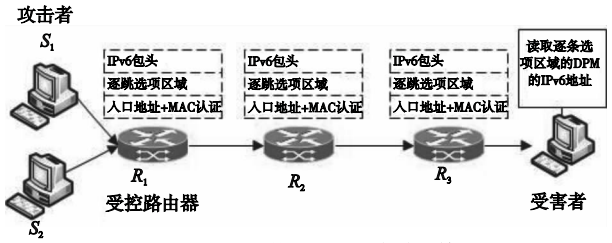

图2　IPv6的ADPM-v6实验环境原理

如果网络中存在受控路由器,可以使攻击者轻易地修改数据包标记,从而阻碍受害者正确恢复地址。IPv4 下,部分研究者提出了用认证的方式来解决该问题。数字签名是认证最有效的方法,但在 IPv6 下并不完全适用,因为 IPv6 地址增加后,基于 128 bit 地址的相关计算开销和带宽验证代价提高。为此,本文提出了基于改进 MAC 的认证方法,与数字签名相比,其计算和验证开销较小。

2.2　MAC 认证原理

IPv6 地址扩展为 128 bit,但 MAC 地址并没有扩展,MAC 继续用于双方消息认证,同时 IPv6 某些自动获得的地址后 64 bit 中就包含 MAC 地址。

首先假设通信双方 A 和 B 共享一个密钥 K, A 向 B 发送通信消息,通过将 A 的 MAC 地址 C_K 函数后得到的 MAC = C_K(SA,DA,IA),然后将 MAC 与消息 M 一起发送给 B。B 用相同的密钥对接收到的信息进行相同的计算得出新的 MAC,与接收到的 MAC 进行比较,若两者相等则消息来自于 A 且未被窜改。共享密钥问题通常采用时间片密钥来实现。

2.3　IPv6 攻击数据包大小分析

和 IPv4 不同,IPv6 为提高安全性,路由器不对数据包进行分片,所有的分片行为都在源端进行。这样可减少数据包处理机制的复杂性,避免信息被人截获窜改。如果改进算法存放标记后导致数据包超过 MTU,则会被路由器丢弃。

IPv6 的 MTU 为 65 535 Byte,当分组大于链路最大传输单元(MTU)时,源节点负责对分组进行分片, IPv6 包的不可分段部分包括 IPv6 包头、逐跳选项包头、目的地选项包头和路由包头。以太网上的 MTU 大小是 1 500 Byte,IPv6 包头是 40 Byte,ADPM-v6 算法标记使用逐跳选项空间,包含 128 bit 入口地址和 F_k 函数值,最大为 32 Byte。只要攻击数据包数据小于 1 428 Byte,路由器就不会丢弃数据包,从而使得本算法有效。ADPM-v6 的 MTU 构成如图 3 所示。

图3　ADPM-v6的MTU构成

因此需考虑攻击者采用的攻击数据包大小的分布情况。为有效检测 IPv6 下 DDoS 攻击时产生数据包大小的分布情况,采用熵值作为攻击检测的数据包大小分布的属性指标。通过观察数据包大小熵值的时间序列,可以获得数据包大小的分布变化。

$$H(t) = -\sum_t \left(\frac{n_l}{S}\right)\log\left(\frac{n_l}{S}\right) \tag{1}$$

设数据包大小为 l,n_l 是数据包时间的序号。定义熵的时间序列由一个指定大小的滑动观察窗内计算的熵 S,这个滑动观察窗的长度应取决于攻击流量的持续时间。用 S_0 表示数据包攻击流量的持续时间,在理想状态下, $q = S_0/S \approx 1$。若 q 太小,会导致异常或丢失;若 q 太大,攻击负载很大,也无法测试,因此设定至少 200 个数据包。无论数据包到达率是否恒定,都可以通过计算到来的数据包数量的差异,在单位时间内判断。假设 X_n 是间隔 n 时间的数据包数量,且 $\sum_n X_n = S$。假定 $\{X_n\}$ 是随机离散过程,即 $\mu = E[X_n]$,变量

$$\mathrm{var}[X_n] = E[(X_n - \mu)^2] \tag{2}$$

定义偏差参数 D 表达变化数据包到达的过程:

$$D = \mathrm{var}\left[\frac{x_n}{\mu}\right] = E\left[\left(\frac{x_n}{\mu} - 1\right)^2\right] \tag{3}$$

由此可得熵值 H 和数据包大小分布成正比。因此根据模拟实验获得的熵值大小,可得知攻击数据包分布情况。

2.4　基于 MAC 的 ADPM-v6 算法实施

ADPM-v6 算法采用 MAC 认证,使用 IPv6 包头逐跳选项的区域作为标记域。根据 IPv6 的新特性,利用逐跳选项进行包标记,当路由器的 DPM 接口收到 IP 数据包时,检查数据包有无包含逐跳选项头。若未含,则申请空间产生新的逐跳选项头,并将标记和 DPM 接口写入其中;若已含有逐跳选项,则直接将用于验证的标记信息写入。ADPM-v6 算法标记流程如图 4 所示。

逐跳选项包含 DPM 接口地址和标记信息。标记区域位于 DPM 接口地址后面,分成两块。传统的 MAC 认证方法需将地址分段后进行处理,因 IPv6 下逐跳选项有足够的空间存放标记信息,ADPM-v6 算法改进了 MAC 认证方法,只存放 MAC 函数值和其对应的 MAC 地址,从而减少了重构的计算量并有效地缩短了重构时间。

F_k 函数是以 K 为密钥的 MAC 函数,DPM 接口对包的源地

址 SA、目的地址 DA 以及自己的地址 IA 计算 MAC = F_K(SA, DA, IA)。DPM 接口收到一个 IP 包,用当前时间片对应的密钥 K,对 SA、DA 和 IA 计算得出 F_k,一并写入 IP 包头的标记域。

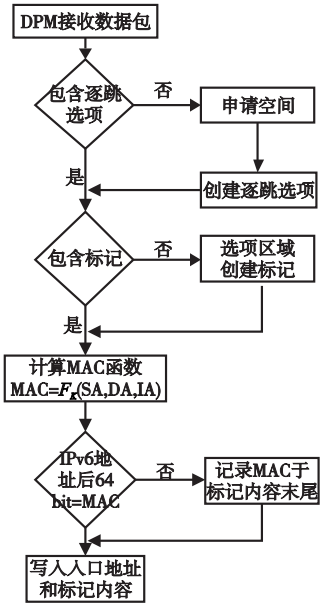

图4 ADPM-v6算法标记流程

IPv6 地址有的直接采用 MAC 地址作为 IP 后 64 bit 地址。ADPM-v6 算法标记前,对比 MAC 地址和 IP 后 64 bit,如果相同,则标记区域只记录 MAC 函数消息,并不记录 MAC 地址,从而简化标记空间,提高标记效率。

标记算法关键代码如下:

DPM 路由器接口标记

```
w. MAC = F_k(SA,DA,IA) //计算 MAC 函数
for each packet w
  if no_w_hop then
    insert_hop(w) //没有逐跳选项,则插入
  if MAC = IPv6_address_64
  //判断 MAC 地址是否和 IPv6 后 64 位相等
    hop_marking = F_k
  else
    hop_marking = F_k and MAC//定义标记信息
insert_marking(w_hop) //插入标记内容
write IA and hop_marking //写入入口地址和标记内容
```

认证确定包标记(ADPM-v6)算法重构过程分为两步:

a)标记记录。受害者端维护一个重构表 RecTbl,表包含到达时间、源地址、MAC 消息、入口地址、发送时间共五个域。每收到一个数据包,就将其相关信息写入到表中对应的域。

b)地址恢复。受害者根据 RecTbl 表中各条记录的到达时间推断出发送时间,并写入发送时间域。根据 RecTbl 表中相同的源地址、MAC 地址、发送时间得到密钥,然后通过 F_k 函数计算新的 MAC。如果得到新的 MAC 与 RecTbl 记录的 MAC 相同,那么该记录得到的是有效的入口地址,无须进行计算,直接读取 RecTbl 中对应的入口地址,将其写入 IngreeTbl 表中。

重构算法关键代码如下:

```
Let RecTbl be a table of tuples(arriving time,source address,MAC,IA address,sending time) /*设定 RecTbl 表 AT 为到达时间,SA 为源地址,MAC 为地址,IA 为入口地址,发送时间域*/
if hop_marking = F_k and MAC
  then MAC = hop_marking. MAC
  else MAC = IPv6_address_64 //得到 MAC
RecTbl. insert(AT,SA,w. MAC,w. IA,NULL)
//插入标记记录,发送时间为空
for each record r in RecTbl
  deduce r. sending-time according to r. arriving-time
//根据到达时间推断发送时间
for each rcd in RecTbl
  Get key K //根据时间片得到密钥 K
for each possible ingress address IA in rcd
  if F_k(SA,DA,IA) = rcd. MAC //通过 MAC 地址判断是否被窜改
    IngressTbl. insert(IA) // 插入正确的入口地址
```

3 仿真实验与结果性能分析

3.1 IPv6 攻击环境测试

为了验证攻击者通常采用的小数据包攻击,本文根据 DARPA 的数据集在 IPv4 下分析攻击包大小的分布,取一周数据,可以看出,IPv4 下攻击数据包主要集中在 0 ~ 64 Byte,如表 1 所示。

表 1 IPv4 下 DARPA 攻击数据大小分布

时间	0 ~ 63 Byte/%	64 ~ 255 Byte/%	256 ~ 511 Byte/%
星期一	66.25	20.5	2.35
星期二	74.5	12.4	3.85
星期三	79.15	15.2	2.05
星期四	77	9.35	3.85
星期五	75.8	10.5	3.55

DARPA 并没有对 IPv6 下的攻击数据进行统计。本文使用 TFN2K 软件模拟在 IPv6 下的攻击,并对其攻击数据包大小进行抓包分析。实验环境原理如图 2 所示,采用 SYN flood、ICMP flood 和 UDP flood 三种最常用的攻击类型,攻击采样时间分别为 2 h。

整个攻击过程的熵值 H 分布如表 2 所示。正常数据的熵值大于 1,攻击数据产生的熵值小于 1;$D >> 0.01$ 数据包为正常数据包的流量,$D << 0.01$ 数据包为攻击数据包的流量。根据熵值和数据包大小分布对应关系,结合 Sniffer 的抓包结果统计,数据包大小主要集中在 64 ~ 127 Byte 之间,约占总体流量的 70% 左右,如图 5 所示。实验说明,由于 IPv6 包头大小增加为固定 40 Byte,IPv6 下 DDoS 常用攻击方法主要集中 64 ~ 127 Byte。

表 2 熵值统计对比

对比项	流量类型			
	正常	SYN	ICMP	UDP
平均值 μ	6.21	5.11	8.3	8.9
变量 var[X_n]	4.5	0.0152	0.095	0.099
时间 D = var[X_n]/μ^2	0.14	0.0052	0.0018	0.0019
熵值 $H(t)$	1.13	0.23	0.45	0.49

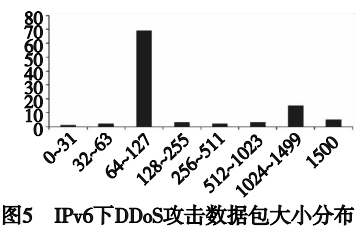

图5 IPv6下DDoS攻击数据包大小分布

从实验结果和表 1 的数据对比可以得出 IPv6 网络攻击中小数据包占了绝大多数,且数据包大小远远小于算法失效最大的 1 428 Byte,证明了 IPv6 下攻击通过小数据包进行并传播,因大数据包在传输过程中需源端进行分片和重组,容易被发现。所以改进算法标记后的数据包大小不会超过 MTU,从而不会被路由器丢弃,其有效且具有实用价值。

3.2　算法性能测试

为了测试 IPv6 下改进算法的性能,算法模拟实验使用 NS2 平台。NS2 不支持 DPM 算法,需对其进行扩展,并通过相应的补丁支持 IPv6 环境。扩展功能主要是在 hdr_ip 结构中增加标记域和在地址分类器中增加 DPM 标记和重构的方法。

实验环境原理如图 2 所示,S_1 为攻击者 2001: da8: 8004: 1: : 1,采用固定速率对地址为 2001: da8: 8004: 3: : 1 的节点 D 进行攻击,R_1 为受控路由器,地址 2001: da8: 8004: 2: : 1。当 S_1 和 S_2 发送数据到达 R_1 后,R_1 内网口收到数据包,写入入口地址和 MAC 标记信息,然后转发至 R_3,地址为 2001: da8: 8004: 3: : 2;R_3 接收数据包后,对其 MAC 进行验证,通过则将入口地址写到 IngressTbl 中。

实验主要通过包标记算法性能的两个基本要素,即收敛时间和误报率来进行定量分析。

1)基本 DPM 收敛时间　根据重构入口地址的概率由 $P = 1 - 0.5^k$ 计算,其期望值 $E(CT)$ 为

$$E(CT) = k \times (\frac{1}{k} + \frac{1}{k-1} + \cdots + 1)$$

基本 DPM 中,$k > 1$,$E(CT) > 1$;算法 ADPM-v6 中,$k = 1$,其期望值 $E(CT) = 1$。仿真实验证明只需一个数据包就可得到完整的入口地址,收敛时间大大缩短。受控情况下,节点 D 需计算 MAC 认证信息,判断数据是否被窜改。实验证明,随着攻击数目的增加,MAC 认证计算量逐渐增大导致时间增加,但也远远小于 DPM 算法。

2)基本 DPM 误报率 FPR　其等于不正确的入口地址数据包除以重构攻击源的数据包总量。仿真实验证明,ADPM-v6 算法通过一个数据包就可以携带完整的入口地址,不需要分片组合和验证计算。受控情况下,未通过 MAC 认证的伪造数据直接丢弃,因此算法的误报率为零。

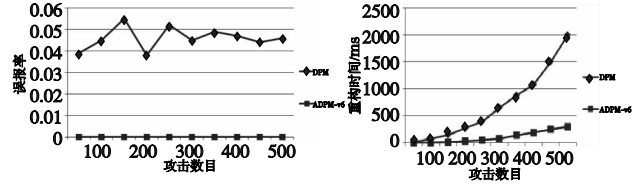

图6　重构时间及误报率实验对比

4　结束语

本文探讨了 IPv6 下实现 IP 追踪技术中的 DPM 算法,并提出基于改进 MAC 认证的 ADPM-v6 算法。对改进后算法进行了仿真实验测试,结果表明改进后的算法在 IPv6 下有效、适用,并大大减少了重构时间和所需数据包数量,提高了重构攻击路径的速度。算法只需通过改进的 MAC 验证,并从逐跳选项中读取入口地址,无须网络拓扑,即可快速追踪攻击源。同时对影响算法实用性的 IPv6 下攻击数据包的大小进行了分析和实验仿真。实验结果说明,IPv6 下常见攻击数据包大小不会超过 MTU,算法具有较强的实用价值。

参考文献:

[1] 黄忠厚,徐川,刘宴兵. DDoS 攻击源追踪算法综述[J]. 计算机应用研究,2010,27(9):3233-3236.

[2] BELENKY A, ANSARI N. On deterministic packet marking[J]. Computer Networks,2007,51(10):2677-2700.

[3] YANG Xin-yu, MA Ting, SHI Yi. Typical DoS/DDoS threats under IPv6[C]//Proc of International Multi-Conference on Computing in the Global Information Technology. Washington DC: IEEE Computer Society,2007:55-60.

[4] ALBRIGHT E, DANG Xuan-hien. An implementation of IP traceback in IPv6 using probabilistic packet marking[C]//Proc of International Conference on Internet Computing. 2005:416-421.

[5] 杨俊,王振兴,郭浩然. 基于扩展头随机标记的 IPv6 攻击源追踪方案[J]. 计算机应用研究,2010,27(6):2335-2337,2340.

[6] YANG Xin-yu, MA Ting. A link signature based DDoS attacker tracing algorithm under IPv6[J]. International Journal of Security and Its Applications,2009,3(2):27-36.

[7] SIRIS V A, STAVRAKIS I. Provider-based deterministic packet marking against distributed DoS attacks[J]. Journal of Network and Computer Applications,2007,30(3):858-876.

[8] BELENKY A, ANSARI N. Tracing multiple attackers with deterministic packet marking (DPM)[C]//Proc of IEEE Pacific Rim Conference on Communications, Computers and Signal Processing. [S. l.]: IEEE Press,2003:49-52.

[9] LIU J, LEE Z J, CHUNG Y C. Dynamic probabilistic packet marking for efficient IP traceback[J]. Computer Networks,2007,51(3):866-882.

[10] DURRESI A, PARUCHURI V, BAROLLI L. Fast autonomous system traceback[J]. Journal of Network and Computer Applications,2009,32(2):448-454.