基于扩展头随机标记的 IPv6 攻击源追踪方案

杨 俊, 王振兴, 郭浩然
(解放军信息工程大学 信息工程学院, 郑州 450002)

摘 要: 针对 IPv4 网络环境下的随机包标记法不能直接应用于 IPv6 的问题,依据 IPv6 自身的特点提出了一种基于扩展报头随机标记的 IPv6 攻击源追踪方案。对 IPv6 报头进行分析研究,选取了合适的标记区域及编码格式,对 PMTU 算法进行了修改以更好地满足追踪的需要,采用攻击树生成算法重构攻击路径。理论分析与仿真实验结果表明,该方案能够有效地重构攻击路径,实现对 IPv6 攻击源的追踪。

关键词: 网络追踪; IPv6; 随机包标记; 扩展包头; 分布式拒绝服务攻击
中图分类号: TP393.03 **文献标志码**: A **文章编号**: 1001-3695(2010)06-2335-03
doi:10.3969/j.issn.1001-3695.2010.06.097

IPv6 attack source traceback scheme based on extension header probabilistic marking

YANG Jun, WANG Zhen-xing, GUO Hao-ran
(*School of Information Engineering, PLA Information Engineering University, Zhengzhou* 450002, *China*)

Abstract: The probabilistic packet marking for IPv4 networks could not applied to IPv6 without any change. This paper proposed an IPv6 attack source traceback scheme based on extension header probabilistic marking according to the features of IPv6. It analysed the IPv6 header, selected the appropriate marking areas and gave the specification of coding, made some changes to PMTU algorithm so as to satisfy the demand of traceback, adopted the algorithm of generating attack tree to reconstruct the attack paths. Theoretic analysis and simulation results prove that it can reconstruct the attack paths effectively, realize the aim of IPv6 attack source traceback.

Key words: IP traceback; IPv6; probabilistic packet marking(PPM); extension header; DDoS

互联网升级和过渡到 IPv6 是必然的趋势。然而,IPv6 并未对现有网络体系结构产生根本性的改变,数据包的网络传输机制与 IPv4 基本相同,这就意味着类似于 IPv4 下的网络安全威胁将依然存在[1],DDoS 就是其中典型的攻击方式。

DDoS 通过耗尽服务器的资源使得合法用户无法获取正常的服务。入侵检测、防火墙等安全措施能比较有效地阻止此类攻击行为,但却无法确定攻击源的真实位置。这主要是 TCP/IP 允许主机自己填写源 IP 地址,并且在 TCP/IP 中也没有提供一种机制来验证源 IP 地址的真实性,因此需要一种安全机制,当攻击发生时能够确定攻击源的真实位置以及攻击路径,这就是攻击源追踪技术。

在攻击源追踪技术的发展过程中,先后出现了多种技术,主要应用于 IPv4 网络环境,包括入口过滤[2]、链路测试[3]、路由器日志法[4]、基于 ICMP 报文的 iTrace 方法[5]以及包标记法[6,7]。有关 IPv6 网络环境下的攻击源追踪的研究目前还较少。文献[8]提出了 IPv6 下基于改进的 SPIE 源追踪方案,作为路由器日志法在 IPv6 环境下的一种实现,该方案仍然需要较大的计算开销和路由器日志存储空间。文献[9]对 IPv6 环境下基于包采样和基于包标记的攻击源追踪策略作了一定的说明,提出可以使用 IPv6 基本报头中的通信流类别字段和流标签字段来存放标记信息,但是使用这两个字段作为标记信息的存放区域带有较大的局限性。

包标记法中的随机包标记法(PPM)因其在追踪速度、路由器开销以及追踪能力等方面都有较大的优势,是当前研究较多的追踪技术。其中比较有代表性的是 Savage 等人[6]提出的分段标记算法(fragment marking scheme, FMS)和 Song 等人[7]提出的高级标记算法(advanced marking scheme, AMS)。但由于 IPv6 在报头格式以及分片机制等方面都与 IPv4 有很大不同,IPv4 下的攻击源追踪技术不能直接应用于 IPv6,要将该技术应用于 IPv6 网络环境还需要进行较大的改进。本文以 IPv6 协议自身的特性为切入点,研究适合于 IPv6 环境的随机包标记攻击源追踪方案。

1 基于 IPv6 扩展报头的随机包标记法

随机包标记追踪技术的基本思想是:路由器以一定概率对经过的数据包进行采样,同时对数据包进行标记,标记内容包括攻击路径上任意两个相邻路由器的地址,受害者根据收到的标记信息恢复攻击路径。

1.1 标记区域的选择

包标记法一般选择报头中不常使用的字段作为标记信息的存放区域,这也是标记区域的选择原则。在 IPv4 下,标记区

收稿日期: 2009-10-19; **修回日期**: 2009-11-30

作者简介: 杨俊(1984-),男,湖北天门人,硕士研究生,主要研究方向为网络安全、IPv6 与下一代互联网(plajunjun@126.com);王振兴(1959-),男,河北晋州人,教授,博导,博士,主要研究方向为 IPv6 与下一代互联网、网络安全;郭浩然(1986-),男,河南桐柏人,博士研究生,主要研究方向为网络安全.

域通常是用于分片的标志域(identification)和偏移域(offset),这主要是由于数据包在途中分段处理的情况很少出现(不超过0.25%[10])。IPv6 与 IPv4 报头相比有很大不同,它由基本报头和扩展报头组成。要实现 IPv6 下的随机包标记,首先需要从 IPv6 报头中寻找合适的标记区域。图 1 显示了 40 Byte 的 IPv6 基本报头的格式。

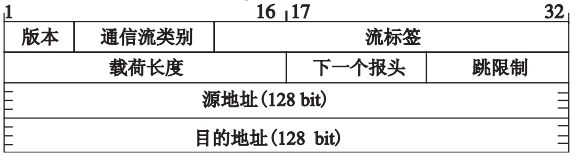

图1　IPv6基本报头格式

基本报头的八个字段为版本、通信流类别、流标签、载荷长度、下一报头、跳限制、源地址以及目的地址。可以看到,IPv6 基本报头的格式更加简洁,去掉了不需要的或很少使用的字段。

IPv6 简洁的基本报头格式也为寻找标记信息的存储区域带来了困难,在 RFC 2460 中除了通信流类别、流标签字段没有定义详细的使用细节,其余的字段都有其特殊的意义及用途而不能用于标记信息的存放。根据标记区域选择原则,20 bit 的通信流类别和 8 bit 的流标签字段可用来保存标记信息,但其使用带有局限性。首先是 28 bit 的标记空间相对 128 bit 的 IPv6 地址长度来说容量偏小,将 2×128 bit 的路径边信息编码成为 28 bit 的字段值,将耗费比较大的计算代价;其次是 FRC 2474 和 RFC 3697 已对两个字段作出说明,其使用也将逐步规范。因此,需要进一步从扩展报头中寻找标记区域。

IPv6 定义了多个扩展报头,能提供对多种应用的支持,同时又为以后支持新的应用提供了可能。RFC 2460 规定的所有 IPv6 节点必须支持的 IPv6 扩展报头有逐跳选项报头、目标选项报头、路由报头、片段报头、身份验证报头以及封装安全有效载荷。比较这些扩展报头,只有逐跳选项扩展报头是每个中间路由器都必须处理的惟一一个扩展报头,同时,该扩展报头能提供足够大的存储空间并且拥有更好的灵活性,在数据存储及格式编码上都存在优势,适合用来存放标记信息。

1.2　标记信息编码

确定逐跳选项扩展报头作为标记信息的存放区域后,需要定义标记信息的存放格式以及编码规范。本方案将标记信息存放区域作为逐跳选项扩展报头的一个选项,满足类型—长度—值(TLV)的编码格式,便于相关节点对其进行处理。图 2 显示了用于存储路径信息的逐跳选项扩展报头格式。

用三元组〈distance,ESAddr,EEAddr〉表示一条边。其中:距离域 distance 表示该边的起点距离受害者的跳数;ESAddr 和 EEAddr 分别记录边的头尾节点的全球 IPv6 地址,这里用未经压缩的 128 bit 的全球 IPv6 地址,可降低受害者重构攻击路径时的处理开销。

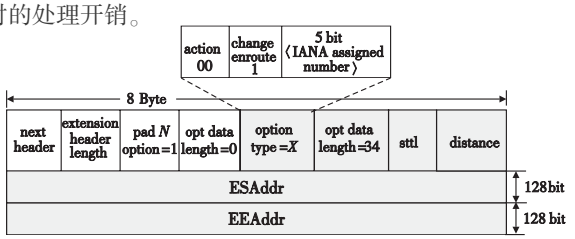

图2　标记信息存放域

在选项类型字段中,最高的两位定义为 00,表示当处理选项的路由节点不能识别选项的类型时应当跳过这个选项,可以避免用于存储标记信息的选项字段被干扰。

选项类型字段中的第三高位定义为 1,表示在通往目标的路径中,选项数据可以改变(0 表示不能改变),可以使选项字段中跳数域 distance 的值随着路由跳数的增加依次递增。

字段 sttl 表示数据包自第一次被标记后经过的路由器跳数,用于自适应边标记算法中标记概率的计算。

1.3　PMTU 发现算法修改

路由器在对采样的数据包进行标记时,在逐跳选项扩展报头中新增加了一个选项字段用做标记信息的存放区域,这会增加报文的长度,有可能导致报文的长度超过路径 MTU。由于 IPv6 路由器不支持报文的拆分,数据包将被丢弃,这会影响到标记信息的传送。为避免这种情况的发生,需要对路径 MTU 发现算法作出一定的修改,使得被标记过的数据包不会因为长度的增加而被丢弃。

首先需要确定数据包增加长度的字节数,分两种情况:a)被采样的数据本身含有逐跳选项扩展报头,这时只需增加一个长度 $l=16\times2+1+1+1+1=36$ Byte 的选项字段;b)被采样的数据本身不携带逐跳选项扩展报头,增加的长度除了 36 Byte 的选项字段,还包括 1 Byte 的下一报头字段,1 Byte 的报头扩展长度字段以及 3 Byte 的填充字节以满足 8 Byte 的对齐要求,增加的总长度为 $l=36+1+1+3=40$ Byte。选取其中较大的值 40 Byte 来重新计算路径 MTU,算法修改后如下:

```
M = max( PMTU,1280 ) - 40 Byte;
for every packet p{
  if p. size > M{
    fp[ i ] = Fragment( p,M );
    send( fp[ i ] );
  }
  else
    send( p );
  }
```

1.4　自适应的边标记算法

自适应的边标记算法因其在受害者重构攻击路径时拥有更好的收敛效果[7],是当前研究的热点。本文所提方案中,每个路由器以概率 r 采样其所转发的 IPv6 报文,先判断采样到的报文是否被标记过。如果没有被标记,则按照图 1 的格式构造逐跳选项扩展报头用于标记信息的存放;如果被标记过则重新计算标记概率为 r/Sttl,并以该值重新进行标记。本文称此标记算法为 V6PPM,其描述如下:

```
路由器 R 的标记过程:
for each packet p
  Select a random number r in the range[0,1)
if( p. distance = =0)
  Set p. ESAddr = R. address
  Set p. distance = p. distance + 1
  Set p. sttl = p. sttl + 1
else if ( p. distance = =1)
  Set p. EEAddr = R. address
  Set p. distance = p. distance + 1
  Set p. sttl = p. sttl + 1
else if( p. distance > 1&& p. EEAddr ! = NULL)
  Set r = r/p. sttl
  Set p. ESAddr = R. address
  Set p. distance = 1
  Set p. sttl = p. sttl + 1
else
  Set p. sttl = p. sttl + 1
```

1.5　攻击路径重组

当受害者 V 检测到有攻击发生时,会收集其所接收数据

包中的标记信息,这些信息将会组成一个边的集合 E,攻击路径的重组即是依据路径边集合生成完整的攻击拓扑图。本文在此提出基于攻击树生成算法的攻击路径重组方案。

通过分析可发现,在 DDoS 中,攻击路径在拓扑形状上表现为一棵树型结构,在此称之为攻击树。

攻击树定义:受害者 V 为树的根节点;攻击路径上除 V 的其余路由器节点构成 $m(m>0)$ 个互不相交的有限集合 $T_1,T_2,\cdots,T_m$。其中每个集合又是一棵树,称为根的子树。

攻击树的生成过程就是攻击路径的重构过程。其主要过程如下:

a)确定 V 为树的根节点。

b)在边集合 E 中找出所有尾节点为 V 的边$\langle 1,R',V\rangle$,将 R'作为攻击树中 V 的孩子节点。

c)在边集合 E 中找出所有尾节点为 R'的边$\langle 2,R'',R'\rangle$,将 R''作为攻击树中 R'的孩子节点。

d)重复以上过程,直到找出所有的边$\langle \max(d),R^*,R^{\#}\rangle$,将 R^* 作为攻击树的叶节点。

e)所得到的树 T 即代表了整个攻击路径。

攻击路径重组算法伪代码描述如下:

```
Set T as a tree with root V
Set S_d as an empty set where 1≤d≤max(d)
for d = 1 to max(d)
  for each packet p in P where p.Distance = = d
    if p not in S_d
      Add p to S_d
for each item w in S_1
    if w.EEAddr = = V.Addr
    Add w.EEAddr to T_1
for d = 2 to max(d)
for each item w in S_d
  for each item m in S_d-1
if w.EEAddr = = m.ESAddr
Add w.EEAddr to S_d
Output T
```

2　性能开销分析

攻击源追踪方法的优缺点主要体现在追踪速度、DDoS 追踪能力、事后追踪能力、路径重组开销[11]等方面。V6PPM 在实现数据包标记时采取对攻击路径边进行标记,由前面的路径重构算法可看到,V6PPM 能同时追踪多个攻击源。由于随机报标记法本身即具备事后追踪的能力,V6PPM 同样具备该特点。以下将主要就追踪速度和路径重组开销对 V6PPM 展开分析。

2.1　V6PPM 算法的收敛时间分析

追踪速度可以用时间衡量,但在目前的大多数研究中都采用报文数目作为衡量标准,也即算法的收敛时间。

收敛时间指的是受害者重构出完整攻击路径所需要收集的数据包的最小数目,计为 M。

设 r 为每个路由器对数据包的采样率,q 为距离受害者 d 跳远的路由器所标记的信息能被接收到的概率,则有

$$q_{\mathrm{V6PPM}}=r(1-r^2/2)(1-r^2/3)\cdots(1-r^2/d)=r\prod_{i=2}^{d}(1-r^2/i)$$

根据对 coupon collector problem 的求解,可以得到 V6PPM 算法的收敛时间 M 的期望值为

$$E(M)_{\mathrm{V6PPM}}\approx\ln(d)/r\prod_{i=2}^{d}(1-r^2/i)$$

而根据 ASM 算法有[7]

$$q_{\mathrm{AMS}}=r(1-r)^{d-1}$$

$$E(M)_{\mathrm{AMS}}\approx\ln(d)/r(1-r)^{d-1}$$

当 $d>1$ 时,显然有 $q_{\mathrm{AMS}}<q_{\mathrm{V6PPM}}$,进一步可以得到 $E(M)_{\mathrm{AMS}}>E(M)_{\mathrm{V6PPM}}$,也就是说,要重构出完整的攻击路径,V6PPM 算法需要收集的数据包数量更少。

2.2　路径重组开销分析

为了解决标记空间不足的问题,FMS 算法将边信息分段,分散标记在多个报文中。受害者重构攻击路径时首先要还原出路径边信息,当有多个攻击源时,片段重组非常耗时。

AMS 算法弥补了 FMS 的弱点,通过记录边的 hash 值,解决标记空间不足的问题,但其需要一个假设前提,即受害者 V 预先知道其上游的网络拓扑结构。以此上游拓扑为背景,受害者 V 将标记信息中包含的边的 hash 值与上游拓扑进行匹配比较来重构攻击路径。

V6PPM 算法利用了 IPv6 扩展报头灵活可扩展的特性,标记信息为 128 bit 未经压缩的 IPv6 全球地址。受害者 V 在重构攻击路径时,收集到的标记信息即为完整的路径边信息。与 FMS 算法相比,V6PPM 算法节省了路径片段重组的开销。相较于 AMS 算法,V6PPM 算法在标记信息时无须计算路径边的 hash 值,也无须知道上游的网络拓扑结构这样的过强假设,具有更好的实用性。

3　仿真实验

为了验证所提出的攻击源追踪方案在 IPv6 环境下的运行效果,以 NS-2(version 2.31)为仿真实验平台,分别实现了 IPv4 环境下的 AMS 算法和 IPv6 环境下的 V6PPM 算法。因为在每条攻击路径上路由器对数据包的采样是独立的,且所有路径重构时间与攻击源的个数呈线性比例关系[7],所以只选取了一条跳数为 15 的攻击路径作为实验对象。图 3 显示了路由器以不同的概率采样数据包时,攻击路径重构完成百分比与受害者接收数据包数量的对应关系。可以看到,IPv6 下的 V6PPM 算法达到了预期的追踪效果。

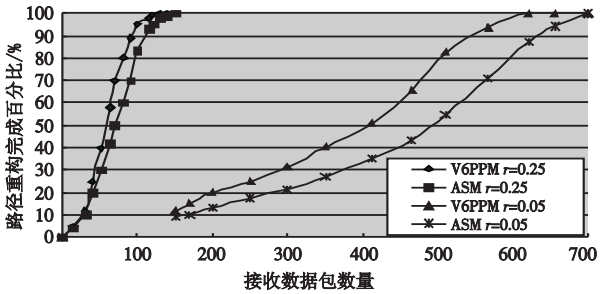

图3　ASM与V6PPM算法重构攻击路径比较

4　结束语

本文根据 IPv6 自身特点提出了一种适于 IPv6 环境下的攻击源追踪方案。理论分析与仿真结果表明,该方案在对 IPv6 攻击源进行追踪时能达到预期的效果。由于所提出的方案需要额外的空间来存储标记信息,势必会占用部分网络带宽,造成一定的影响,如何减小这种影响将是下一步努力的方向。

(下转第 2340 页)

行病毒检测的流程分为两部分:第一部分是行为分析器(实际就是分类器)的构造如图 3(b),这一部分实际是一个长期的过程,这个分析器也是处于长期更新的状态,从新的病毒中不断学习其模式,因此除了第一次称为分析器构造外,以后实际上是分析器更新,这一部分的执行过程是,已知病毒进行虚拟执行,得到病毒行为报告,然后据此构造分析器;第二部分是对病毒的检测如图 3(c),同样需要将病毒进行虚拟执行,得到病毒行为报告,然后据此利用构造好的分析器进行行为分析,得到病毒检测报告。

3.3 实验与结果分析

本文的实验方案在 Rieck 等人[3]的方案上进行修改而制定的。根据上述策略和实现算法,实验方案分为以下几个步骤:

a)收集病毒样本并打上标签;

b)对训练样本进行虚拟执行并监测行为,训练分类器(行为分析器);

c)对测试样本进行虚拟执行并监测行为,然后进行行为分析(即用分类器分类),得到分类准确率。

本文用病毒收集工具在互联网上收集了 10 类 10 000 个病毒(其分类标准源于文献[3]),基于这些样本,以数据库的形式建立了一个规范的小型病毒库,该病毒库的每一项对应一个病毒样本,包含序号、名称、家族等属性。所有的样本被分为训练样本和测试样本。其中 80% 用于训练,20% 用于测试。

按照上述实验方案对病毒样本进行了实验,得到图 4 所示的结果。

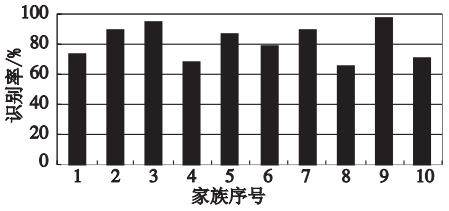

图4 病毒检测系统对10类病毒的识别率

图 4 显示的是采用了针对用户行为模式的主动防御策略的病毒检测系统对这 10 类病毒的识别率,横坐标表示 10 个病毒家族,纵坐标表示检测系统对相应家族病毒的识别率。可以看到,检测系统对绝大部分家族的病毒识别率都在 60% 以上,而个别家族的识别率几乎达到 100%。其中,家庭 3 和 9 的识别率是比较令人满意的。当然,家族 4 和 8 的结果不尽如人意,还存在很大的提升空间。

4 结束语

本文在病毒主动防御技术的启发下,基于已有的病毒行为分析和模式识别技术,提出了针对用户行为模式的病毒防御策略,将该策略应用到以病毒行为分析算法为核心的病毒检测系统中,并进行了测试。实验数据表明,该策略的应用能够使病毒检测系统作出比较准确的判断,从而证明了从用户行为模式出发对病毒实施主动防御的方法是可行的。

参考文献:

[1] 曹骞,樊晓平,谢岳山. 大型分布式管理信息系统的安全问题研究[J]. 计算机应用研究, 2007, 24(3):121-124.

[2] CHRISTODORESCU M, JHA S, KRUEGEL C. Mining specifications of malicious behavior[C]//Proc of the 6th Joint Meeting of the European Software Engineering Conference and the ACM SIGSOFT Symposium on the Foundations of Software Engineering. 2007.

[3] RIECK K, HOLZ T, WILLEMS C, *et al*. Learning and classification of malware behavior[C]//Proc of DIMVA. 2008.

[4] BAECHER P, KOETTER M, HOLZ T, *et al*. The nepenthes platform: an efficient approach to collect malware[C]//Proc of the 9th Symposium on Recent Advances in Intrusion Detection. 2006:165-184.

[5] BAILEY M, OBERHEIDE J, ANDERSEN V, *et al*. Automated classification and analysis of Internet malware[C]//Proc of the 10th Symposium on Recent Advances in Intrusion Detection. 2007:178-197.

[6] BAYER U, KRUEGEL C, KIRDA E. TTAnalyze: a tool for analyzing malware[C]//Proc of EICAR. 2006.

[7] BURGES C. A tutorial on support vector machines for pattern recognition [J]. Knowledge Discovery and Data Mining, 1998, 2(2): 121-167.

[8] 谈文明. 病毒防治技术的前沿地带[R/OL]. http://www.antiviruschina.org.cn/forum/zhjyzh_2002_virus/06-rising/virus.ppt.

[9] LEE T, MODY J J. Behavioral classification[C]//Proc of EICAR. 2006.

[10] CHRISTODORESCU M, JHA S. Static analysis of executables to detect malicious patterns[C]//Proc of the 12th USENIX Security Symposium. 2003.

(上接第 2337 页)

参考文献:

[1] 李振强,赵晓宇,马严. IPv6 安全脆弱性研究[J]. 计算机应用研究,2006,23(11):109-112.

[2] SUNG M,XU J. IP traceback-based intelligent packet filtering: a novel technique for defending against Internet DDoS attacks[J]. IEEE Trans on Parallel and Distributed Systems,2003,14(9):861-872.

[3] STON R. Center track:an IP overlay network for tracking DoS floods [C]//Proc of USENIX Security Symposium. 2000.

[4] LEE T H, WU Wei-kai, YAU T, *et al*. Scalable packet digesting schemes for IP traceback[C]//Proc of IEEE International Conference on Communications. 2004:1008-1013.

[5] MANKIN A,MASSEY D, WU C,*et al*. On design and evaluation of "intention-driven" ICMP traceback[C]//Proc of IEEE International Conference on Computer Communications and Networks. 2001:159-165.

[6] SAVAGE S,WETHERALL D. Practical network support for IP traceback[C]//Proc of ACM SIGCOMM Conference. 2000:295-300.

[7] SONG D,PERRING A. Advanced and authenticated marking schemes for IP traceback[C]//Proc of IEEE INFOCOMM Conference. 2001: 878-886.

[8] 占勇军,谢冬青,周再红,等. IPv6 下基于改进的 SPIE 源追踪方案[J]. 计算机工程与科学,2007,29(4):11-13.

[9] 周曜,刘耀宗,蒋道霞,等. 移动 IPv6 中的攻击源追踪问题研究[J]. 计算机应用研究,2008,25(3):903-905.

[10] STOICA I, ZHANG Hui. Providing guaranteed services without per flow management[C]//Proc of ACM SIGCOMM Conference. New York:ACM Press, 1999:81-94.

[11] 夏春和,王海泉,吴震,等. 攻击源定位问题的研究[J]. 计算机研究与发展,2003,40(7):1021-1027.