文章编号:1001-5132（2009）03-0348-06

基于 Snort 的 IPv6 协议分析技术的研究

陈　征，傅松寅，龚松春

（宁波大学 信息科学与工程学院，浙江 宁波 315211）

摘要：在讨论协议分析技术的基本原理、数据包封装与分解、以太网帧结构和 IPv6 数据包结构的基础上，进一步分析了 IPv6 下协议分析技术的变化，并着重研究了 IPv6 数据包解码. 结果表明：在 Snort 中添加 IPv6 解码模块，使 Snort 对 IPv6 协议具有初步的入侵检测功能.

关键词：IPv6；入侵检测；协议分析; Snort

中图分类号：TP393　　文献标识码：A

基于网络计算机系统的安全问题解决方案分为安全保护和入侵检测两大类[1]. 而入侵检测系统(Intrusion Detection System, IDS)作为主动式、动态地保护网络安全的有效手段，能够管理配置不当、用户误操作、软件漏洞等造成的攻击，它在 IPv6 环境下仍将继续发挥重要作用[2]. 与传统的模式匹配技术相比，协议分析技术在准确性和整体性等方面都具有一定优势. 因此，在处理IPv6包头时采用协议分析技术，而在处理数据部分时，采用成熟的模式匹配技术能够充分发挥二者的优点，提高监测效率.

1　协议分析技术原理

协议分析技术是新一代 IDS 探测入侵的主要技术，其基本思想是在目标地址属于受保护的网络中，将捕获到的数据包送往协议分析模块，并通过具体协议字段判断各层协议；而送往相应协议解析器解析数据包的数据部分，再与协议对应的特征库进行模式匹配，判断该数据包是否有入侵企图，最后由事件响应模块对该数据包做出相应的响应. 协议分析技术主要利用网络通信协议特有的高度规则性，对各层协议的解析结果进行逐层分析，从而快速探测攻击的存在[3]，与传统模式匹配技术相比有更多的优势.

协议分析技术能够智能地理解协议，有利于对数据包进行针对性的分析. 因为在每层协议上，它都沿着协议栈向上解析，可用所有已知的协议信息来排除所有不属于该协议结构的入侵[4]，从而避免传统模式匹配检测方法所做的大量无用功，减少匹配计算量，明显提升系统性能. 理论上，协议分析技术能够解决 IDS 领域长期以来的应用瓶颈问题、检测准确性以及大流量应用网络环境下的系统性能[4].

2　数据包封装与分解

在 TCP/IP 体系中，所有 TCP、UDP 及 ICMP

收稿日期：2009-04-10.　　宁波大学学报（理工版）网址: http://3xb.nbu.edu.cn
基金项目：浙江省教育厅科研项目（20070978）.
第一作者：陈　征（1965－），男，浙江宁波人，高级实验师，主要研究方向：计算机应用和网络安全. E-mail: chenzheng@nbu.edu.cn

协议包都以 IP 数据报格式进行传输.

当传送数据时, 则需进行数据包封装. 封装过程是将用户数据用协议来进行封装, 其中每层对收到的数据都要添加一些协议头信息. 首先用户数据由应用层协议进行封装(如 HTTP 协议), 而 HTTP 协议是基于 TCP 协议, 因此它被 TCP 协议进行封装, 然后再加上 IP 协议头, 构成 IP 数据包. 假设在以太网环境下, 那么最后就被封装成以太网帧, 这样就可通过物理介质进行传送.

当接收网络数据时, 就要进行数据包分解. 分解过程与封装过程恰恰相反, 从以太网帧中读出用户数据, 需要一层层地进行分解. 首先去掉以太网头和以太网尾, 再将剩下部分传递给 IP 层软件进行分解, 去掉 IP 头, 然后把剩下的传递给传输层(如 TCP 协议), 去掉 TCP 头, 剩下应用层协议部分数据包(如 HTTP 协议), 最后 HTTP 协议软件模块会进一步分解, 把用户数据给分解出来(如 HTML 代码), 这样的用户数据就可用浏览器进行浏览了. 其过程如图 1 所示.

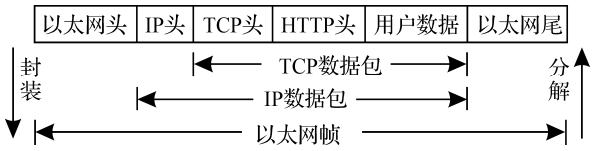

图 1　数据包封装与分解示意图

3　以太网帧结构

在以太网环境下, 数据被封装成以太网帧, 帧格式如图 2 所示, 包含 1 个头部与 1 个数据区.

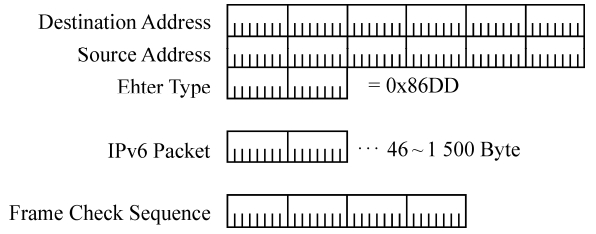

图 2　以太网帧封装格式

头部包括目的 MAC 地址(6 Byte)、源 MAC 地址(6 Byte)、以太网帧类型(2 Byte) 3 个字段, 即头部的大小与格式固定, 而数据区的大小由帧内要传输的数据决定, 因此所有以太网帧都有相同的头部格式. 也就是说, 可以通过分析以太网帧头部各域, 诸如源物理地址、目的物理地址或以太网帧类型信息(表 1)来决定如何处理该帧.

表 1　以太网帧类型的标识

上层协议(第 3 层协议)	帧类型字段
保留给 IEEE LLC/SNAP 使用	0000~05DC
IPv4	0800
CCITT 的 X.25	0805
ARP	0806
…	…
Novell 公司的 IPX	8137~8138
摩托罗拉公司	818D
IPv6	86DD
保留	FFFF

因此, 如果以太网帧中封装的是 IPv6 数据包, 则其帧头部 Ether Type 的值就是 0x86DD(图 2), 该帧也就是 IPv6 协议分析技术要处理的帧.

4　IPv6 数据包结构

IPv6 数据包结构如图 3 所示.

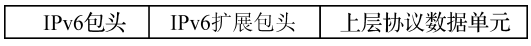

图 3　IPv6 数据包结构

IPv6 头通常由 1 个固定长度的基本包头加上 1 组可选、数量变化的扩展头组成, 也就是基本包头总是存在, 是每个 IPv6 数据包所必需的, 而扩展头则根据具体情况设置为可选项. 此外, 上层协议数据单元一般由上层协议头和有效载荷(如 ICMPv6、UDP 或 TCP)组成.

4.1　IPv6 头格式

根据 RFC2460 中对 IPv6 技术规范规定[5], 简化了包头格式, 除版本号字段外, IPv6 头与 IPv4 头的结构和字段都有很大的差别, 一些 IPv4 头字段被删除或者成为可选字段, 减少了需要检查和处理的字段的数量. IPv6 使用固定格式的包头[5], 如图 4 所示, 总长为 40 Byte, 包含 8 个字段域, 各域

的含义见表 2.

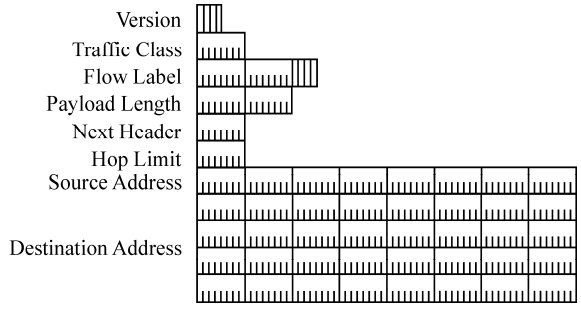

图 4 IPv6 头格式

表 2 IPv6 头各域的含义

域名	长度/位	含义
版本号	4	IP 协议版本号(IPv6 版本为 6)
传输类别	8	数据包优先级
数据流标签	20	标识那些需要 IPv6 路由器特殊处理的通信数据
有效载荷长度	16	IPv6 的负载大小(任何扩展头都被认为是有效负载的一部分)
下一个包头	8	标识紧接在 IPv6 包头后面的下一个包头的类型
跳数限制	8	数据包能经过最大路由跳数
源地址	128	包的制作者的地址
目的地址	128	包的预期接收者的地址(如果存在路由扩展头，可能是源路由列表的下一路由接口)

4.2 IPv6 扩展头

IPv6 定义了多种扩展头，取代 IPv4 头的可选字段，使IPv6能够提供多种支持，能更好地支持未来的扩展需求. 这些包头被放置在 IPv6 包头和上层协议包头之间, IPv6 头中的下个包头字段标明了下个扩展头，然后每个扩展头都有下个包头字段标明下个扩展头，最后 1 个扩展头的此字段标识负载内的上层协议. 根据 RFC2460、2402、2406、1700 技术规范规定，下个包头字段值标识扩展包头和上层协议包头类型，相应标识见表 3.

表 3 IPv6 包头中下个包头标识

	下一个包头类型	字段值
扩展包头	下一跳(Hop-by-Hop)选项包头	0
	目的(Destination)选项包头	60
	路由(Routing)包头	43
	分片(Fragment)包头	44
	AH(认证包头)	51
	ESP(封装安全载荷包头)	50
	最后一个包头、无下一个包头	59
传输层协议包头	TCP	06
	UDP	17
	ICMPv6(控制报文协议)	58

1 个标准 IPv6 的数据包不含扩展头，但如果中继路由器或目的要求特殊处理，可以添加 1 个或多个的扩展头，如图 5 所示.

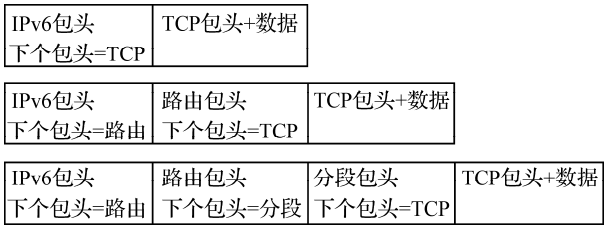

图 5 IPv6 扩展包头

在处理扩展包头时，必须严格按照它们在数据包中出现的顺序进行处理，例如数据包的接收者不能对整个数据包进行扫描，找 1 个特殊种类的扩展头而且优先于它前面的包头处理这个包头.

RFC2460 建议扩展头按以下顺序出现: IPv6 基本包头、下一跳选项包头、目的选项包头(当路由头存在时供中继目的使用)、路由头、分片头、认证头、ESP 头、目的选项头(供最终目的使用)、上层协议头，而且下一跳选项包头必须紧随 IPv6 包头出现. 每个扩展头最多只能出现 1 次，唯一例外的是目的选项包头，它可以在不同的地方出现 2 次，一次在路由包头之前，另一次在上层包头之前.

5 在 Snort 上实现 IPv6 协议分析技术

Snort 系统以开放源代码的形式发行，是个轻量级的网络入侵检测系统[6]. 具有实时数据流量分析和日志 IP 网络数据包的能力，能够检测各种不同的攻击方式，能够进行协议分析，对内容搜索、匹配，对攻击进行实时警报等.

Snort2.0 解码引擎包括数据链路层解码、网络

层解码和传输层解码[7]. 目前, Snort 支持的数据链路层协议包括 Ethernet、IEEE 802.11、Token Ring、FDDI、SLIP、PPP 等, 该部分不用修改. 但是 Snort 在网络层只能解析 IPv4 数据包, 对于 IPv6 不作解码, 仅将其用于统计; 为支持 IPv6, 必须对其进行扩展, 添加 IPv6 解码模块.

5.1　协议分析技术的变化

在 IPv4 中, 由于 IP 数据包头的长度较为固定, 只有选项部分会导致包头长度的变化, 而选项部分也由于路由器和主机支持较少而极不常用. 因此, 协议分析技术对数据包的分析过程较为简单, 只需根据包头首部长度的偏移量就可以直接找到 TCP 报文的位置. 而对于未加选项部分的数据包头来说, IP 数据包头的长度便是固定的 20 Byte. 而在 IPv6 下, 虽然 IP 数据包头的长度完全固定, 但由于扩展头的存在, 使得查找 TCP 报文时需要逐个读取扩展包头. 由于 TCP 报文本身并未发生变化, 在确定 TCP 报文的位置之后, 根据端口号判断协议类型, 并根据协议类型判断和查找特征字符串, 这在 IPv4 和 IPv6 下是完全相同的.

5.2　IPv6 解码

根据数据进入协议栈时的封装情况, 以数据链路层协议中占绝对优势的以太网为例, 根据上述分析的以太网协议规则规定, 以太网数据包在第 13 Byte 处开始, 包含 2 Byte 的网络层协议标志. 以太网解码模块利用该信息忽略捕获的数据包前 12 Byte, 直接跳到第 13 Byte 位置, 并读取 2 Byte 的协议标志, 如 0x0800 表示上层协议是 IPv4, 0x86DD 表示上层协议是 IPv6, 并根据此字段调用不同的解码函数, 以实现对 IPv4 和 IPv6 的解析[8].

IPv6 协议解码模块主要功能是实现 IPv6 数据包从网络层到传输层的解码. 模块首先将经以太网解码模块分解得到的 IP 数据包(去掉了以太网帧头部 14 Byte)套用 IPv6 包头结构, 然后根据上述分析, 利用 IPv6 协议规则, 读取 IPv6 头部版本信息域值, 进一步判断是否为 IPv6 数据包, 即该域值是否为 6, 若不是, 退出 IPv6 解码模块; 若是, 则读取 IPv6 包头第 5 Byte 开始处 2 Byte 的有效载荷长度域值, 进一步判断数据包自身标识的长度是否符合数据包的实际长度, 若不符则退出 IPv6 解码模块, 若相符则继续读取 IPv6 包头第 7 Byte 处 1 Byte 的下个包头域值, 该域标识扩展头和上层协议包头类型(表 3). 如果该域值标识的是扩展头, 则调用相应的扩展头解码模块继续直至最后 1 个头; 如果最后 1 个扩展头域值是 4, 则表明该包是 IPv4 in IPv6 隧道包, 则调用 IPv4 in IPv6 隧道包解析模块对其进行解析; 如果值是 47, 则调用 GRE (通用路由封装)解析模块[9]对其进行解析; 如果下个包头域值标识的上层协议包头, 若值是 58, 则调用 ICMPv6 解析模块对其进行解析; 若是 6 或 17, 则调用 TCP 或 UDP 解析模块对其进行解析. 对传输层(TCP/UDP)解析而言, IPv6 和 IPv4 没有差别.

5.3　在解析 IPv6 包头时的注意事项

(1) IPv6 地址是 128 位, 不能像 IPv4 地址那样用长整型表示, 应定义为数组结构; (2) IPv6 包头采用简化固定长度以加速路由器等网络设备的处理速度, 以及 IPv4 选项相应的部分在 IPv6 中通过扩展头的实现; (3)ICMPv6 在 IPv6 包头中用“next header=58”标识, ICMPv4 在 IPv4 中则用“protocol=1”标识, ICMPv6 和 ICMPv4 有很多不同, 必须为它们分别编写解码函数; (4)单从捕获的数据包本身分析, 无法知道 1 个包是否来自双栈节点, 或者是否经过了地址协议翻译, 只能将它当作普通的 IPv4 或者 IPv6 数据包进行处理.

5.4　IPv6 解码模块的具体实现

协议分析技术需要对每种协议编写一段解码检测代码[10], 即不同的协议有不同的解码检测代码, 但是所有协议分析都有类似的结构框架, 其检测思路是一致的, 即都遵照协议分析的检测思路. 首先将报文解码, 然后进行检查, 并与预期的值进行比较, 若有异常则报警[10]. IPv6 数据包解码主要是由函数 DecodeIPv6()来完成.

根据 RFC2460 中对 IPv6 包头结构定义，定义 IPv6 解码模块采用的 IPv6 包头数据结构如下:

```
typedef struct_IPv6
{
    u_int8_t ip_ver:4;   //版本号
    u_int8_t ope:8;      //传输类别
    u_int8_t fla:20;     //数据流标签
    u_int8_t ple:16;     //有效载荷长度
    u_int8_t nh:8;       //下一个包头
    u_int8_t jl:8;       //跳数限制
    struct in_addr ip_src;     //源 IP 地址
    struct in_addr ip_dst;     //目标 IP 地址
}IPv6;
```

以下是函数 DecodeIPv6()的主要实现过程:

```
void DecodeIPv6(u_int8_t *pkt, const u_int64_t len, Packet *p)
{
    /*定义变量: 数据包的长度、包头长等*/
    u_int32_t ipv6_len;
    u_int32_t hlen;
    /*将数据包结构套用 IPv6 包头的结构*/
    p->iph=(IPv6*)pkt;
    /*判断数据包长度合法性*/
    if(len<IP_HEADER_LEN)
      {
        ……提示出错信息
        p->iph=NULL;
        return;
      }
    /*判断是否为 IPv6 数据包*/
    if(p->iph->ip_ver！=6)
      {
        ……提示出错信息
        p->iph=NULL;
        return;
      }
    /*判断数据包的自身标识的长度是否符合数据包的实际长度*/
    ipv6_len=ntohs(p->iph->ple_len)+40;
    hlen=40;
    if(ipv6_len!=len)
      {
        ……提示出错信息
      }
    if(ipv6_len<hlen)
      {
        ……提示出错信息
        p->iph=NULL;
        return;
      }
    /*检查下一个包头，调用对应的扩展头或上层协议解码函数*/
    Switch(ntohs(p->iph->nh))
    {
      case IPHEAD_TCP:
        调用 TCP 解码函数 DecodeTCP()
        return;
      case IPHEAD_UDP:
        调用 UDP 解码函数 DecodeUDP;
      ……
      case IPHEAD_ROUTE:
        调用路由扩展头解码函数
      ……
    }
}
```

在实验室配置的IPv6网络环境中，经测试，添加了 IPv6 解码模块的 Snort，可以捕获到 IPv6 数据包，并能解析出包头各个域中的信息.

6 结束语

入侵检测系统是网络安全立体、纵深防御体系

中不可或缺的有机组成. 由于各种攻击的存在, 在网络攻防这个永恒的主题下, 对于入侵检测技术的研究将永远不会停止, 且会在下一代网络研究中更加完善和发展. 通过以上研究, 我们分析了在 IPv6 下协议分析技术的变化, 并着重讨论了 IPv6 数据包解码, 在 Snort 系统中添加 IPv6 解码模块, 使 Snort 对 IPv6 协议具有了初步入侵检测功能.

参考文献:

[1] 刘文涛. Linux 网络入侵检测系统[M]. 10 版. 北京: 电子工业出版社, 2006.

[2] 蒋建春, 冯登国. 网络入侵检测原理与技术[M]. 北京: 国防工业出版社, 2001.

[3] 庄绪春, 孟相如, 韩仲祥. 高速网络环境中入侵检测技术探讨[J]. 信息与电子工程, 2006, 4(4):288-291.

[4] 王艳秋, 赵昭灵, 兰巨龙. 一种基于 IPv6 的网络入侵检测系统[J]. 计算机应用研究, 2007(2):142-144/147.

[5] RFC2460. Internet Protocol, Version 6 Specification[S].

[6] Roesch M. Snort-lightweight intrusion detection for networks[EB/OL]. [1999-05-16]. http://www.snort.org/docs/lisapaper.txt

[7] Roesch M, Green C. Snort users manual 2.8.3[EB/OL]. [2009-1-16]. http://www.snort.org/docs/snort_htmanuals/htmanual_2832.

[8] Richard Stevens W. TCP/IP 详解——卷 1:协议[M]. 北京: 机械工业出版社, 2000.

[9] RFC2784. Generic routing encapsulation[S].

[10] 李建武, 卢选民, 侯新宇. 基于 IPv6 协议分析的网络入侵检测系统设计[J]. 计算机应用研究, 2005(12):135-137/140

Snort Based Study on IPv6 Protocol Analytical Techniques

CHEN Zheng, FU Song-yin, GONG Song-chun

(Faculty of Information Science and Technology, Ningbo University, Ningbo 315211, China)

Abstract: This paper discusses the basic principle of protocol analytical techniques, packet encapsulation and decomposition, Ethernet and IPv6 data packet structure. Further more, the paper analyses the changes of protocol analytical techniques under IPv6 environment, with stress on discussing IPv6 packet decoding. The preliminary intrusion detection function is finally embedded in IPv6 protocol by adding IPv6 decoding module in Snort.

Key words: IPv6; intrusion detection; protocol analysis; Snort

CLC number: TP393　　**Document code:** A

（责任编辑　章践立）