针对 IPv6 路由扩展首部的入侵检测机制 *

王相林, 刘立朋
(杭州电子科技大学 计算机学院, 杭州 310018)

摘　要：针对 IPv6 路由扩展首部漏洞所带来的安全问题,设计并实现了一种适用于 IPv6 路由扩展首部的入侵检测机制。以开源入侵检测系统 Snort 为基础,不改变其原有的数据包检测规则结构,采用 IPv6 协议分析技术对 Snort 数据包解析模块进行改进。设计实现了支持 IPv6 路由扩展首部的解析模块、内部网络受保护系统模块(IPSM),并给出了实验验证过程和结果分析。实验证明该设计方案能准确检测出利用 IPv6 路由扩展首部漏洞所实施的攻击行为。

关键词：IPv6; 路由扩展首部; 入侵检测; Snort; 协议分析
中图分类号：TP393　　**文献标志码**：A　　**文章编号**：1001-3695(2012)01-0322-04
doi:10.3969/j.issn.1001-3695.2012.01.089

Intrusion detection mechanism for IPv6 routing extension header

WANG Xiang-lin, LIU Li-peng
(*College of Computer*, *Hangzhou Dianzi University*, *Hangzhou* 310018, *China*)

Abstract: Aiming at the security issues caused by holes of IPv6 routing extension header, this paper designed and implemented an intrusion detection mechanism applicable to IPv6 routing extension header. Basing on the open source intrusion detection system Snort, instead of changing the structure of the existing Snort detection rules, using the technology of IPv6 protocol analysis, improved the packet analysis module of Snort. It designed and implemented the module supporting for parsing IPv6 routing extension header and the internal protected system module. It also gave the process of the proving experiment and the analysis of result. The experiment proves that the design scheme in this paper can exactly detect the attack behaviors which use holes of IPv6 routing extension header.

Key words: IPv6; routing extension header; intrusion detection; Snort; protocol analysis

0　引言

IPv4 网络逐步向 IPv6 网络过渡已经成为一种趋势。在 IPv6 网络中 IPSec 不再是协议的补充,而是作为 IPv6 自身功能的一部分,使得 IPv6 网络层的安全性有所增强[1]。但是随着研究的进一步深入,IPv6 协议的一些缺陷也逐渐被人们所发现,并提出了一系列相应的解决方案。IPv6 路由扩展首部存在一定的漏洞,它的滥用使得网络更容易遭受协议欺骗攻击。利用 IPv6 路由扩展首部,攻击者可以绕过受保护网络的防火墙和入侵检测系统,入侵到网络内部受保护的主机[2]。

针对这一安全漏洞,基于 Snort 开源入侵检测系统,采用 IPv6 协议分析技术,设计并实现了一种简单且有效的解决方案,并进一步提出了联合 Snort 和 Linux IPv6 网络协议栈以实现入侵防御的研究方向。

1　IPv6 路由扩展首部存在的安全漏洞分析

IPv6 中的路由扩展首部是用来指出 IPv6 分组从源节点到目的节点的过程中需要经过的一个或多个中间节点[3],替代了 IPv4 中实现的源选路。应用了 IPv6 路由扩展首部的例子,如图 1 所示。

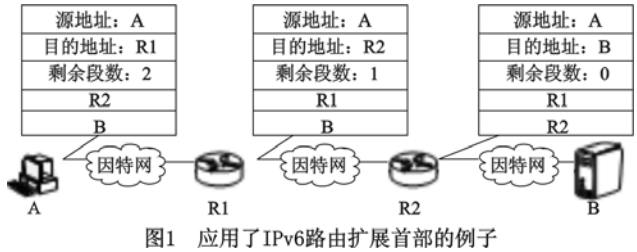

图1　应用了IPv6路由扩展首部的例子

主机 A 访问服务器 B,中间要经过路由器 R1、R2,从每个节点发出的数据包都要在 IPv6 固定首部的目的地址字段中指定下一个要经过的节点地址,并将剩余段数值减 1。例如,从 A 发出的 IPv6 数据包中,目的地址字段填充 R1 的地址,路由扩展首部中剩余段数字段的值为 R1 之后还要经过节点数目 2,即 R2 和 B。

由于现在的入侵检测系统还不支持对 IPv6 路由扩展首部的检测功能,使得一些利用路由扩展首部安全漏洞进行的协议欺骗攻击很容易成功。利用 IPv6 路由扩展首部安全漏洞的攻击实例如图 2 所示。

内网中数据服务器为受保护区域,禁止外部网络访问;Web 服务器允许外部网络访问,同时它也可以访问内网中受

收稿日期：2011-05-07; **修回日期**：2011-06-13　　**基金项目**：国家自然科学基金资助项目(60971046,60804064);浙江省大学生新苗人才计划科技成果推广项目(2010R407052);浙江省大学生新苗人才计划创新创业孵化项目(2010R407060)

作者简介:王相林(1953-),男,河南舞钢人,教授,主要研究方向为网络安全、IPv6 技术、网络性能评价;刘立朋,男,硕士,主要研究方向为计算机网络技术(llphdu@126.com).

保护区域。Snort 设置了 alert 告警规则 alert ip $ EXTERNAL_NET any→数据服务器 any (…)。该规则的意思是当有源地址是外网地址,目的地址是数据服务器地址的 IPv6 数据包经过时,Snort 产生告警信息。根据规则设置,数据包 A 为合法数据包,数据包 B 为非法数据包,而包含了路由扩展首部的 IPv6 数据包 C 本应该是非法数据包,由于现有的 Snort 还不具备检测 IPv6 路由扩展首部的能力,所以数据包 C 经过时 Snort 不会产生告警信息。当数据包 C 到达 Web 服务器之后,Web 服务器经过路由扩展首部的处理,得出下一节点是数据服务器,于是将数据包 C 转发给数据服务器。由于 Snort 并没有产生告警信息,网络管理人员也就很难发现有来自外部网络的数据包访问了内部的数据服务器,假如数据包 C 中存在恶意代码,这将会是十分危险的。

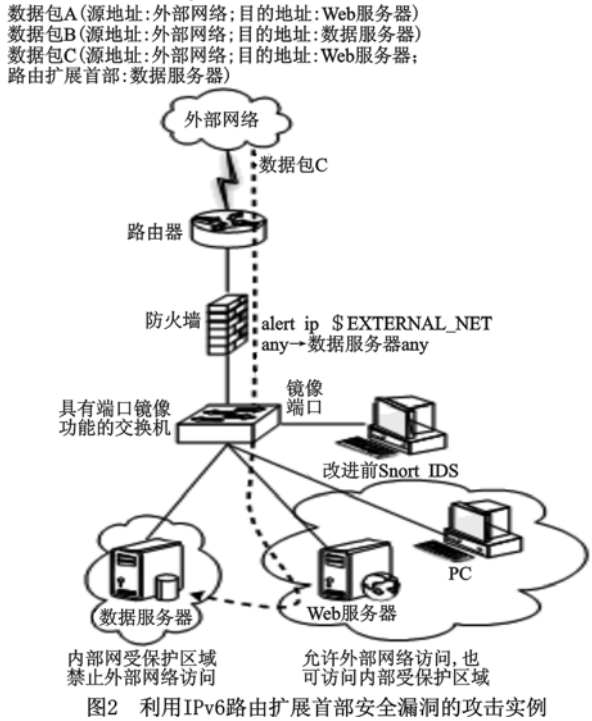

图2　利用IPv6路由扩展首部安全漏洞的攻击实例

针对 IPv6 路由扩展首部的安全漏洞,一些研究人员提出针对路由报头建立过滤规则[4],即更改现有的数据包检测规则结构,在规则头部分增加对路由扩展首部地址列表字段的支持。在 IPv4 中数据包的检测规则是基于〈源地址,源端口号,目的地址,目的端口号,协议类型〉这个五元组来设计的,IPv6 中则需要检查源地址和目的地址。要实现在检测规则头部分增加对路由扩展首部地址列表字段的支持,需要修改整个检测引擎模块的代码,这对于升级现有的入侵检测系统来说难度和工作量都会很大,所以此方案不被推荐。协议分析技术作为目前最流行的网络入侵检测关键技术之一,它能够提供很高的检测准确率和系统性能[5]。本文采用 IPv6 协议分析技术,在不改变原有数据包检测规则结构的前提下,对 Snort 数据包解析模块进行改进,设计并实现了一种简单且有效的解决方案。

2　IPv6 路由扩展首部漏洞解决方案设计与实现

2.1　Snort 各模块间的关系

Snort 各模块间的关系如图 3 所示。由图 3 可知,Snort 还支持数据包解析模块告警功能和预处理器模块告警功能。鉴于修改 Snort 检测引擎模块所面临的实际困难,本文的解决方案采用 IPv6 协议分析技术,将 Snort 对 IPv6 路由扩展首部安全漏洞的检测功能放在数据包解析模块来实现。Snort 从 2.8.0 版本开始,提供了对 IPv6 协议的支持[6],不过目前仅支持 IPv6 固定首部的解析功能,而对 IPv6 扩展首部则采取了相应的默认处理。本方案通过在 Snort 数据包解析模块添加路由扩展首部的解析函数 DecodeIPV6Route0()来实现 Snort 对路由扩展首部的解析支持。同时,设计了一个新功能模块,称为内部受保护系统模块(internal protected system module,IPSM),该模块的功能是向 Snort 数据包解析模块传递网络内部受保护主机的地址参数

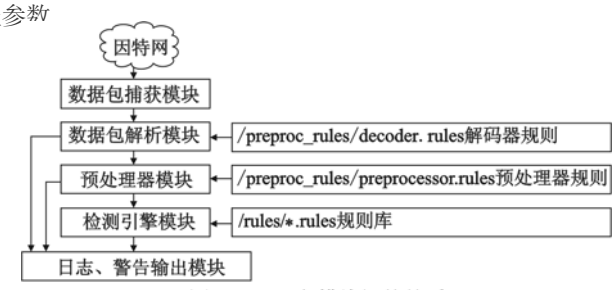

图3　Snort各模块间的关系

2.2　支持 IPv6 路由扩展首部的解析模块设计

Snort 数据包解析模块包括数据链路层解码、网络层解码和传输层解码[7],由源代码文件中的/src/decode.h、/src/decode.c 这两个文件组成,其中/src/decode.h 中定义了各种报文格式的数据结构。本方案中 IPv6 路由扩展首部的数据结构就定义在此文件中,如下所示。

```
typedef struct _IP6Route0
{
    uint8_t ip6rte0_nxt;
    uint8_t ip6rte0_len;
    uint8_t ip6rte0_type;
    uint8_t ip6rte0_seg_left;
    uint8_t ip6rte0_reserved;
    uint8_t ip6rte0_bitmap[3];
    struct in6_addr ip6rte0_addr[1];
} IP6Route0;
```

文件/src/decode.c 是各种协议解码函数的实现,本方案设计的函数 DecodeIPV6Route0()就添加在此文件中。Snort 通过 Libpcap 驱动程序从网卡捕获数据包后,通过函数 PcapProcessPacket()调用 ProcessPacket()链接到数据包解析模块,由函数指针 grinder 指向具体的解析函数。Snort 解析 IPv6 数据包的函数间调用关系如图 4 所示。

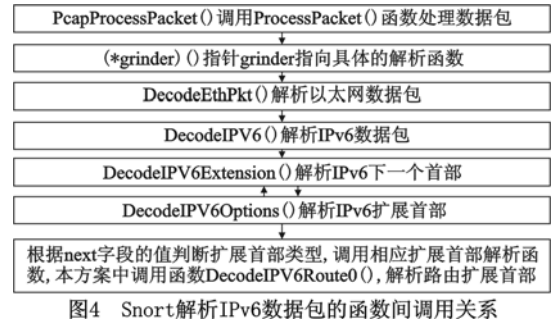

图4　Snort解析IPv6数据包的函数间调用关系

函数 DecodeIPV6Extensions()的功能是解析 IPv6 的下一个首部,包括上层协议首部和扩展首部。通过判断“下一个首部(next)”字段的值来调用相应的解析函数,如果下一首部为扩展首部,则调用函数 DecodeIPV6Options()来处理。函数实

现思路如下：

```
void DecodeIPV6Extensions(uint8_t next, const uint8_t *pkt, uint32_t len, Packet *p)
{…
switch(next)
{
  …
   case IPv6 扩展首部：
   DecodeIPV6Options(next, pkt, len, p);
}
}
```

函数 DecodeIPV6Options( ) 通过判断“扩展首部类型(type)”的值来调用相应的扩展首部解析函数。本文中所涉及到的是 IPv6 路由扩展首部，所以会调用函数 DecodeIPV6Route0( ) 来处理。如果存在下一个首部则循环调用函数 DecodeIPV6Extensions( ) 来处理。函数实现思路如下：

```
void DecodeIPV6Options(int type, const uint8_t *pkt, uint32_t len, Packet *p)
{…
switch (type)
{
  case IPPROTO_ROUTING:
  DecodeIPV6Route0(pkt, len, p);
}
}
```

函数 DecodeIPV6Route0( ) 的功能是解析 IPv6 路由类型为 0 的路由扩展首部，检测利用该类型路由扩展首部漏洞实施的攻击行为。设计思想是利用 IPv6 协议分析技术，首先获取路由 0 型路由扩展首部的内容，然后根据协议字段的高度有序性，分别获取路由类型字段和路由扩展首部长度字段的值。如果该路由扩展首部为路由 0 型扩展首部并且扩展首部长度在合法范围内，则循环提取路由地址列表中的地址，依次将提取出的地址与源地址、环回地址、内部网络受保护主机地址进行比较，如果结果相同，则触发 Snort 数据包解析模块告警规则，产生相应的告警信息。DecodeIPV6Route0( ) 函数流程如图 5 所示。

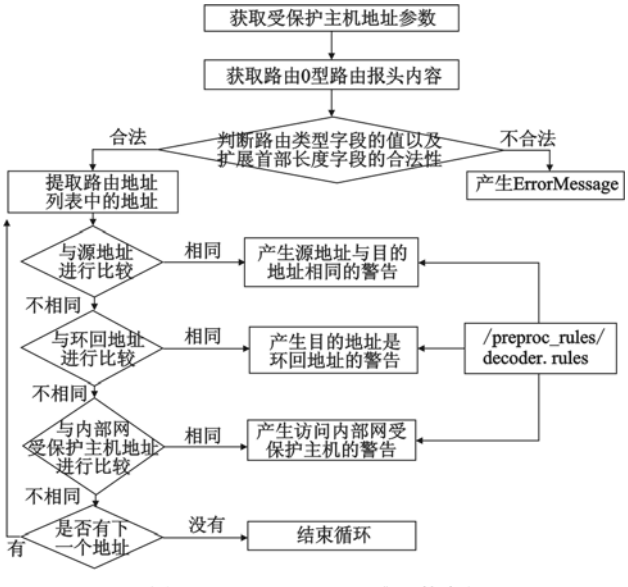

图5　DecodeIPV6Route0()函数流程

其中，受保护主机地址参数通过在 Snort 启动初始化阶段由管理员输入内部受保护系统模块传递进来。在文件/preproc_rules/decoder. rules 中添加了针对访问内部网络受保护主机的解析告警规则，如下所示：

```
alert(msg:"DECODE_ACCESS_PROTECTED_INTERNAL_IP";sid:10;gid:116;rev:1;metadata:rule-typedecode;classtype:protocol-command-decode;)
```

在文件/src/generators. h 中添加宏定义，如下所示：

```
#define DECODE_ACCESS_PROTECTED_INTERNAL_IP 10
#define DECODE_ACCESS_PROTECTED_INTERNAL_IP_STR
"(snort_decoder) Ah... You are trying to access the protected internal system. Catch you!"
```

2.3　内部受保护系统模块(IPSM)设计

IPSM 的功能是从外部传递内部网络中受保护主机的地址参数到 Snort 数据包解析模块。该模块对应的源代码文件是/src/internalsystem. h 和/src/internalsystem. c，具体实现采用 Libnet 库和 GTK + 2.0 库编写。该模块通过接口函数 GtkEntryInternalSystemIp( ) 向 Snort 主程序传递参数，SnortMain( ) 函数在执行 Snort 初始化工作时会调用该函数来加载 IPSM。

IPSM 实现的关键问题是如何将字符串形式的 IPv6 地址转换成网络字节形式，它是通过调用 Libnet 库中的封装函数 libnet_name2addr6( ) 来实现的，函数思路如下：

```
void text_convert_to_addr6(GtkWidget *widget, gpointer data)
{…
/*初始化句柄*/
l = libnet_init(LIBNET_LINK, /* libnet 类型 */device, /* 网络设备 */error_buffer /* 错误信息 */);
/* 地址转换*/
protectIpv6Address = libnet_name2addr6(l, (char *)protect_ipv6_addr_entry_text,LIBNET_RESOLVE);
/* 销毁句柄 */
libnet_destroy(l);
}
```

添加了 IPSM 的 Snort 体系结构如图 6 所示。

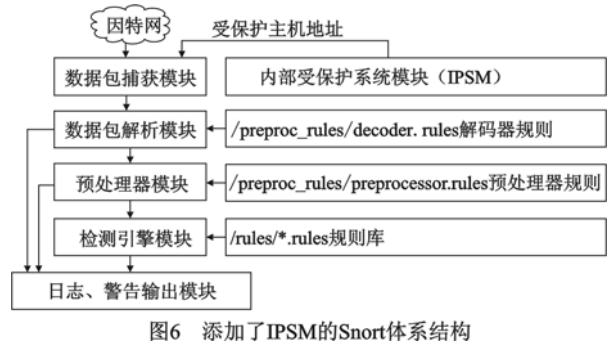

图6　添加了IPSM的Snort体系结构

3　验证实验设计与分析

验证实验网络拓扑如图 7 所示。由于实验条件限制，验证实验中采用一台安装有 IPv6 数据包构造和发送程序(IPv6Sender)的 Linux 主机模拟一台中间路由器，Snort 采用改进之后的版本。

IPv6Sender 构造并发送带有路由扩展首部的 IPv6 数据包，类型如下：

```
源地址(外部网络)：
2001:0410:0000:0001:0000:0000:0000:000A;
目的地址(Web 服务器)：
```

2001:0410:0000:0001:0000:0001:0000:000B;

路由扩展首部地址列表(数据服务器):

2001:0410:0000:0001:0000:0001:0000:000C;

首先在主机 A 上以入侵检测模式启动 Snort,命令(root 权限)如下:

```
$ /opt/snort-inline-gtk/bin/snort -v -c
/opt/snort-inline-gtk/etc/snort.conf
```

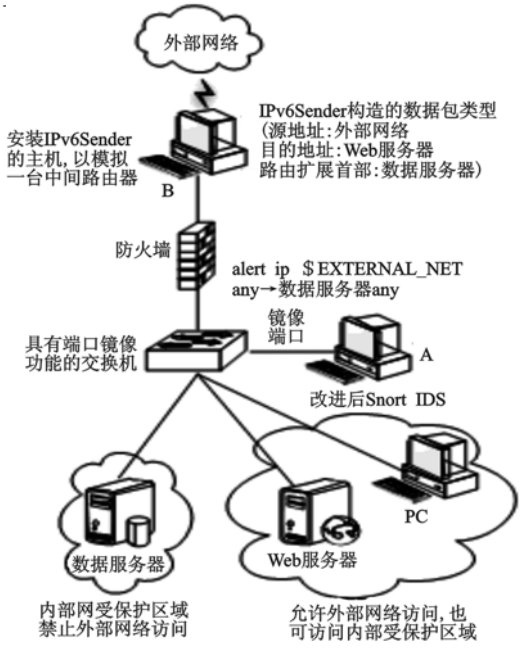

图7　验证实验网络拓扑

Snort 在初始化阶段,加载新添加模块 IPSM,弹出对话框提示管理员输入受保护主机的地址,IPSM 界面如图 8 所示。然后在主机 B 上运行 IPv6Sender 程序,IPv6Sender 界面如图 9 所示。

图8　IPSM界面

图9　IPv6Sebder界面

上述类型的数据包经 IPv6Sender 程序发送,被改进后的 Snort 捕获到,在进行数据包解析时,准确检测到了攻击行为,产生告警信息。Snort 检测到利用路由扩展首部漏洞的攻击如图 10 所示。

以上结果与预期的目标一致,表明改进后的 Snort IDS 可以准确检测到针对 IPv6 路由扩展首部安全漏洞所实施的攻击行为。目前由于条件限制,本文只在实验室环境下进行了系统测试,今后还将进一步部署在 IPv6 实验床中对检测的准确性进行验证。

另外,本文还利用 Linux 下的 KProf 工具分析了该检测机制对 Snort 系统整体性能带来的影响。KProf 通过分析程序运行过程中各个函数消耗的时间及每个函数在程序运行过程中所耗时间的百分比来帮助分析程序的性能。该检测机制涉及的主要函数是 DecodeIPV6Route0( ),因此,通过分析此函数在 Snort 运行过程中所耗时间百分比就可以估计出该检测机制对 Snort 系统整体性能带来的影响。实验中 IPv6Sender 程序发送了 100 个带有 IPv6 路由扩展首部漏洞攻击特征的数据包,Snort 均准确检测到并发出告警信息。用 KProf 分析 Snort 运行过程中 DecodeIPV6Route0( )函数的运行状况如图 11 所示。

图10　Snort检测到利用路由扩展首部漏洞的攻击

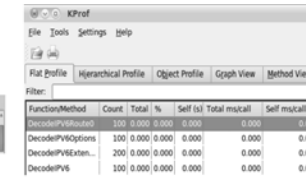

图11　DecodeIPV6Route0函数的运行状况

由图 11 可知,在整个运行过程中 DecodeIPV6Route0( )函数一共被调用了 100 次(Count 栏所示),所消耗的时间及所占百分比均为 0(Total(s)丨% 栏所示)。由于 KProf 工具是以 s 为单位统计函数运行消耗的时间,所以上述结果表示 DecodeIPV6Route0( )函数运行 100 次所需要的时间还不到 1 s。

由上述分析得知,本文设计的针对 IPv6 路由扩展首部的入侵检测机制同样具有高效性,对 Snort 系统整体性能影响小。

4　结束语

本文针对 IPv6 路由扩展首部安全漏洞,设计并实现了一种简单且有效的解决方案。该方案的设计思想是利用 IPv6 协议分析技术,在 Snort 数据包解析模块实现对 IPv6 路由扩展首部的检测功能,从而避免了修改原有的数据包检测规则结构。实验证明该方案能有效检测到针对 IPv6 路由扩展首部安全漏洞所实施的攻击行为。改进后的 Snort IDS 可以广泛应用于 IPv6 网络环境中,准确监测受保护的网络。由于 IDS 是旁路在网络上,当它检测到攻击行为时,攻击可能已到达目标造成损失,所以在本文基础上进一步的研究考虑是:实现 Snort 与 Linux IPv6 网络协议栈的联合,通过利用 Linux 内核中的 ip6_queue 或者 netfilter_queue 机制,改进 Snort 数据包捕获模块,将 Snort 串接在网络链路中,当检测到攻击数据包时自动丢弃,避免被攻击目标遭受损失,以达到入侵防御的目的。

参考文献:

[1] SZABOLCS S, PETER R. Will IPv6 bring better security[C]//Proc of the 30th EUROMICRO Conference. Piscataway: IEEE Computer Society, 2004: 532-537.

[2] PHILIPPE B, ARNAUD E. IPv6 routing headers security[EB/OL]. (2007-04-23)[2011-05-05]. http://www.secdev.org/conf/IPv6_RH_security-csw07.pdf.

[3] 刘超. IPv6 路由报头安全隐患解析[C]//第十七届全国信息保密学术会议论文集. 北京: 金城出版社, 2007: 287-293.

[4] SAVOLA P. Security of IPv6 routing header and home address options [EB/OL]. (2002-12-26)[2011-05-05]. http://tools.ietf.org/id/d-raft-savola-ipv6-rh-ha-security-03.txt.

[5] YAO Guo-xiang, GUAN Quan-long, LIN Liang-chao, *et al*. Research and implementation of next generation network intrusion detection system based on protocol analysis[C]//Proc of ISECS International Colloquium on Computing, Communication, Control, and Management. Piscataway: IEEE Computer Society, 2008: 353-357.

[6] 王相林, 李蓓蕾. 基于 Snort 的 IPv6 入侵检测技术[J]. 计算机工程, 2010,36(8): 144-146.

[7] 李振强, 徐一元, 马严. 基于 Snort 的 IPv6 入侵检测系统的研究与实现[J]. 电信科学, 2005,21(8): 32-36.