

自适应散列映射的弱跳完整性研究

高 磊, 张德运, 赵东平, 郑卫斌

(西安交通大学电子与信息工程学院, 710049, 西安)

摘要: 提出了自适应散列映射的弱跳完整性校验方法(AHMWHI), 该方法的思想是: 先校验其他数据包, 而将校验周期内的大数据包进行散列映射, 即将大数据包缓存; 当散列表产生冲突时, 将数据包序列的有态信息和大数据包的校验信息发送至下一跳进行校验; 根据当前网络吞吐率自适应调整散列表长度, 使校验周期内的大数据包时延满足期望值. 理论分析和实验结果表明, AHMWHI 解决了大数据包无法封装上 X 字节校验信息的问题, 改进了在弱跳完整性校验中对数据包重放和丢失的检测功能. 在测试示例中, 当时延期望值设为 5 ms 时, 大数据包的平均校验时延小于 019 ms.

关键词: 跳完整性; 自适应散列; 完整性校验

中图分类号: TP393 **文献标识码:** A **文章编号:** 0253 - 987X(2004)12 - 1232 - 04

Research on Adaptive Hash Mapping of Weak Hop Integrity

Gao Lei, Zhang Deyun, Zhao Dongping, Zheng Weibin

(School of Electronics and Information Engineering, Xi an Jiaotong University, Xi an 710049, China)

Abstract: The method of adaptive hash mapping of weak hop integrity (AHMWHI) was proposed. The main idea is as follows: hash mapping is performed in the datagram during verifying cycle time, i. e. caching the big datagram, and verifying the integrity of other datagram; the state information of datagram sequence and verification information of the big datagram are sent to the next hop router for verifying when the hash table collision occurs; the length of hash table is adaptively adjusted by current network throughput to make the delay of the big datagram within verifying cycle satisfying the expectation value. Theoretic analysis and experiments demonstrate that the problem that the big datagram packeted with X bytes verification can't be sent is solved by AHMWHI; the data replay and lose testing functions in weak integrity verification are improved. In demonstrations, the average delay of most big datagram is less than 019 ms while the expectation value is set to 5 ms.

Key words: hop integrity; adaptive hash; integrity verifying

跳完整性^[1]可保证所有数据包的完整性, 保证校验所有路由器之间的数据包, 保证在第一跳就发现 DOS 攻击下的伪造数据包并予以阻止, 而且阻止攻击能做到精确定位. 文献[1]提出了强、弱 2 种跳的完整性模型, 它们对每个数据包封装的上 X 字节校验信息进行校验. 但是, 任何类型网络上的数据包都有最大传输单元(MTU)限制, 该限制会使大小为 $[U_{MTU} - X + 1, U_{MTU}]$ 的数据包无法封装上校验

信息, 此类数据包被称之为大数据包. 为此, 本文在弱跳完整性校验基础上加以自适应散列 (Hash) 映射改进, 提出了自适应散列映射的弱跳完整性校验方法(AHMWHI).

1 弱跳完整性校验层协议描述与分析

1.1 协议描述

弱跳完整性校验过程 phw 有 4 个输入, 即密钥

收稿日期: 2004 - 02 - 26. 作者简介: 高 磊(1977 ~), 男, 博士生; 张德运(联系人), 男, 教授, 博士生导师. 基金项目: 国家信息化计算机网络与信息安全基金资助项目(2001 - 研 1 - 010).

sp、密钥 sq、 U_{MTU} 和 X , sp 和 sq 在 phw 处理中不会被修改,仅在密钥交换层 pe 中定时更新.同时,散列校验包信息经重组后放在数组 da 中,当前 q 至 p 的通道散列表为 htqp,其大小为 htspq,而当前 p 至 q 的通道散列表为 htpq,其大小为 htspq.布尔变量 qphc 为真(True)表示 htqp 已冲突,为假(False)表示 htqp 未冲突.报文的文本、摘要和大小分别用 t 、 d 和 s 表示.

在 phw 中还用到一些函数,即:TYPE(t)用于判断 t 是否为散列校验包,其值为 1 表示确定,为 0 表示否定;消息摘要计算函数 MD(t ,scr)用于计算报文文本 t 的摘要,其中的 scr 为密钥;散列表冲突检测函数 HC(t ,size,ht)用于检测将报文文本 t 映射到大小为 size 的散列表 ht 中是否会冲突;NXT(t)用于计算出报文文本 t 的下一跳路由器;NXTHS(t)用于从散列校验包的 t 中提取出下一轮散列表的大小;CHK(da,htqp)用于将 da 中摘要信息与 htqp 中每一个报文 t 进行校验比较,检验为不正确者报告,正确者转发,若产生发送散列冲突,结束 p 至 NXT(t) 一轮的散列校验,并发送散列校验包.

下面采用抽象协议符号^[2]对弱跳完整性校验层协议进行表述.

```
process phw
inp sp,  $U_{MTU}$ ,  $X$ : integer
  sq: array [0 ... 1] of integer
var  $t$ ,  $d$ ,  $s$ , htspq, htpq, newhtsq: integer
  qphc: boolean
  da, htqp, htpq: array [0 ... Max] of integer
begin
  rev packet( $t$ ;  $d$ ;  $s$ ) from qhw
    if TYPE( $t$ ) = 1
      {接收散列校验包} RCVHASH
     $\emptyset$  TYPE( $t$ ) = 0
    if qphc
      {报告丢失散列校验结束包}
    if da 非空
      CHK(da;htqp)
    fi;
    {清空 da 和 htqp,将  $t$  映射到 htpq 中}
    qphc = false;
    htspq = newhtsq;
    newhtsq = 1
  fi;
```

```
    if  $s > U_{MTU} - X$ 
      if MD( $t$ ;sq[0]) =  $d$  MD( $t$ ;sq[1]) =  $d$ 
        RTDATA
           $\emptyset$  MD( $t$ ;sq[0]) =  $d$  MD( $t$ ;sq[1]) =  $d$ 
          {报告数据包被修改} skip
        fi
       $\emptyset U_{MTU} - X < s < U_{MTU}$  {缓存  $t$  至 htqp}
      fi;
      if HC( $t$ ; htspq; htpq)
        qphc = true;
      fi
    fi
   $\emptyset$  true
  { $p$  收到  $q$  以外其他路由器的数据包 packet( $t$ ),
  并进行相应散列映射处理}
  RTDATA
   $\emptyset$  true
  {packet( $t$ ) 为  $p$  相邻主机发送或  $p$  自身发送}
RTDATA
end.
```

除了 MD 和 NXT 2 个函数外,过程 phw 还用了接收数据包 RTDATA 和接收散列校验包 RCVHASH 2 个声明.接收数据包声明的具体描述如下.

```
If NXT( $t$ ) =  $p$ 
  {接受  $t$ } skip
 $\emptyset$  NXT( $t$ ) =  $q$ 
   $d := MD(t;sp)$ ;
  send packet( $t$ ;  $d$ ;  $s$ ) to qhw;
  {将  $t$  信息映射到 htqp 之中}
  if HC( $t$ ; htspq; htpq)
    {确定新一轮散列表大小并存至 htspq,将 htspq
    与 htpq 中的校验信息发送给 qhw} skip
  fi
   $\emptyset$  NXT( $t$ ) =  $p$  NXT( $t$ ) =  $q$ 
  {计算出  $t$  的摘要  $d$ ,发送  $t$  至 NXT( $t$ ),并进行相应散列映射处理} skip
fi.
```

接收散列校验包声明的具体描述如下.

```
if MD( $t$ ;sq[0]) =  $d$  MD( $t$ ;sq[1]) =  $d$  {报告校验包被修改} skip
 $\emptyset$  MD( $t$ ;sq[0]) =  $d$  MD( $t$ ;sq[1]) =  $d$ 
  if newhtsq = 1 newhtsq = NXTHS( $t$ )
    newhtsq = NXTHS( $t$ );
```

```

{提取 t 中信息放入 da 中}
if t 为最后一个校验报文
    qphc = false;
    htscp = newhtscp;
    newhtscp = 1;
    CHK(da;htqp);
    {确定下一轮散列表大小并发送至下一跳}
fi
Ø newhtscp NXTHS(t)
{报告错误} skip
fi
fi.

```

1.2 协议验证

通过过程 qhw 到过程 phw 的通道状态转换图 (如图 1 所示),可验证该弱跳完整性校验层协议的正确性. 状态图中的每个变迁表示一个合法动作(由 phw 或 qhw 发起)或攻击下的一个违法动作,可用事件:消息标出. 事件有以下几种,即发送报文 S、接受报文 R、报文丢失 L、报文修改 M 和报文重放 P. 消息有以下几类型,即任意报文 pack2 et、散列校验包 hpkt、任意数据包 data、大数据包 bd 和其他数据包 ld. 因此,有{packet} = {hpkt} {da2 ta}, {data} = {bd} {ld}. 状态图中的 ch. qhw. phw 表示从过程 qhw 到过程 phw 的通道.

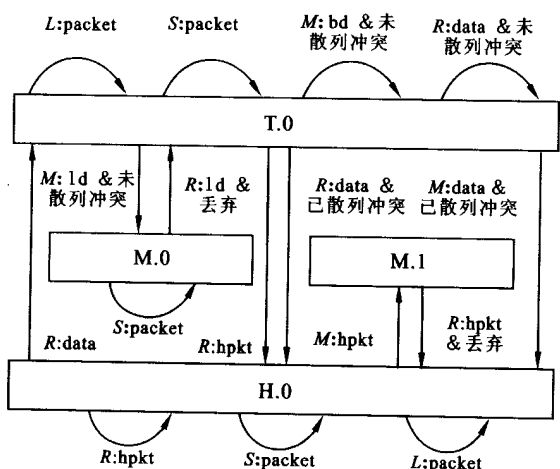


图 1 弱跳完整性校验层协议系统状态

状态图中有 2 个良好状态,即 T.0 和 H.0, T.0 为接收数据包的散列映射状态, H.0 为散列校验包重组校验状态,校验包接收完毕后将根据校验信息进行校验.

在 T.0 状态下,接收 hpkt 的动作和已发生散列冲突时接受 data 的动作将使协议进入到 H.0 状态,

其他合法动作将维持在 T.0 状态下.

在 H.0 状态下,仅当 data 到来后进入 T.0,并开始新一轮散列映射校验,其他合法动作将维持在 H.0 状态.

状态图的抽象化描述如图 2 所示.

$$\begin{aligned}
 T.0 &= I \wedge ((\forall ld(t; d) \text{ in ch.qhw.phw, 未散列冲突} \wedge \\
 &\quad d = MD(t; sq)) \vee (\forall bd(t; d) \text{ in ch.qhw.phw,} \\
 &\quad \text{未散列冲突})) \\
 M.0 &= I \wedge (\forall ld(t; d) \text{ in ch.qhw.phw, 未散列冲突} \wedge \\
 &\quad ((\neg \text{Head}(ld(t; d)) \Rightarrow d = MD(t; sq)) \vee \\
 &\quad (\text{Head}(ld(t; d)) \Rightarrow d \neq MD(t; sq)))) \\
 H.0 &= I \wedge (\forall hpkt(t; d) \text{ in ch.qhw.phw, } d = MD(t; sq)) \\
 M.1 &= I \wedge (\forall hpkt(t; d) \text{ in ch.qhw.phw, } ((\neg \text{Head}(hpkt(t; d)) \\
 &\quad \Rightarrow d = MD(t; sq)) \vee (\text{Head}(hpkt(t; d)) \\
 &\quad \Rightarrow d \neq MD(t; sq)))) \\
 \text{其中 } I &= sq \text{ in qhw} = sq[0] \text{ in phw} \vee \\
 &\quad sq \text{ in qhw} = sq[1] \text{ in phw}
 \end{aligned}$$

图 2 状态图的抽象化描述

在良好状态下的合法动作使协议保持着 T.0 H.0 T.0 的良好循环. 下面讨论协议在攻击动作的恶性影响下如何保持良好循环.

首先是丢失动作,丢失报文的类型可为 data 和 hpkt,其将使 T.0 和 H.0 都维持在原状态.

其次是修改动作,描述如下.

(1) 修改 data (bd 和 ld),即将在 H.0 状态下接收 data,并将其返回到 T.0 状态,且在 T.0 状态下对 data 进行处理. 在 T.0 状态下会将接收到的修改的 bd 进行缓存处理,维持在 T.0 状态,而对于修改的 ld,其状态会从 T.0 进入 M.0,然后返回到 T.0,丢弃 ld.

(2) 修改 hpkt,即将 T.0 状态下接收到的 hpkt 送入 H.0 处理,在 H.0 状态下接收到的修改的 hpkt 将进入 M.1,然后返回到 H.0,丢弃 hpkt.

虽然对于数据包的重放动作并不能立即发现,但当结束一轮散列校验后,根据校验包中的有态信息可判断出是否重放,并能丢弃缓存中重放的大数据包.

因此,无论是数据包的丢失、修改还是重放,都会维持在 2 个良好的状态下,使协议运行在良好的循环中.

2 协议性能分析

本文校验层先对大数据包进行缓存,其他数据包根据附带的摘要进行校验转发. 当一轮散列映射发生冲突时,会收到校验周期内的数据包的校验包,

此时根据校验包中的摘要信息对大数据包进行校验转发.因此,校验周期内的大数据包校验时延较大,而其他数据包的校验时延与文献[1]中的相同,这取决于摘要的计算时间.

校验层散列表的桶长为 L ,是一维散列表,当两项映射抵达同一桶时会产生映射冲突,此时触发发送校验包.对于到来的数据包 t ,散列映射按下式计算

$$\text{HASH}(t) = \begin{cases} (\text{ip}_{\text{src}} + \text{port}_{\text{src}} + \text{ip}_{\text{dst}} + \text{port}_{\text{dst}}) \text{ Mod } L \\ \text{TCP, UDP} \\ (\text{ip}_{\text{src}} + \text{ip}_{\text{dst}}) \text{ Mod } L \\ \text{ICMP, OSPF, RIP} \end{cases}$$

当一轮散列映射结束时,数据包个数的期望值为

$$E(L) = \sum_{i=1}^L \frac{i(i+1)L!}{L^{i+1}(L-i)!}$$

当前网络吞吐率为 S_{tp} ,校验周期 t 的期望值为 $E(t)$,由此得出

$$E(L) = E(t) S_{\text{tp}} = \sum_{i=1}^L \frac{i(i+1)L!}{L^{i+1}(L-i)!} \quad (1)$$

依据式(1),可在当前网络吞吐率 S_{tp} 下动态自适应地调整散列表的桶长 L ,以满足 t 规定的期望值.当然,在一轮映射过程中,对于自适应确定的当前桶长 L_c ,最坏情况下的大数据包时延为 $(L_c + 1) / S_{\text{tp}}$,其出现的概率为 $L! / L^L$,此时需要的存储空间为 $(U_{\text{MTU}} + X)L$.

在百兆级环境下对原型系统进行了测试,设定期望值 $E(t)$ 为 5 ms,吞吐率为 3414 kp/s (p 为单位“包”的符号)、11164 kp/s 和 146119 kp/s 分别代表正常流量、大数据包多和大数据包少 3 种情况,测试结果如表 1 所示.从表 1 可看出,散列映射周期

表 1 原型系统测试结果

吞吐率/kp s ⁻¹	34140	11164	146119
带宽/Mb s ⁻¹	84187	125160	109130
散列映射周期平均值/ms	0125	0184	0122
散列映射最大包数	69	37	108
每轮散列映射平均包数	9168	6198	21157

t (即大数据包的时延)的平均值远小于 $E(t)$,主要是因为 IP 流的本地性^[3]使散列映射产生冲突,从而结束一轮映射.

3 结 论

本文提出了自适应散列映射弱跳完整性校验模型,该模型对 t 时间内的大数据包进行缓存,对其他数据包根据附带的校验信息立即校验.当散列表映射产生冲突而结束一轮校验时,根据收到校验包中包含的校验信息对缓存中的大数据包进行校验,根据校验包中的有态信息对 t 时间内所有数据包的重放和丢失做进一步的检查,同时动态自适应地调整当前网络吞吐率下的散列表的桶长 L ,以满足给定的校验周期期望值 $E(t)$.同文献[1]弱跳下的模型相比,该模型增加了大数据包的校验、数据包重放校验的功能,并提高了对数据包丢失的检测能力.

该模型也有不足之处:虽然可有效发现并阻断大数据包的重放,但对其他数据包的重放仅能事后发现却不能阻断;对于散列校验包的重放没有一个有效的发现机制;易受到报文重放攻击或校验包丢失攻击,从而丢弃缓存中的大数据包.当然,这些不足仅会影响大数据包,其他数据包仍会携同其校验码在校验后转发,不会产生额外的延迟以及丢包和重放.本模型还应作进一步改进,使之成为强校验模型.

参考文献:

[1] Gouda M G, Huang C T, McGuire T M. Hop integrity in computer networks [J]. IEEE/ACM Transactions on Networking, 2002, 10(3): 308 - 319.
 [2] Gouda M G. Elements of network protocol design [M]. New York: Wiley, 1998.
 [3] Claffy K C. Internet traffic characterization [D]. San Diego, USA: Department of Computer Science and Engineering, University of California, 1994.

(编辑 苗 凌)