DDOS攻防与追踪

Refdom

12/21/2002

Xfocus 焦点峰会 2002

内容介绍

- DDOS攻击
- DDOS防御技术
- 数据包特征识别技术研究
- 源追踪（Traceback）技术
- DDOS监控技术

一、DDOS攻击现状与趋势

- 大分布型的高强度攻击
- 产生随机源IP地址
- 数据包结构位的随机性
- 协议缺陷与系统处理缺陷
- 使用多种协议及多种形式

加强DDOS的强度

- 混合的攻击

  例：Syn-cookie等防御在CPU消耗方面：

  - Syn包在Cookie表中的检查
  - 查询ACK标志包对应的Cookie表

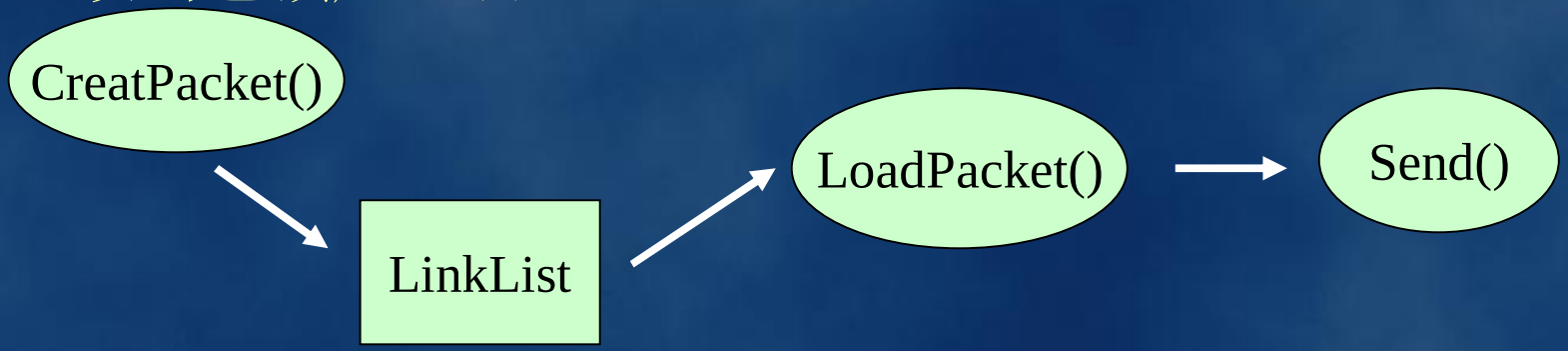
  SYN 、ACK flood;

  对Syn-cookie的探测

加强DDOS的强度

- 提高发包速率，减小checksum的计算量；

  攻击包预产生法



加强DDOS的强度

- 无指纹可识别;

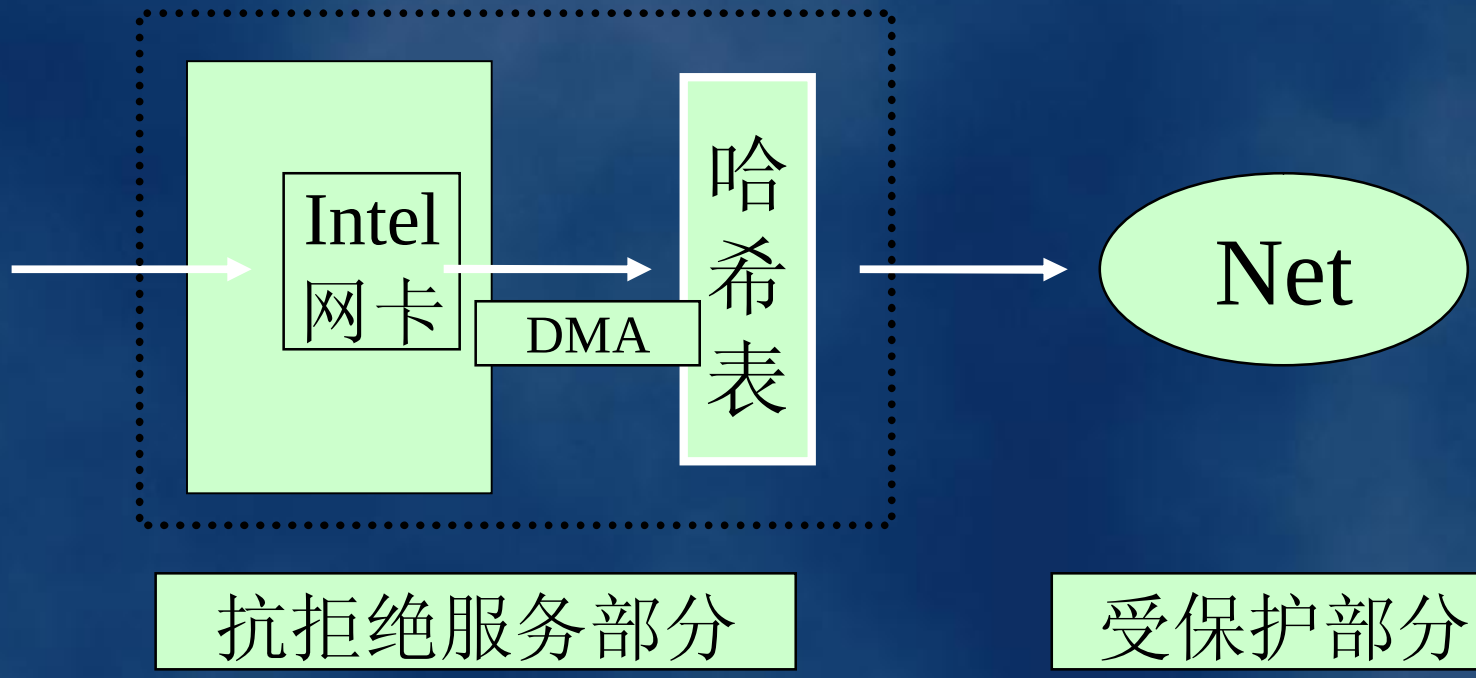
```
ipheader.id = SYSIdent.id + random;
ipheader.ttl = SYSIdent. TTL + random(10);
tcpheader.th_win = SYSIdent.window;
tcpheader.seq != random();
tcpheader.sport != rand(65535);
……
```

二、防御方法现状

- 数据包过滤（包括特征分析）
- 随机丢包
- SYN-Cookie, SYN-Cache等
- 被动消极忽略
- 主动发送RST
- 其他方法（动态DNS等）
- 拔网线 **J**

抗拒绝服务设计

- 简单结构（Syn-Cookie）



Syn Cookies

- Kernel/ drivers/ char/ random.c
- Cookie:

  MD5(sec1,saddr,sport,daddr,dport,sec1)

  + their sequence number

  + (count * 2^24)

  + MD5(sec2,saddr,sport,daddr,dport,count,sec2) % 2^24)

Where count increases every minute by 1.

IDS对SYN Flood的检测

- 现在的基于SYNs + TIME的特征；
- 真的没有问题么？
  - 正常SYN的高流量问题
  - 调整TIME问题
- RSTs + TIME是解决办法么？

路由器上的防御

- Access-list 访问控制列表

  access-list 101 deny ip 192.168.0.0 0.0.255.255 any

- Rate-limit 流量限制(bps)

  rate-limit output 512000 56000 64000 conform-action transmit exceed-action drop

- 请参考相关资料

路由器的Intercept模式

- 开启该功能：ip tcp intercept list access-list-number
- Watch和intercept（默认）模式。
- 设置模式命令：ip tcp intercept mode {intercept | watch}

路由器的Intercept模式

- Watch模式下，路由器允许SYN直接达到服务器。如果该会话在30秒（默认）内没有建立，就向服务器发送RST清除该连接。
- Intercept模式下，TCP连接请求达到目标主机之前，路由器拦截所有TCP请求，并代表服务器建立客户机的连接，并代表客户机建立与服务器的连接，当两个连接都成功实现，路由器就将两个连接进行透明的合并。

三、基础的数据包特征分析

- 从攻击代码中获得数据包的固定值，包括window、sourceport、seq或id等，比如mstream： win= htons(16384)； teardrop： id=htons(242)
- Land攻击的源IP与目的IP一样；
- Ping of Death分片ICMP包， 重组的包大于65535，通常为65538；
- ……

目前的特征分析缺陷

- 局限于对某种特定攻击或者工具；
- 太依赖于攻击程序中的固定值；
- 对于随机数则无法特征化；
- 无法普遍标识攻击流；

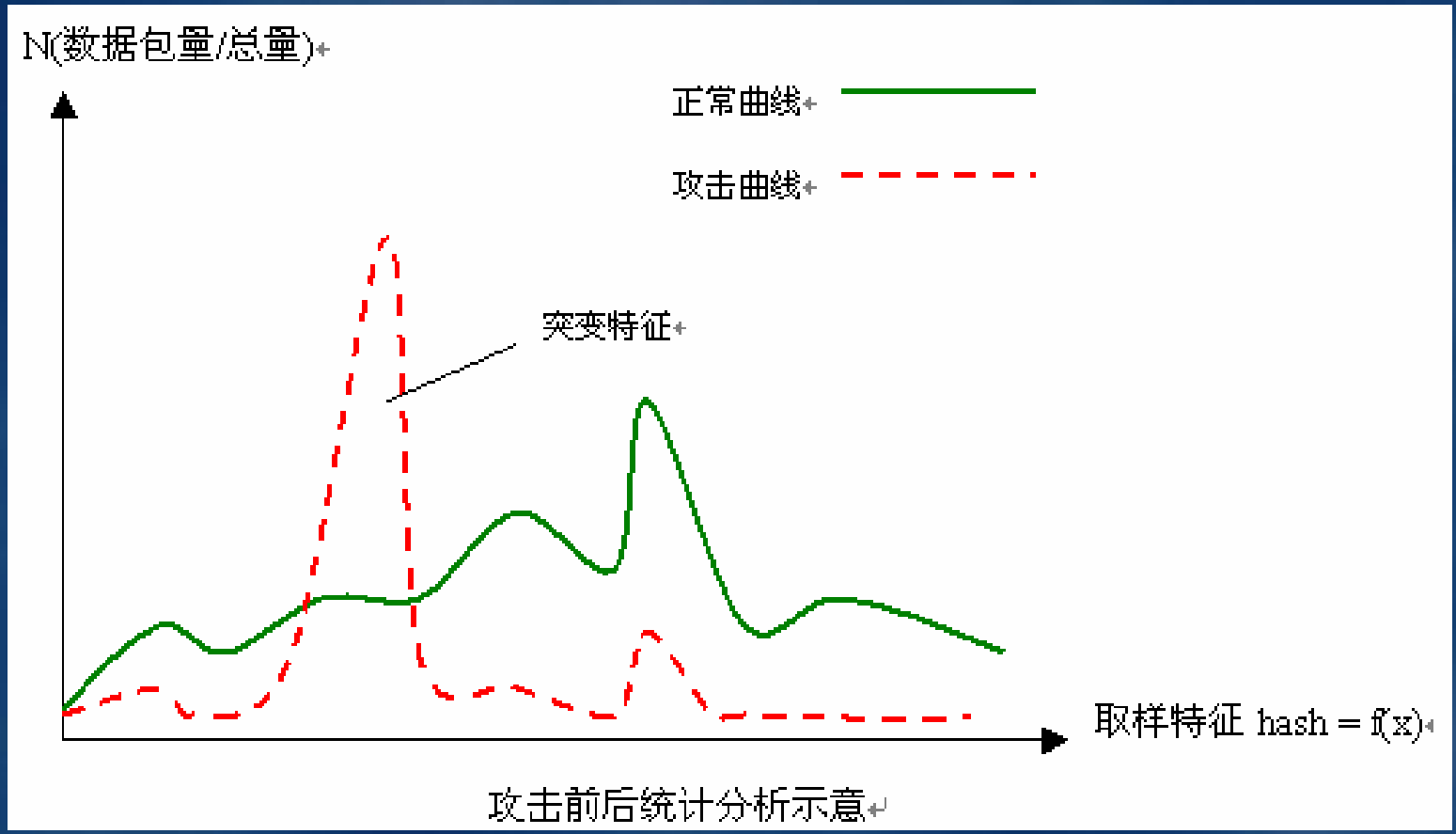
v 基于统计的特征分析

- **Statistic-Based Fingerprint Identification**
- Inside XFocus's DDOS Research Project
- 用某种Hash标志数据包。对正常的数据流量进行记录统计，得出正常的特征A，这些特征以数据包数量间关系的分布。在攻击发起的一段时间内，再次统计得到一个新的流量特征B
- 从分布曲线对比，获得B在A上的突变特征C；
- 通过将具有突变特征C的数据过滤，来减少攻击的损失，尽可能地保证最大化的正常访问。

基于的假设

- 攻击发起的数据包与统计没有关系
- 攻击发起的包程序化
- 攻击者发送足够多的数据包
- 攻击没有造成网络通路上的堵塞

突变特征的产生



攻击前后统计分析示意

对于TTL值的分析

- 系统默认的TTL值为255、128、64、32。
- 通常的路由Hop为10-20
- 正常的TTL范围：235~245、108 ~118、44 ~54、12 ~22

对TTL值的分析

- TFN3K的TTL算法：

  ih->ttl = getrandom (200, 255)

  TTL的范围为：

  （MAX）180~245；（MIN）190~235

- 通过TTL值即可过滤最大84.6%的攻击包

对TTL值的分析

- Mstream的TTL计算方法：

  packet.ip.ip_ttl = 255;
- 单一的TTL值255意味着TTL的统计分析能出现突变，对定位攻击源的位置也有一定帮助。

难度与缺陷

- 统计分布分析花销问题
- 只能尽可能地弱化攻击
- 将影响一部分正常数据包
- 不可特征化问题
- 过滤操作问题

仍进行中的研究……

- 用SEQ序号分析
- Hash函数研究

四、源追踪 Traceback

- Traceback简介
- Link Tesing
- Controlled Flooding
- ICMP Message
- Marking Packet Traceback

Traceback简介

- 目的
  - 杜绝攻击数据包的源头
  - 攻击取证与诱捕
- 难度
  - 随机伪造IP地址
  - 源头只是傀儡
  - 目前协议和硬件的局限
  - 路径重构的花销

Link Testing

- Hop-By-Hop
- 利用Cisco路由器
  - 实例介绍
- Cisco的IP Source Tracker

Controlled Flooding

- 实际上就是制造flood攻击，通过观察路由器的状态来判断攻击路径。首先应该有一张上游的路径图，当受到攻击的时候，可以从victim的上级路由器开始依照路径图对上游的路由器进行控制的flood，因为这些数据包同攻击者发起的数据包同时共享了路由器，因此增加了路由器丢包的可能性。通过这种沿路径图不断向上进行，就能够接近攻击发起的源头。

Itrace Working Group(IETF)

- 目前进度为：draft-ietf-itrace-02-ICMP Traceback Messages.txt
- 路由器以一定概率发送ICMP Traceback 消息。
- ICMP Traceback Messages重构攻击路径

ICMP Traceback的缺陷

- 在1/20000概率下增加0.1%的包
- ICMP包很可能被过滤掉
- 一些路由器没有input debugging功能
- 伪造ICMP Traceback问题
- 路由器的贫穷与富裕问题

Packet Marking Traceback

- Node Append
- Node Sampling
- Edge Sampling
- Compressed edge fragment sampling

附加节点算法 (Node Append)

- 把每一个节点地址附加在数据包的末尾，表明这是从哪里传入的。

- 缺点：
  - 对于高速路由器来说将增加负担
  - 可能导致不必要的分片，或者因为MTU而破坏
  - 协议中保留空间可能导致攻击者伪造内容

节点取样算法 (Node Sampling)

- 在路径中的一个节点取样，让一个样本来代替整个路径。
- 当接收到一个数据包，每个路由器就以概率p，选择性地将地址写到节点地址区间内。
- Victim通过一系列样本重构攻击路径

节点取样算法 (Node Sampling)

- 可以通过每个节点的相关数推理得到路径次序。
- 由于路由器被成序列地安排在一起，而且数据包被路由器标记的概率有一个概率基数，那么距离victim越近的路由器被标记的概率就会越小。
- 每个路由器的概率基数是p那么，经过d级路由标记的概率就是$p(1-p)^{d-1}$

节点取样算法 (Node Sampling)

- 算法:

Marking procedure at router R:

```
for each packet w
    let x be a random number from [0...1)
    if x<p then,
        write R into w.node
```

节点取样算法 (Node Sampling)

Path reconstruction procedure at vitim v:

```
let NodeTbl be a table of tuples(node,count)
for each packet w from attacker
    z:= lookup w.node in NodeTbl
    if z!= NULL then
       increment z.count
    else
       insert tuple(w.node,1) in NodeTbl
sort NodeTbl by count
extract pacth(Ri...Rj) from ordered node fileds in NodeTbl
```

节点取样算法 (Node Sampling)

- 缺陷
  - 改变IP头，并添加32 bits的节点空间
  - 需要重新计算checksum
  - 从有用的的取样中推理得出整个路由是一个相当慢的过程，如果d=15, p=0.51，那么接收方至少需要接收到42000个数据包才可得到一个标记取样。要确保正确率在95%以上，那么，还需要是这个数字的7倍。
  - 完全不适合分布式攻击

边缘取样(Edge Sampling)

- 在数据包中保留两个地址空间，用来标记路由开始点(start)和终止点(end)，并且用一个区间来表示距离(distance)，用它来表示一个样本到victim的距离。
- 当一个路由器决定标记数据包的时候，它把自己的地址填充到start，把distance域填0。如果distance域已经是0了，就表示这个数据包已经被前一个路由器标记过。在这种情况下，路由器就把自己的地址填入end域。如果路由器不标记数据包，那么就始终在distance域中加1。

边缘取样(Edge Sampling)

- 算法：

Marking procedure at router R:

For each packet w

```
    let x be a random number from [0,1]
    if x<p then
      write R into w.start and 0 into w.distance
    else
      if w.distance = 0 then
        write R into w.end
      increment w.distance
```

边缘取样(Edge Sampling)

- Path reconstruction procedure at vitim v:

Let G be a tree with root v

let edges in G be tuples(start,end,distance)

for each packet w from attacker

if w.distance=0 then

insert edge(w.start,v,0) into G

else

insert edge(w.start,w.end,w.distance) into G

remove any edge(x,y,d) with d!=distance from x to v in G

extract path(Ri,...Rj) by enumerating acyclie paths in G

边缘取样(Edge Sampling)

- 上面的算法表示边缘取样回溯IP过程。因为接收样本的可能性是与它同Victim的距离成几何递减的，对于一个有d级远的路由器来说，期望值就是 $1/[p(1-p)^{(d-1)}]$
- 要求从Victim能重新构建d深度路径的数据包数X，表达式是：

  $E(x) < \ln(d)/[p(1-p)^{(d-1)}]$ （Ln(d)为影响因数）

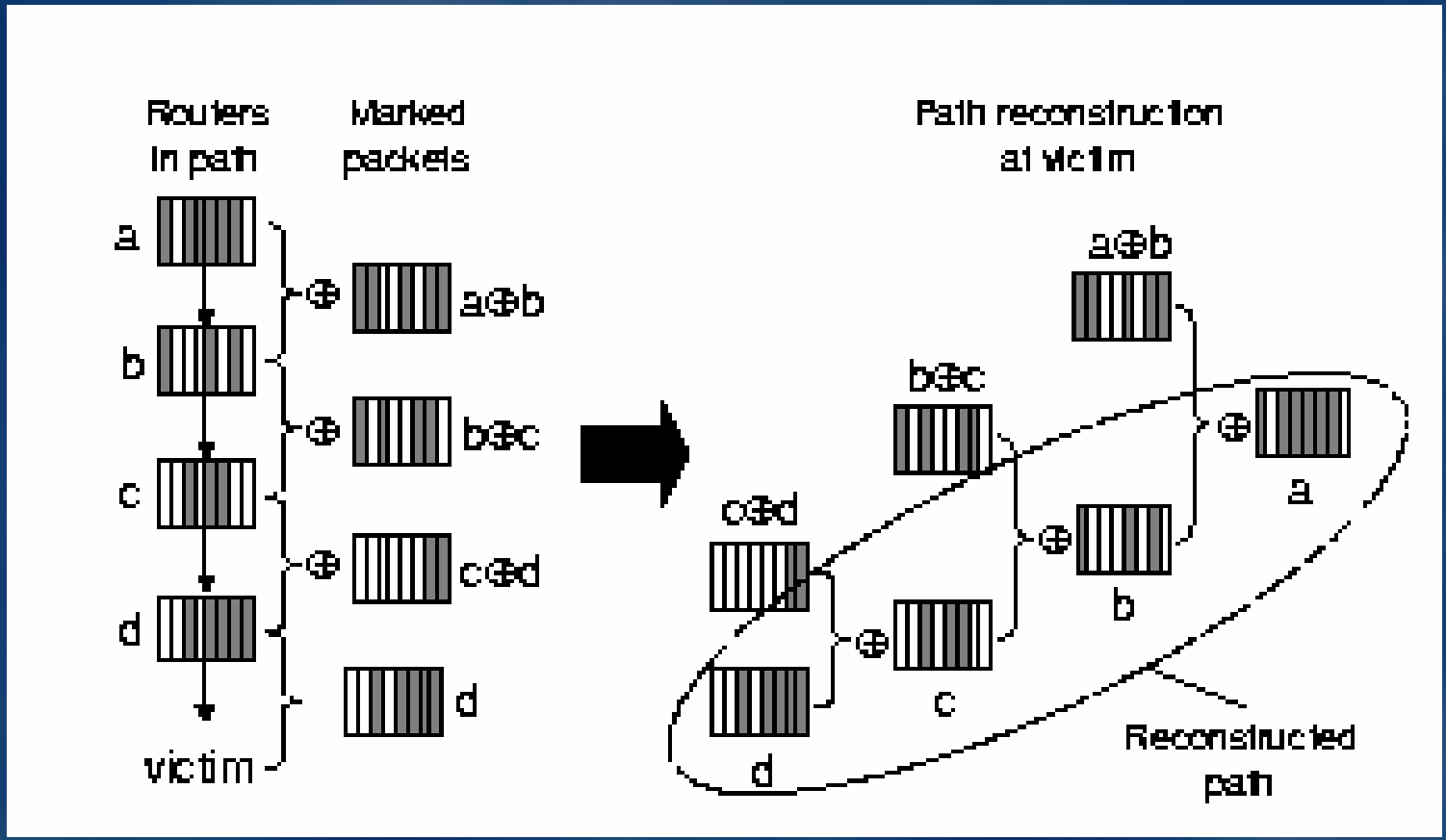
边缘取样(Edge Sampling)

- 缺陷
  - 在边缘取样算法中每个IP包需要多72bit空间；
  - 在包后面附加额外的数据花费昂贵，并且不一定有足够的空间来附加这些数据；

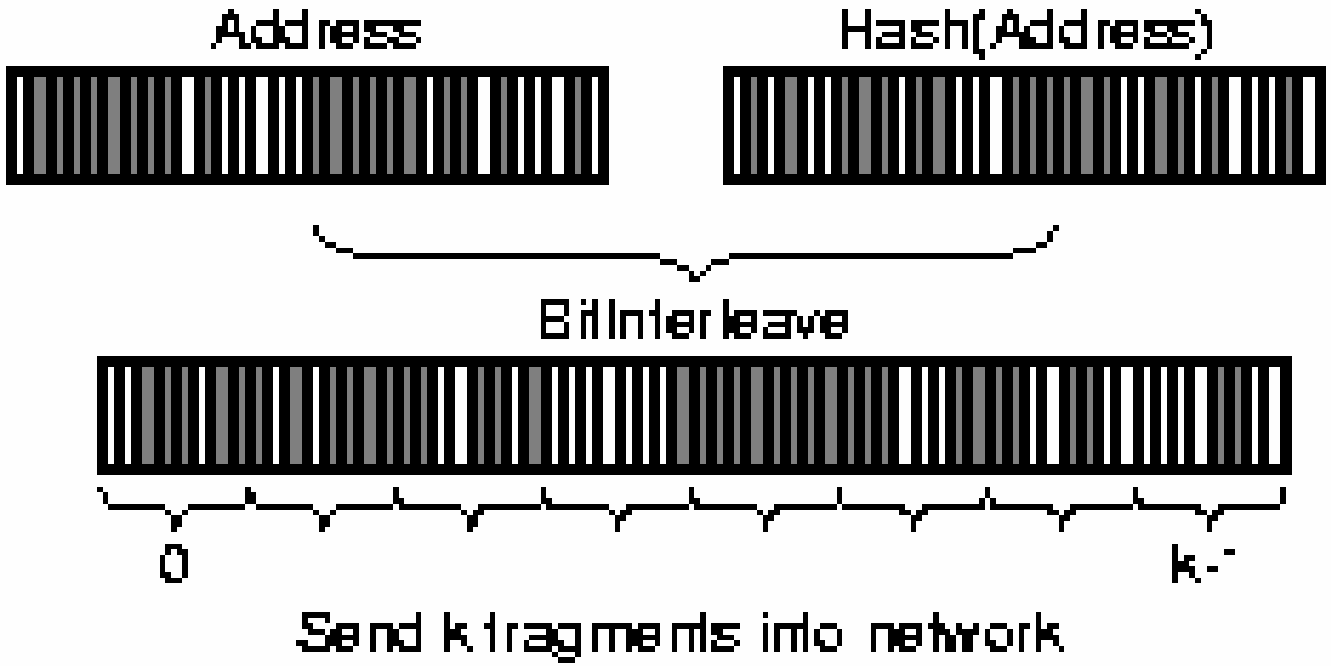
Compressed edge fragment sampling

- 建立在重载16位的IP identification域的编码方式；
- 通过标记边缘的两个IP地址进行XOR运算，可以只需要一半的空间；
- 将每个边界标记再进行细分成；

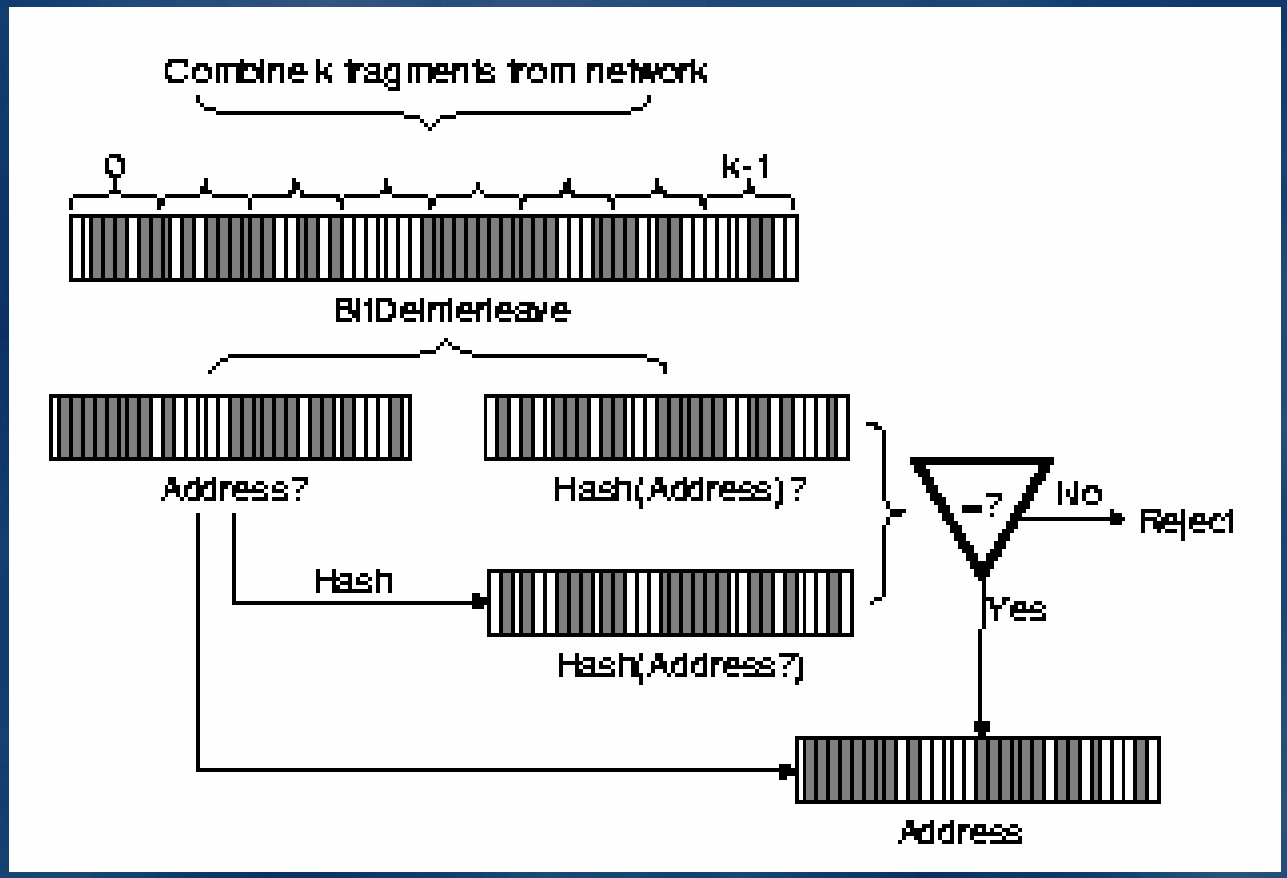
对IP地址进行XOR运算



将边界标记细分


Address
Hash(Address)
BitInterleave
0
k-1
Send k fragments into network

边界标记的重组判断


Combine k fragments from network
0
k-1
Bit Deinterleave
Address?
Hash(Address)?
Hash
Hash(Address?)
=?
No
Reject
Yes
Address

重载IP头的identification域

- 当前统计表明只有0.25%被分片；
- 重载IP头部的identification域；
  - 3bit的offset（可允许8次分片）
  - 5bit的distance可以允许31级路由
  - 8bit的边缘分片

缺陷

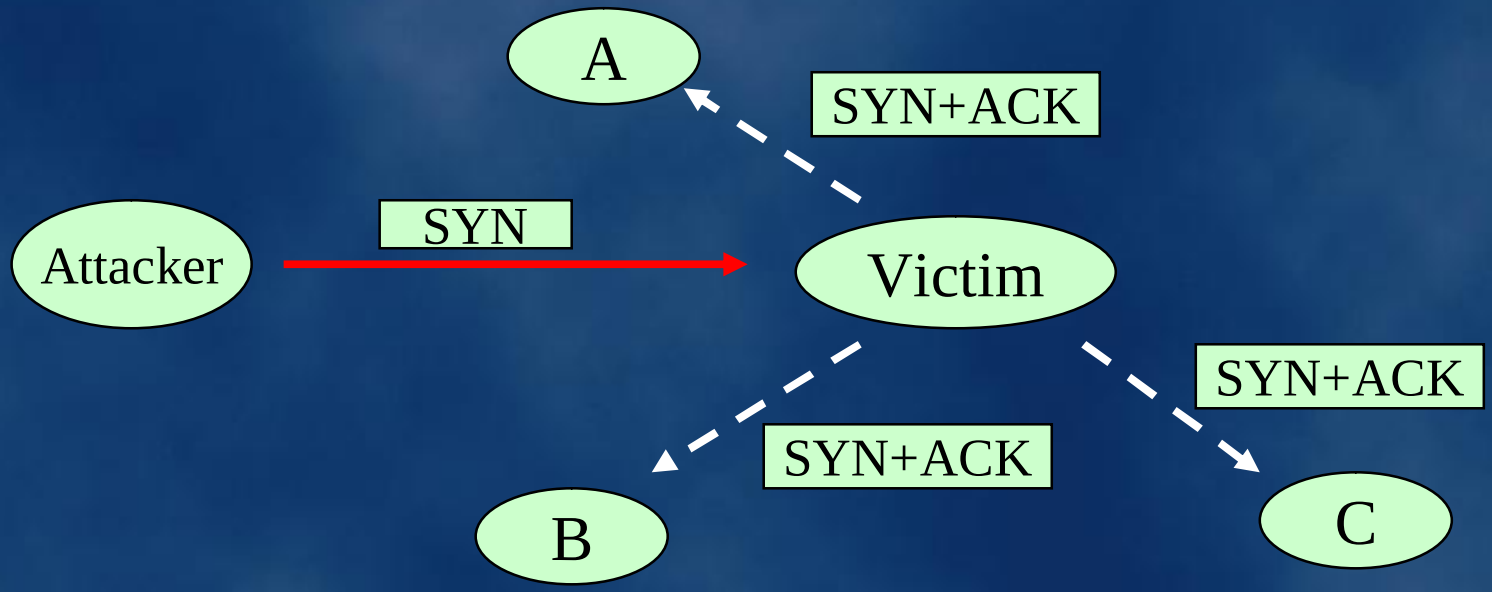
- 向后兼容性问题;
- 分布式攻击问题;
- 路径确认问题;

与Traceback交叉的技术简介

- Centerback
- Pushback

五、DDOS监控技术

- 反向散射分析（分布被动取样监控）



反向散射分析

- 假设源地址是完全随机产生的，也没有任何选择性。这些源地址就覆盖所有的IP地址范围，如果攻击者发送了m个攻击包，那么整个网络上的主机接收到Victim的回应包概率就是$m/2^{32}$
- 如果监视n个单独的IP，那么监视到攻击的期望值就是：$E(X) = nm/2^{32}$
- 在一个足够大的IP范围内做监控，那么就能够有效地对这些DOS攻击作采样分析。

DDOS监控未来的研究

- 监控与数据采样;
- 数据统计分析;
- 付之实施吧!

About: Xfocus's DDOS Project

- 对DDOS现状的分析；
- DDOS攻击的分析；
- DDOS防御与追踪研究；
- **Statistic-Based Fingerprint Identification**
- 反向散射分析（监控）；

Free Is All !

谢谢大家！

XFOCUS.ORG