

Overview. IP traceback support.

1 Some security weaknesses in the Internet architecture

- Hacking into machines. Well, maybe not a weakness of the network architecture per se, but rather in end-host implementations. Or in people picking lousy passwords amenable to dictionary attacks. But many attacks take advantage of buffer overruns, e.g., in weakly-tested code paths like reassembling overlapping fragments, ICMP handling, etc. Or, for example, the Internet worm.
- Connection hijacking. Saw this in the context of the connection migration paper.
- Denial-of-service. SYN floods. Ping floods. Distributed attacks. This paper addresses the question of how to trace an attack back to near the source.
- Compromising congestion control. E.g., ACK every byte! Or ACK unreceived segments and rely on application-level retransmissions (e.g., HTTP range requests) to get remaining stuff.

Fundamental trade-off in architecture between anonymity (consequence of statelessness) and accountability (for which state is required). Most applications don't seem to care about end-to-end security, usually punting on it.

2 Denial of service

- Goal: To make a service unusable, usually by overloading a server, router, or network link.
- SYN floods. Started a few years ago. Solution: SYN cookies.
- Bandwidth attacks. More recent trends, e.g., trinoo, tfn. General attack architecture as shown in Figure 1.
- Consider what happened to Yahoo! Hacked into machines at colocation centers and sent ping broadcasts to machines with a source address of Yahoo! Ping responses went to Yahoo!, since that was the observed ping requestor.
- Figure 2 summarizes this attack.

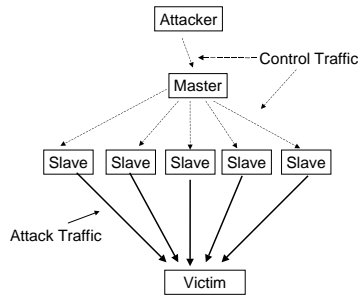


Figure 1: Architecture of bandwidth attacks.

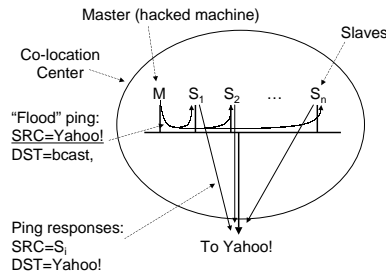


Figure 2: Mechanics of a DoS attack.

2.1 Possible solutions

- Ingress filtering. RFC 2827. To handle asymmetric routes, need separate ACL.
- Link testing. Either try and discern attack signature or do controlled flooding to see how attacker traffic is perturbed.
- Logging. Consumes lots of resources, but possibly viable.
- ICMP traceback. Sample and send back ICMP towards victim.
- IP traceback support. (This is the paper in the readings.)

2.2 IP traceback support

- Record route. Doesn't work too well because of space required and poor IP options support.
- Node sampling. Each router en route marks it's IP address on the packet with some probability, p . $P(\text{receiving mark from router distance } d) = p(1-p)^d$. Sharply falls with d . But

problem is that routers far away take a while to show up, especially since routers closer to victim have a chance to overwrite marks.

- Q: why do we allow routers to overwrite?!
- Edge sampling. Solution to this problem. Encode distance from victim as well. “Start router” marks outgoing interface IP in the “start field” and sets the “distance” field to 0, with some probability. If the distance field is 0, then router assumes it is the end point and marks the “end” field. If the router decides to not mark the packet, it *must* increment the “distance” field.
- This scheme is incrementally deployable, but must be done so in pairs of routers.
- As before, each router makes autonomous probabilistic decisions about marking an edge.
- For a router at distance d away from victim, expected number of packets before we get one mark from it is $\frac{1}{p(1-p)^{d-1}}$.
- What is the number of packets before we get at least one mark from every participating edge, assuming distance from victim to attacker goes through D participating router edges?
- $P(\text{receiving mark from router at distance } d) \leq P(\text{receiving mark from router at distance } D) \geq p(1-p)^{d-1}$. Now use the result of the *coupon collector's problem*, which says that if you have K distinct tickets and you keep drawing one after another with replacement, the expected number of tries before you get them all is $O(K \ln K)$.
- In our situation, if we receive N packets from an attacker, at least $Nd(p(1-p)^{d-1})$ packets will be marked. From the coupon collector's result, $Nd(p(1-p)^{d-1}) < d \ln d$, or $N < \frac{\ln d}{p(1-p)^{d-1}}$.
- Encoding. We don't have enough space in the IP header for this! So, take XOR's of start and end router addresses, and only ship part of the 32 bits! Their suggestion is to ship 8 bits at a time and use 5 bits to represent distance, and 3 bits to tell the victim (and intermediate routers) which offset into the 32 bits is being encoded.
- Suggested implementation uses IP fragment ID field.

2.3 Weaknesses

- Tunnels. Other forms of stepping stones.
- Multiple attackers may be very time consuming to detect, with the encoding approach.