

Security Topics in Networking and Distributed Systems

CS 590D

Lecture 6

Department of Computer Sciences
Purdue University

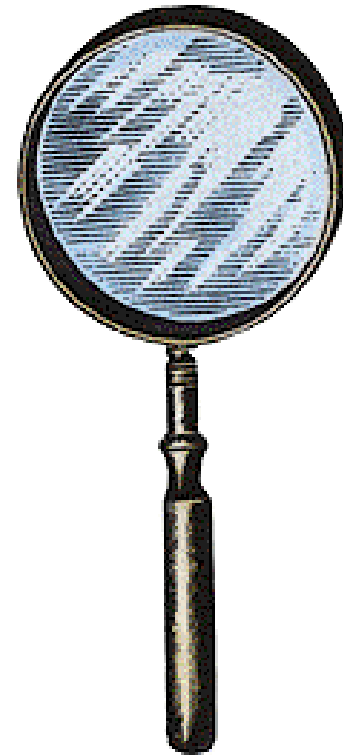
Outline

- Hash-Based IP Traceback
A. C. Snoeren, C. Partridge, L. A. Sanchez, C. E. Jones, F. Tchakountio, S. T. Kent, and W. T. Strayer.
SIGCOMM 2001
- On the Effectiveness of Route-Based Packet Filtering for Distributed DoS Attack Prevention in Power-Law Internets. Kihong Park, Heejo Lee.
SIGCOMM 2001
- Implementing Pushback: Router-Based Defense Against DDoS Attacks.
John Ioannidis, Steven M. Bellovin.
NDSS 2002.



More on IP Traceback...

- Probabilistically mark packets: false positives, multiple attackers, requires many packets to track attack
- What about attacks that were caused by one packet?
- How to trace one packet?
Logging



Revisiting Logging

- **Technique**: log packets at key routers and use data mining to determine the path the packets traveled; **This seems good!**
- **Requires**: high resources and integration of large scale inter-provider database. **Hmmm... not that good!**



The Problems with Logging

- High-resource consumption: assume store entire packet, for high speeds links(OC-192 - 1.25GB/sec), 16 links, a query each 60 seconds, requires 1.2 TB storage
- Packet is transformed while traveling, which makes difficult the attack path reconstruction
- If logs get compromised, privacy risk

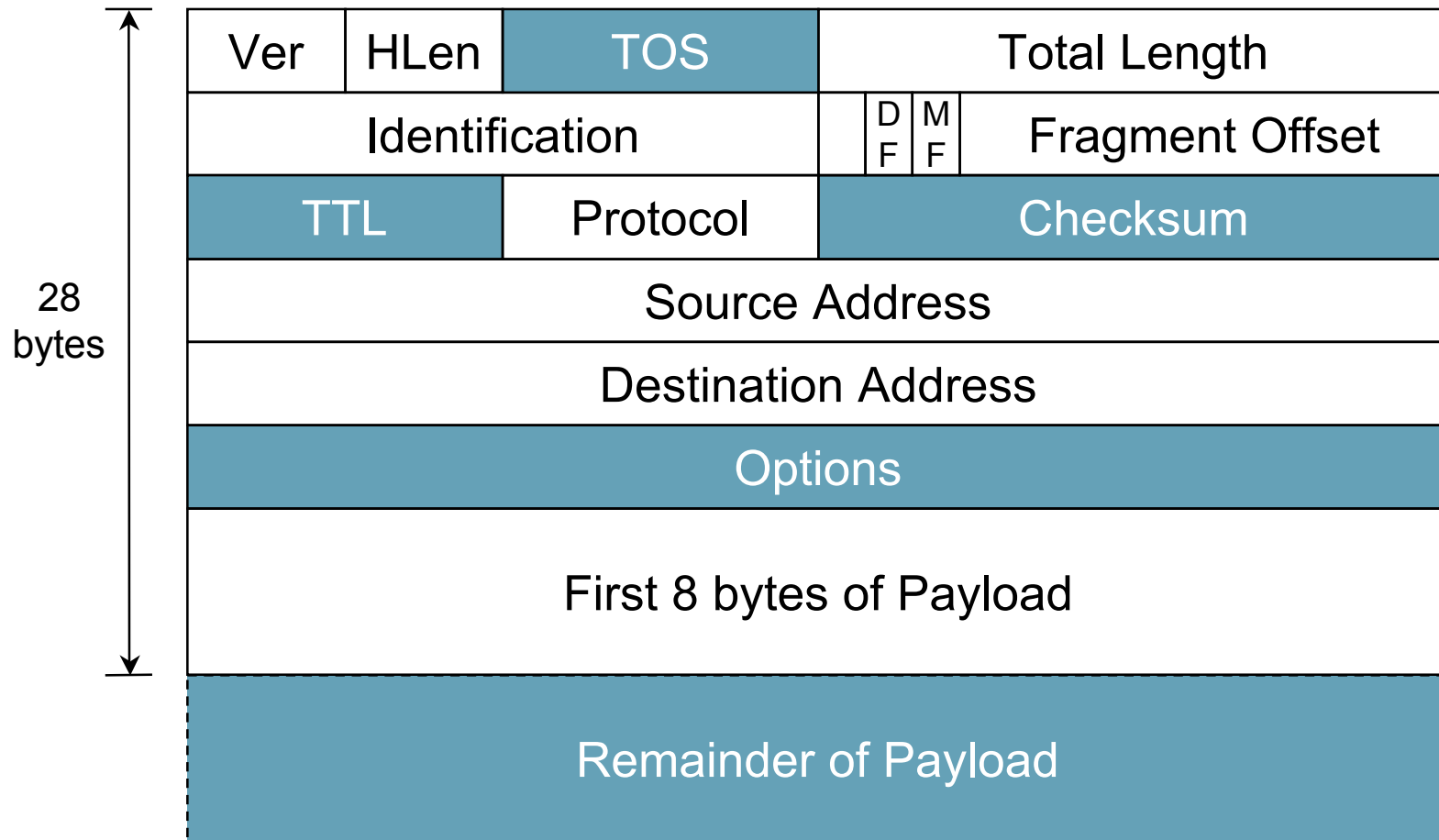
What is Needed

- **Efficient way to store packets**
- Data aggregation
- Build attack graphs fast when authenticated requests come
- Low false positives, no false negatives.

Saving memory ...

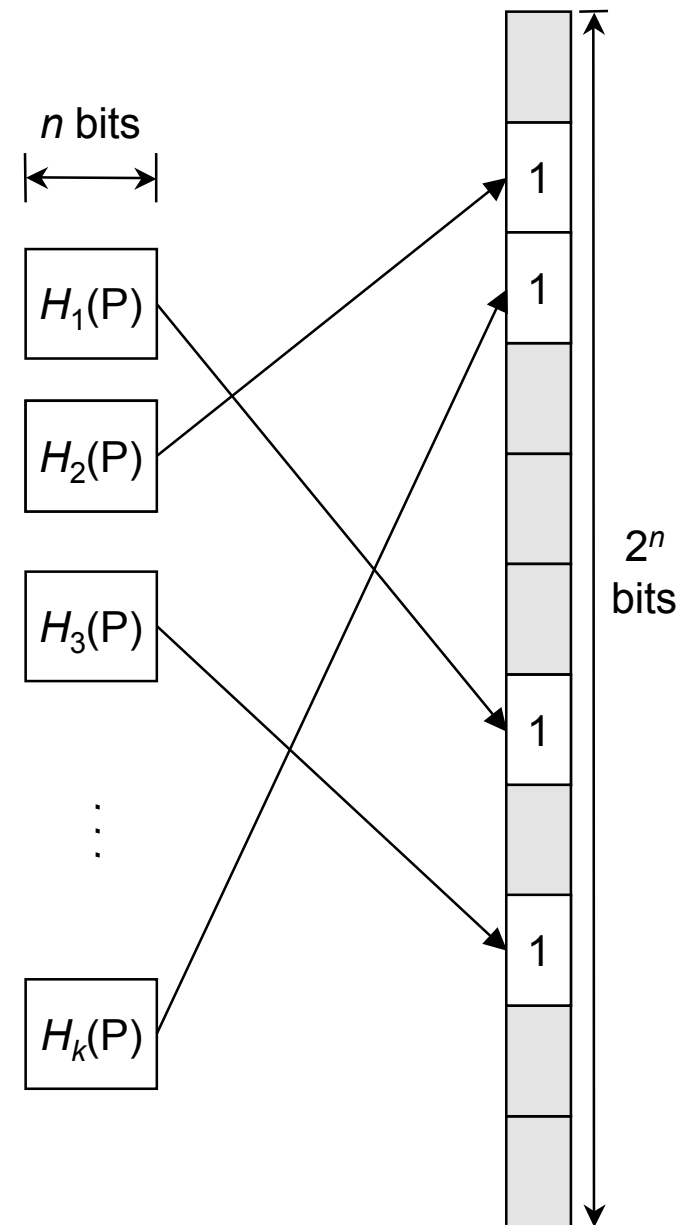
- Main idea: store hashes of messages
- Compress more: use Bloom filters
- If 28 bytes hash input, 0.00092% WAN collision rate
- Hash applied only on selective fields.

Hash Input



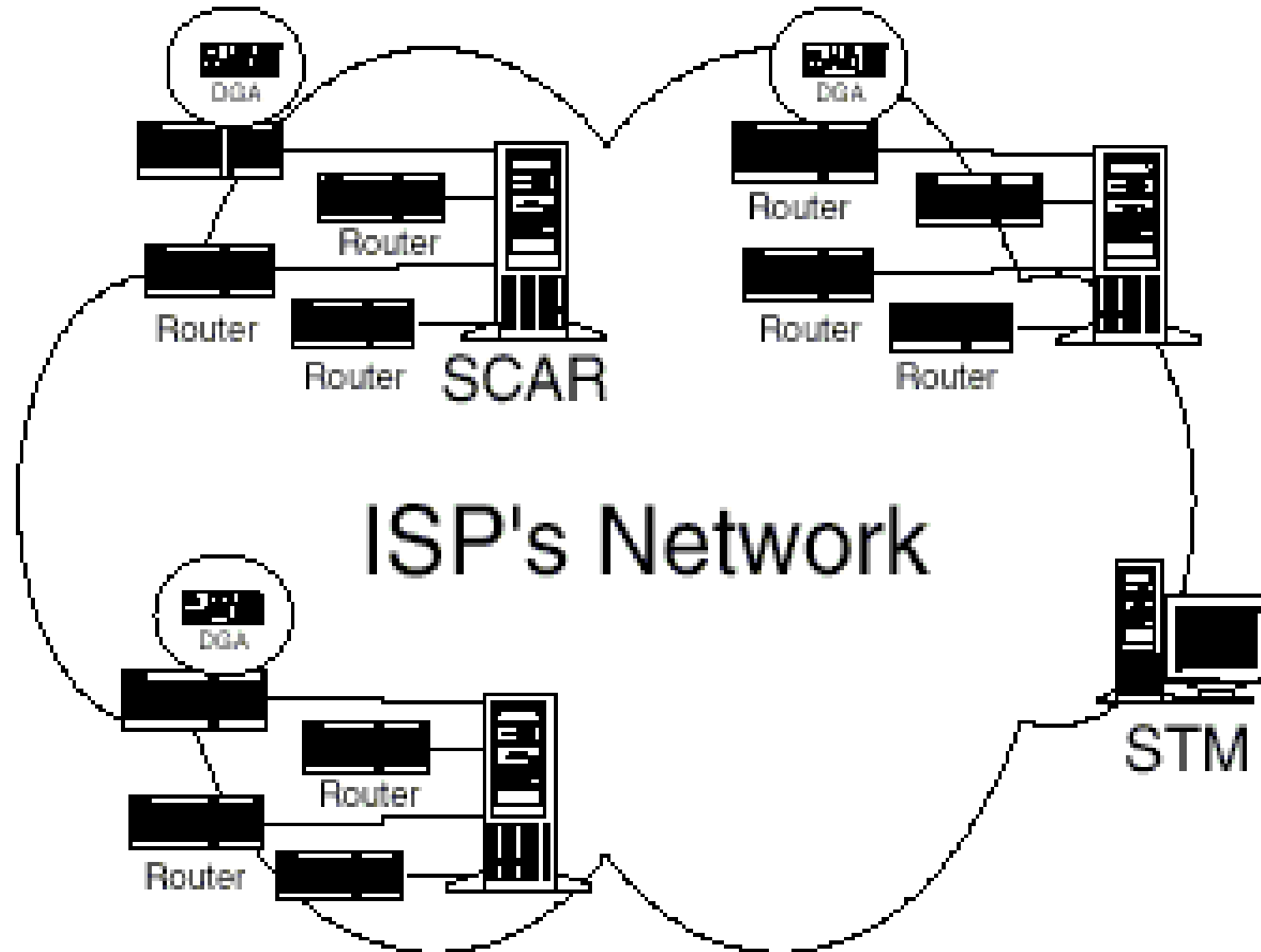
Bloom Filters

- Uses k distinct hash functions (n-bit output) packet
- Computes k digests for each packet
- Build a 2^n array, bits set to 1 for each of the k hash results
- Membership test: compute the k digests on the packet and check the indicated bit position
 - If any bit = 0, packet not stored in table
 - If all bits = 1, packet most probably stored in table



Source Path Isolation Engine

- **DGA: Data Generation Agent**
 - computes and stores digests of each packet on forwarding path.
 - 1 DGA per router
- **SCAR: SPIE Collection and Reduction agent**
 - long term storage for packet digests
 - assembles attack graph for local topology
- **STM: SPIE Traceback Manager**
 - interfaces with IDS
 - verifies integrity and authenticity of traceback call
 - sends requests to SCAR for local graphs
 - assembles attack graph from SCAR input



Proactive Countermeasures

- GOAL: LIMIT THE ATTACK
- PLAYERS: routers
- **Pushback**: limit attack traffic at a router that is close to the attack source treating DDoS as a congestion control problem
- **Route-based filtering (Distributed packet filtering)**: generalization of ingress packet filtering, extended to to core

Pushback



- Main idea: **congestion control of the attacker's traffic, without interfering with the normal traffic**
- Congestion control is already implemented in routers (**active queue management**), but **this control can not differentiate traffic**, so normal traffic is also dropped.
- Proposed solution: **Aggregate-based Congestion Control (ACC)** where the traffic is aggregated based on a “signature” (common property).

Aggregate-Based Congestion Control

- Identify and control locally aggregate flows.
- Pushback: request upstream routers to also limit the flow of aggregates.
- Rates are not fixed and are updated periodically.

Dropped Packet Report

- Magic number
- IP destination address
- Input interface
- Output interface
- Timestamp
- Packet size
- Reason: tail-queue drop, RED drop

Identifying Aggregates

- List high-bandwidth 32-bit addresses.
- Cluster into 24-bit prefixes.
- Try longer prefixes with most of the drops.
- Try shorter prefixes by merging aggregates.
- Repeat if $w_i - w_b > 1.2 w_0$

Rate Limiting

- Sort list of aggregates by drop rate.
- Estimate arrival rates over a time interval seconds.
- Calculate the excess rate, traffic that must be dropped to reach a certain rate
- Find minimum number i of aggregates that must be rate-limited to achieve this drop.

Pushback Report

- Header
- RLS-ID
- Maximum Depth
- Depth of Requesting Node
- Bandwidth limit
- Expiration time
- Congestion Signature

BGP

- The internet is divided into disjoint sets called autonomous systems (**AS**)
- BGP is the routing protocol between AS similar with distance vectors protocols
- Routing updates contain an ordered list or AS path of traversed autonomous systems and a set of network prefixes belonging to the first AS in the list.
- BGP routing messages are the highest precedence traffic on the Internet,
- Routes are not optimal because peering

Link-State Routing

- Each node:
 - Maintains **global view** of the network.
 - Sends periodically the current state of all links (link-state updates or advertisements) to all nodes (via flooding).
 - Notes the change and recompute its routes (use **shortest-path** – Dijkstra algorithm) to destination.
- Less bandwidth-intensive than Distance-Vector, but more complex and more compute- and memory-intensive.
- Examples: OSPF uses link-state routing.

Distance-Vector Routing

- Each node:
 - Maintains a vector with distances to all of the nodes.
 - Sends periodically its distance-vector to all its neighbors.
 - Updates its distance vector based on the information received from the neighbors (shortest path Bellman-Ford): for each network path, the receiving routers pick the neighbor advertising the lowest cost, then add this entry into its routing table for re-advertisement.
- Examples: RIP uses distance-vector routing.

Distributed Packet Filtering

- Main idea: move filtering from the ends to the backbone
- Goal:
 - achieve the same protection as if ingress filtering was applied by everybody
 - Maximum coverage with minimum number of filters
- Results show that packet filters in 18% of ASs in Internet can significantly reduce spoofed packets

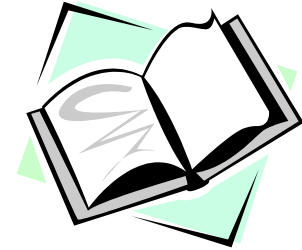
Distributed Packet Filtering: How?

- Checks if packet arriving at border router at an AS is valid with respect to its source/destination given the reachability constraints imposed by routing and topology
- If does not verify, packets is dropped
- Collective filtering on autonomous systems
- Exploit the power-law topology of the Internet
- Allows to point to a number of sites as potential source (support for reactive measures).
- Requires finding the minimal vertex cover

Deployment Issues

- What happens if the routing table (routing updates) are the target of the denial of service?
- The assumption is that the filtering mechanism is not subject of DDOS itself
- Assumption that DPF is safe, never discards “good” packets
- Requires inferring the reachability of the source, currently BGP does not provide this.

Next Week: Intrusion Detection



- Tuesday
 - An Intrusion-detection Model. De. E. Denning. IEEE Trans. on Software Engg., SE-13, pp.222-232, February 1987.
 - Detecting Intruders in Computer Systems T. F. Lunt Conference on Auditing and Computer Technology, 1993.
- Thursday (NO HOMEWORK)
 - Temporal sequence learning and data reduction for anomaly detection. T. Lane and C. E. Brodley ACM Transactions on Computer Security, vol 2, num 3, pp 295-331, August 1999.