CSE599N1: Homework 1

Julie Newcomb, Maaz Ahmad

October 21, 2019

1 Header Space Analysis

We implemented the dataplane verification technique Header Space Analysis, as described in Kazemian et al [1]. Header Space Analysis (HSA) is a protocol-agnostic static checking technique which can verify network properties such as reachability, isolation, and being loop-free. We implement a portion of HSA in Python to check invariants on a test network.

1.1 The HSA model

HSA uses a geometric model in which a packet header, represented in binary, is a point in the header space $\mathcal{H} = \{0,1\}^L$, where L is the upper bound on the length of the header. To represent a region in $\mathcal{H}$, we introduce the wildcard variable x, which can be used to show the value of a particular bit is not specified. Wildcard expressions represent sets; for example, $10x = \{100, 101\}$. The network space $\mathcal{N}$ is the cross-product of $\mathcal{H}$ with the set of all ports in the network. Packet traversal in the network is represented by the network transfer function Ψ, in which each node has a transfer function mapping a header and port to a set of emitted headers and ports:

$$T(h,p) : (h,p) \rightarrow \{(h_1,p_1),(h_2,p_2)...\}$$

and Ψ is defined:

$$\Psi(h,p) = \begin{cases} T_1(h,p) & \text{if } p \in switch_1 \\ ... & ... \\ T_n(h,p) & \text{if } p \in switch_1 \end{cases}$$

Finally, the movement of packets through the network topology is modeled by the topology transfer function Γ:

$$\Gamma(h,p) = \begin{cases} \{(h,p')\} & \text{if } p \text{ connected to } p' \\ \{\} & \text{if } p \text{ is not connected.} \end{cases}$$

The behavior of packets as they move through the network can be modeled by alternating applications of Γ and Ψ.

1.2 Our implementation

We assume headers always have the same five fields: a source IP address, destination IP address, source port, destination port, and protocol.

In our implementation, the transfer function for each node p checks that the header is allowed by its inbound ACLs, looks the header up in its forwarding table to find the node p' to pass it on to, checks that the p' outbound ACLs allow the header, and emits (h, p') (if either ACLs do not allow the header, or the header does not match any forwarding rule, the packet is dropped and the function outputs an empty set). If the packet header is fully specified, it will never be altered by the transfer function as we do not model NATs or other dynamic network functionality.

However, in the HSA model, when a header contains wildcards, it represents a set of headers. When we apply the transfer function Ψ to a wildcard header h and some port p, we get back pairs of all possible ports that any of the headers in h could reach and the respective sets of headers that could reach those ports. For example, let node p have a forwarding table with only one entry. We can represent that entry with a wildcard expression representing the set of all headers that could match it. The transfer function for that node would output (h', p'), where h' is the set intersection of the input header set h and the wildcard expression of the forwarding table entry for p' (of course, if h' is the empty set, the transfer function will output the empty set). Now imagine a forwarding table with two entries. The entry with the longer prefix length will behave as above, but for any header to match the second entry, it must not have matched the first. Thus, for that entry we emit (h'', p'') where h'' is the intersection of the input header, the entry wildcard expression, and the set complement of the first entry's wildcard expression. ACLs are implemented similarly. (Note: due to lack of time, we only implemented wildcard matching for source and destination IP address fields.)

To check reachability of port b from port a for a given packet set, we choose a network diameter d as a bound on the number of network hops to check. We apply the transfer function Ψ and the topology transfer function Γ to get a set of all header/port pairs that can be reached with a single hop from a. We next call the transfer and topology function on each pair in that set, continuing up to d times. If at any point we produce some pair (h', b) (where $h' \neq \varnothing$), we know b is reachability from a.

References

[1] Peyman Kazemian, George Varghese, and Nick McKeown. Header space analysis: Static checking for networks. In *Presented as part of the 9th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 12)*, pages 113–126, 2012.