ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Prof. L. Vanbever

Master thesis proposal:

SDN-based Network Obfuscation

Today's networks are quite sensitive to eavesdropper, *i.e.* malicious hosts that can observe the traffic. Indeed, even if the traffic being exchanged is encrypted, an attacker can still infer a lot of critical of information about the network such as: the MAC/IP addresses of the routers, the IP of the DNS server, the presence (or absence) of middleboxes, the hosts and who they are talking to, etc. The attacker can then use these information to perform a targeted attack, for instance, to overload the gateway router.

The goal of this Master Thesis is to use SDN to fool the eavesdropper into believing that the network is something else that what it physically really is. Several aspects of the network can be "faked". For instance, the number of hosts seen by the eavesdropper can be increased or decreased by rewriting the IP header at each hop, bearing similarities with applying TOR or onion routing inside the network itself. Likewise, TCP ports can be rewritten to hide the services that are running behind a given IP. The IP topology can be faked as well, by faking IP subnets and the MAC address of the routers. Doing so, the network can pretend that they are more or less routers, and (possibly) that they are located at different locations than the real physical locations. TCP flags could also be rewritten using dedicated MB to prevent the attacker to monitor ongoing sessions. Another way to limit eavesdropping is to use multi-paths so as to limit the probability that one attacker sees all the traffic.

Obviously, the traffic still needs to eventually reach the destination, with the appropriate header so as to guarantee bi-directional communication. A way to remap obfuscated traffic to the original headers is therefore needed.

Bootstrapping questions:

- How should we evaluate the obfuscation costs? Cost could be expressed in terms of network resources (*e.g.*, link bandwidth, extra delays due to routing along non shortest-path, forwarding table size) *vs* security benefits obtained? What kind metric can be used to do that?
- How much protection can actually be provided? Characterize the amount of information an attacker can deduce using a given forging model and a given set of tapping locations? Attacker can wiretap at different points. His power is given by how many taps he has. How can we adapt the forging given the assumed power of the attacker?
- Implement the system using a SDN controller such as POX. An alternative would be to use a Hyvervisor such as Flowvisor that could be used to obfuscate existing controller output.

Related work:

[1] OpenFlow Random Host Mutation: Transparent Moving Target Defense using Software Defined Networking, http://conferences.sigcomm.org/sigcomm/2012/paper/hotsdn/p127.pdf
[2] FlowVisor, https://github.com/OPENNETWORKINGLAB/flowvisor/wiki
[3] The TOR project, https://www.torproject.org/