Advanced Topics in IP Networks
Lecture 09

Lecture by Dr. Anat Bremler-Barr
Typeset by Steven Karas

2018-12-13
Last edited 20:56:44 2018-12-13

Disclaimer These notes are based on the lectures for the course Advanced Topics in IP Networks, taught by Dr. Anat Bremler-Barr at IDC Herzliyah in the fall semester of 2018/2019. Sections may be based on the lecture slides prepared by Dr. Anat Bremler-Barr.

1 Homework

Lior will arrive next week and present the next homework. The current homework is due next Sunday?[1]

2 Agenda

- Opportunities in Middleboxes

3 Network Function Virtualization

Middleboxes provide various network services, such as firewalls, proxies, intrusion detection, network traffic loggers, etc. Most of these middleboxes are mandated by regulatory requirements. It has gotten to the point where sometimes there as many middleboxes as switches and routers in a network. These can take up many rack Us, power consumption, etc, so there is a huge incentive to virtualize the functionality.

Conventional wisdom says to place a middlebox in each path, which means that middleboxes must be duplicated throughout the network, scaling with the number of paths. SDN helps with this by being able to force traffic through a centralized middlebox.

Some vendors saw this opportunity for consolidation, and built multi-function middleboxes. This mixes the advantages of NFV with the added commercial benefit of vendor lock in. Another approach was to move middlebox handling to the cloud, which can only be provided by larger providers with sufficient PoPs. A classical example of this is DDoS protection.

At some point, this push towards virtualization, the kernel became the bottleneck for handling packets. The current state of the art is to do a kernel bypass using things like PSIO, DPDK, and friends.

3.1 Software Defined Networking

Removes the restrictions on where middleboxes need to be installed. Enables placing various middleboxes (or virtualized middleboxes) at different switches, and bouncing packets in and around them before delivering them to the actual destination.

This in combination with NVF provides us with all the advantages and only two disadvantages: overhead of routing packets (which can be reduced by providing better network fabric between middleboxes), and virtualization overhead of middleboxes. There is active work to reduce the virtualization overhead, which has already achieved significant reductions in this overhead.

[1] The website stated it was due on the 21st, but she said verbally Sunday

3.2 DPI based Middleboxes

Many middleboxes perform Deep Packet Inspection, which ultimately leads to a duplication of efforts. Worse, the DPI portion of a MB typically takes between 30-80% of the work done by a MB. Combining DPI engines to perform multiple NFVs is an active area of research. One approach used by Dr. Bremler-Barr's group is DPI as a service.

This consolidates the DPI engine to a central location, adds the matching rules to the Network Service Header, and forwards to the rest of the DPI chain. Most DPI engines will use simple DFA string matching. Regular expressions are typically supported by extracting required substrings from the expression and adding them to the string matching table (Aho-Corasick), and only running the full regular expression if all the constant substrings are found in the input. This is because DFA regex engines suffer from table explosion very quickly, and NFA engines spend a significant amount of time backtracking.

Yotam Harchol's work[2] was to combine the various components of MBs into a framework for developing new MBs. He extracted the component graph and did work on merging graphs for similar NVFs.

3.3 P4: Programming Protocol-independent Packet Processors

The next programming homework will focus on P4.

The gist is that P4 is the next step for SDN. The idea is that OpenFlow is a rule-based control system, whereas P4 is a programmable logic approach. In this case, P4 abstracts the chip logic for the data plane. This allows more complex logic to help implement new link, network, and transport protocols.

4 Spoofed Attacks

The basic concept of spoofing is that you masquerade your source address to hide where you are located. Generally speaking, a "Good Netizen" will filter egress traffic that originates in their own network has a source address from within their own network. However, this is based on good will and cooperative behavior.

4.1 Unicast Reverse Path Forwarding

In strict mode, enforces that the receiving interface is shorter to the source. This breaks for asymmetric routes or multi-homing.

In loose mode, enforce that the receiving interface has a path to the source. This basically enforces that the source is routable (non-martian or bogon).

In feasible mode, enforce that among k paths to the source, the receiving interface is among the n best. Theoretically speaking, loose is when $k = n$ and strict is when $k = 1$.

4.2 SYN flood

An attack invented in 1996 by a 16-year old known as "mafiaboy". The idea is to send lots of SYN packets to force servers to allocate resources. Due to extremely small backlog sizes and long timeouts, it was sufficient to send 10 packets every 3 minutes to push over most servers.

4.2.1 Measurement of attack velocity

By measuring the replies, we can gather a statistical idea of how much traffic is being sent to a victim.

4.2.2 SYN Cookies

Proposed by Daniel Bernstein. Only saves state for fully synchronized connections by including "random" sequence number by using the top 5 bits of the clock. On ACK, if the sequence number matches, then it's a legit attack.

[2] OpenBox Sigcomm 2016

4.2.3 TCP Intercept

Protects against SYN floods by using SYN Cookies before proxying "good" connections to the actual server.

References

[1] Mark Crovella and Balachander Krishnamurthy. *Internet Measurement: Infrastructure, Traffic and Applications*. John Wiley & Sons, Inc., New York, NY, USA, 2006.

[2] James F. Kurose and Keith Ross. *Computer Networking: A Top-Down Approach Featuring the Internet*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2nd edition, 2002.

[3] George Varghese. *Network Algorithmics,: An Interdisciplinary Approach to Designing Fast Networked Devices (The Morgan Kaufmann Series in Networking)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2004.