Introduction to Network Function Virtualization

Introduction

1. Decouple network services needed for any enterprise, such as a firewall or malware inspection, from proprietary hardware appliances often referred to as "middleboxes"
    - Services can be run as software entities on standardized services using virtualization
    - Network services become virtual network functions

NFV Overview

1. Introduction to types of network functions and the pathway to virtualizing them
2. A concrete example of a technology enabler for virtualizing network functions
3. Synergy between SDN and NFV
4. NFV on public cloud

What are Network Functions?

1. Types of network functions
    - Firewall
        - Filters traffic based on pre-defined rules
        - Rules are simple since filtering is in the critical path of packet flow
    - Intrusion detection/prevention
        - Perform more complicated analysis of packet traffic
        - Identify complex patterns fo network traffic belonging to an attack or suspicious activity
    - Network Address Translation (NAT)
        - Translates private IP address space to public IP address space and vice versa
        - Useful for organizations that have limited public IP network presence
    - WAN optimizers
        - Reduce WAN bandwidth consumption of an enterprise
        - Perform multiple techniques like caching, traffic compression, etc for reducting traffic and latency
    - Load balancer
        - Distribute traffic to a pool of backend services
    - Virtual Private Network (VPN) Gateway
        - Provides abstraction of same IP address space for network that are physically separate
        - Multiple sites communicate over WAN using tunnels between gateways
2. Why do enterprises need network functions?
    - Users view enterprises (Google, Amazon) as a monolith
    - Internal view of an enterprise computing environment
        - Clusters of machines serving many internal functions (sales, marketing, inventory, purchasing)
        - Employees access these clusters on-premise and remotely
    - Enterprises may have several points of presence
        - Interconnected by WAN
    - Enterprises may inter-operate
        - Internet connects Intel, Samsung, Microsoft, etc.
3. Network functions give the necessary safeguards and facilities for enterprises
    - Intrusion prevention: Performs inspection of packet payload to identify suspicious traffic
    - Firewall: Filters packets based on their src, dst IPs, ports and protocol
    - Load balancer: Evenly distributes incoming connections to one of the backend servers
    - WAN Accelerator: Reduces WAN bandwidth consumption by data deduplication and compression
    - VPN: Provides illusion of same network address space across multiple sites and provides encryption for inter-site traffic

Middleboxes

1. Middleboxes: Standalone hardware boxes (aka network appliances) providing specific network functions (e.g., firewall)
2. Consider the example of a retail organization (like Walmart) that holds inventory information on premises, but uses and enterprise datacenter for long- running batch processing (demand prediction, etc.)
    - End-clients communicate with on-premise application
        - Needs to scale horizontally to handle peak traffic -> need for load balancer
        - Limit ports for traffic -> need for firewall
        - Detect/prevent suspicious activity -> need for intrusion prevention
    - Communication with enterprise datacenter
        - Need for VPN for encryption of traffic and illusion of continuous IP address space
        - Need for WAN accelerator to reduce WAN bandwidth usage (reduce $)
    - Office personnel access content on the Internet
        - Firewall checks and filters the websites accessed (and blocks restricted ones)
        - WAN accelerator contains HTTP proxy that can cache content (reducing WAN bandwidth and $)
3. Middleboxes (network appliances)
    - Computer networking devices that analyze/modify packets
        - For purposes other than packet forwarding
    - Typically implemented as specialized hardware components

Examples of Middleboxes

1. Intrusion prevention system (IPS)
    - Security appliance
    - Monitors all open connections to detect and block suspicious traffic
    - Sysadmin configures signatures in IPS box to detect suspicious traffic
    - Can work in inline mode (can filter out suspicious traffic) or passive mode (analyzes packets outside critical/data path)
    - Tabel shows the various traffic signatures that Cisco's IPSs are pre- configured with
    - The system administrator can select all or any of these signatures to be searched for in packet traffic
    - This particular screenshot is for a search result for "botnet" showing 10 signatures that characterize botnet traffic

IPS Signatures

Signature ID	Signature Name	Latest Release Date	Alarm Severity	Release
21806/0	Swizzor Botnet Traffic	2017 Feb 03	High	S965
22142/0	SQL Botnet User-Agent uil2pn	2017 Feb 03	High	S965
11025/0	IRC DCC File Transfer	2017 Feb 03	Informational	S965
6537/0	Kraken Botnet Traffic	2017 Feb 03	High	S965
6537/1	Kraken Botnet Traffic	2017 Feb 03	High	S965
16754/0	PushDo Botnet	2017 Feb 03	High	S965
5440/0	IRC Bot Activity	2017 Feb 03	Low	S965
6013/0	IRCBOT_JK DNS Lookup	2017 Feb 03	High	S965
6013/1	IRCBOT_JK DNS Lookup	2017 Feb 03	High	S965

Botnet Traffic

2. HTTP Proxy

  - Performance-improving appliance
  - Caches web content to reduce page-load time
  - Reduces bandwidth consumption
  - Can filter out blocked websites
3. Middleboxes in core cellular networks
    - Serving gateway (S-GW)
        - Responsible for routing/forwarding of packets
        - Executes handoff between neighboring base stations
    - Packet gateway (P-GW)
        - Acts as interface between cellular network and Internet
        - NAT between internal IP subnet and Internet
        - Traffic shaping
    - Mobility Management Entity (MME)
        - Key control node of LTE (Long Term Evolution)
        - Performs selection of S-GW and P-GW
        - Sets up connection when device is roaming
    - Home Subscriber Server (HSS)
        - User identification and addressing using IMSI (International Mobility Subscriber Identifier) number
        - User profile info: service subscription rates and QoS
4. How are middleboxes different from routers and switches?
    - Middleboxes are stateful
        - Packet processing is dependent on fine-grained state
        - Updated frequently (per packet/per connection)
    - Middleboxes perform complex and varied operations on packets

Network Management and Proliferation of Middleboxes

1. Similar challenges that motivated shift of IT to cloud services
2. Leads to lock-in to the hardware vendor of each specific middlebox
    - Difficult and expensive to migrate to a different solution
3. Failures of middleboxes lead to network outages
4. High capital and operational expenditure
    - Provisioning is done based on peak capacity
    - Management/maintenance cost is high

Network Services as Software Entities

1. Network functions as software entities on COTS servers
    - Replace middleboxes with software entities
    - Run such network functions as an "application" on general-purpose servers
    - Benefits
        - Low cost of deployment
        - Better resource utilization
        - Scaling is easily possible: lower capital expense
        - Can switch between vendors easily
        - Failures are easier to deal with
2. Examples of "software" middleboxes
    - Linux iptables: provides NAT and firewall
    - SoftEther VPN
    - Squid HTTP proxy
    - nginx load balancers
    - Bro Intrusion Detection System (1999)
3. Fundamental components of software middleboxes

- Use Unix sockets -> opening a socket creates a file descriptor
- Use system calls read() and write() to Linux kernel for reading and writing to a socket
  - Raw Linux sockets enable developer to read/write raw bytes (MAC layer data) from/to NIC
4. Architecture of a load balancer network function
   - Distribute client connections to a pool of backend service instances
     - For example, HTTP server
   - Use packet's 5-tuple to choose backend instance
     - 5-tuple: src address, dst address, src port, dst port, protocol
     - Provides connection-level affinity
     - Same connection is sent to same backend instance
   - What happens when a packet arrives?
     - NIC uses DMA to write incoming packet to memory
     - NIC generates an interrupt
     - CPU handles the interrupt, allocates kernel buffer and copies DMA'd packet into buffer for IP and TCP processing
     - After protocol processing, packet payload is copied to application buffer (user-space)

Virtualization Technology

1. Why virtualization for NF?
   - Using a VM for hosting NF (instead of running NF on bare metal servers)
     - Better portability because entire environment can be deployed, all dependencies are inside VM image
     - Network management becomes easier
     - Each NF instance is shielded from software faults from other network services
2. How to virtualize?
   - Traditionally two approaches
     - Full virtualization
     - Para virtualization
   - Full virtualization is attractive since the VM on top of hypervisor can run unmodified
     - "Trap-and-emulate" technique in the hypervisor to carry out priveleged operations of the VM which is running in user mode
     - Unfortunately, for network functions that are in the critical path of packet processing this is bad news
3. How "Trap-and-emulate" works
   - I/O is performed via system calls (priveleged operation)
   - When guest VM performs I/O operation
     - Executes system call
     - Guest kernel is context switched in
     - Priveleged instructions are invoked for reading/writing to I/O device
   - But guest kernel is actually running in user-space!
     - Guest VM is a user-space program from the host's perspective
     - Execution of priveleged instruction by user-space program results in a trap
   - Trap is caught by the hypervisor
     - Performs the I/O on behalf of the guest VM
     - Notifies the guest VM after I/O operation finishes
4. Downsides of "Trap-and-emulate" for NF
   - Host kernel (e.g., Dom-0 in Xen) has to be context switched in by the hypervisor to activate the network device driver and access the hardware NIC
   - Duplication of work by the virtual device driver in the guest and the actual device driver in the host
   - NF incurs the above overheads
     - For each packet that is sent to the NIC
     - For each packet received from the NIC

- NF is in the critical path of network processing and such overheads are untenable

Eliminating Overhead of NFV

1. How do we solve the overhead problem?
    - Fortunately, hardware vendors have been paying attention
    - We will mention two approaches to eliminate I/O virtualization overheads
        - Intel VT-d
        - Intel SR-IOV
2. Intel Virtualization Technology for Directed I/O (VT-d)
    - Allows efficient access to host I/O devices (e.g., NIC)
    - Avoids overheads of trap-emulate for every I/O access
    - Allows remapping of DMA regions to guest physical memory
    - Allows interrupt remapping to guest's interrupt handles
    - Effectively direct access for guest machine to I/O device hardware
    - Configuration registers are mapped to guest VM's memory for memory-mapped I/O
3. Benefits of VT-d
    - Avoid overheads of trap-and-emulate
    - DMA by NIC is performed to/from memory belonging to guest VM's buffers
    - Interrupts are handled directly by the guest instead of hypervisor
    - Effectively, the NIC is owned by the guest VM
4. Single Root I/O Virtualization (SR-IOV) Interface
    - An extension to the PCIe specification
    - Each PCIe device (physical function) is presented as a collection of virtual functions
    - Practical deployments have 64 VFs per PF
    - Each virtual function can be assigned to a VM
    - Allows higher multi-tenancy and performance isolation
    - Separate config register space for each VF
5. Benefits of SR-IOV
    - Allows same physical NIC to be shared by multiple guest VMs without conflicts

Putting it All Together

1. Virtual Network Function implementation
    - Host machine (NICs, etc.)
    - SR-IOV
    - VT-d direct access
    - Virtual machine with DPDK driver
        - DPDK covered in the next lecture
    - NFs implemented as a user-space application running inside VM
    - DMA from SR-IOV VF directly into VM buffers

Conclusion

1. Hardware vendors made middleboxes to handle specialized network functions
    - Middleboxes sat between enterprise computing and wide-area Internet
    - This became a network-management nightmare and would make cloud computing an impossibility
    - To ensure these functions run in a platform-agnostic manner, it makes sense to execute them in a virtual layer
        - Intel developed special hardware to eliminate the overhead inherent to virtualizing these functions