Openstack Nova Security groups

A concept overview of security groups in nova is present at [1]. In general, security groups have rules (eg: filter, nat) and are associated to instances on compute nodes belonging to projects. Every security groups map to a one of the firewall filters, which is configured in the nova conf using the flag --firewall_driver [2]. There are two firewall_drivers implemented in nova - iptables and network filters (nwfilter). The default is iptables but all instances get a basic nwfilter applied, which provides protection against MAC/IP/ARP spoofing. Security groups for a vm instance are passed at launch time by the cloud controller to the compute nodes and applied at the compute node when a vm is spawned. Data models are created for security groups. These are referenced and modified based on API requests [3] and subsequently rules on all nodes that have instances belonging to the modified security group are updated. rpc.cast messages is used to communicate with the compute nodes. Filters are removed when an instance is destroyed and are ensured that they exist on the host upon vm instance live migration. From an api perspective - trigger_security_group_rules_refresh gets invoked when a rule is added or removed from a security group and add_security_group/remove_security_group called when security groups are added and deleted resp. trigger_security_group_members_refresh() is called when a security group gains or loses a member.

There are static filters and are configured at the start of any instance. The following filters are pre-defined in nwfilter format -
- "nova-base" filter - consists filter references to - no-mac-spoofing, no-ip-spoofing, no-arp-spoofing and allow-dhcp-server
- "nova-vpn" consists filter references to - allow-dhcp-server

- "nova-base-ipv4" consists rules to allow outgoing and drop incoming tcp, udp and icmp traffic
- "nova-base-ipv6" - same as above
- "nova-allow-dhcp-server" - allows incoming and outgoing dhcp traffic from the dhcp server ip
- "nova-ra-filter" - allow inout traffic from ra server ipv6
- "nova-project" - allows incoming tcp, udp and icmp traffic from the instances in the same project network
- "nova-project-v6" - same as above
- "nova-instance-xxxx" - references either nova-base or nova-vpn based on the instance type

The nwfilters allow each vm instances network traffic filtering rules to be configured individually on a per interface basis using the instance XML description file in libvirt. The rules are applied on the host when the virtual machine is started and can be modified while the virtual machine is running. The latter can be achieved by modifying the XML description of a nwfilter. The XML attribute "interface" supports three types for nwfilters – network, ethernet and bridge [4] (interface type “direct” doesn’t seem to be supported)

Every security groups consists of one or more of the static filters (described above) based on the instance settings. "nova-Instance-secgroup" references these static filters in addition to any user defined security groups to allow incoming traffic into the instance corresponding to parameters defined in the security group rule. security_group_to_nwfilter_xml() in firewall.py is used to translate rules in security groups into an XML format. A provider filter ("nova-provider-rules") is also created that blocks incoming traffic based on firewall parameters.

Currently, the iptables firewall filter uses the iptables_manager class implemented in linux_net.py structured under nova/network. Classes

and methods to add/remove chains and rules to table type "filter" in iptables are provided by linux_net.py. The "apply" method provides the functionality to atomically apply a set of chains and rules to the nodes after any new addition or modification. Similar to the nwfilter, the instances have static chains and rules defined at the start (""inst-xxx") and security groups ("nova-sg-xxxx") and provider firewall rules ("provider") that also get translated to iptables chains and rules. instance_rules() in firewall.py provides ip rule implementation.

xenapi-security group blueprint detail is provided at [5]

[1] http://nova.openstack.org/nova.concepts.html#concept-security-groups

[2] - http://docs.openstack.org/diablo/openstack-compute/admin/content/hypervisor-configuration-basics.html

[3] http://wiki.openstack.org/os-security-groups

[4] http://libvirt.org/formatnwfilter.html

[5] http://wiki.openstack.org/xenapi-security-groups