includeOS

About the Company

IncludeOS was started in 2016, as a spinout from Oslo Metropolitan University by researcher Alfred Bratterud.

The company worked together with Norway's leading managed hosting provider, Basefarm, to create a new generation of Network Function Virtualization software, based on research done at the university.

Supported Hypervisors

The IncludeOS NFV suite supports VMware VSphere as well as the Openstack suite. Support for Microsofts HyperV platform is planned.

Small and efficient

IncludeOS® NFV instances only take up about 10 megabytes of disk storage each and can use as little as 128 megabytes of memory, allowing you to run hundreds or thousands of instances in your network.

Introduction

The IncludeOS Network Function Virtualization (NFV) Platform allows you to deploy fast, small, secure and cost-effective virtual machines that can provide network functions in your virtual network.

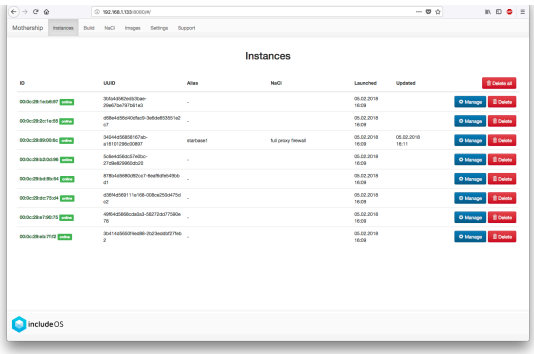
The solution consists of a central administration node and distributed virtual machines running network functions in the network.

Operating System

The IncludeOS operating system is a Unikernel design. Instead of the traditional operating system design where the application and operating systems interact, in a dynamic manner, IncludeOS applications have the operating system functionality built in, giving great benefits concerning security, performance, and manageability.

The IncludeOS Mothership management platform

The Mothership is the management platform that manages the individual NFV instances. When a new NFV instance is deployed it boots up and connects to the Mothership. It then awaits instructions, and when instructed it takes the role of a firewall or load balancer. Mothership builds the image with the application and configuration compiled into the image. You can think of



Modern web UI

IncludeOS Mothership comes with a modern web UI giving you full control over your running fleet of IncludeOS instances. There is a built in editor for NaCl configuration complete with syntax validation and highlighting.

Fully API driven

The Mothership comes with a complete and documented API, making it easy to integrate into your existing configuration management system. This makes it possible to automate every aspect of the NFV management.

Protocol support

IncludeOS supports IPv4 and IPv6, auto configuration through DHCP and SLAAC.

Firewall rules can operate on IPv4 and IPv6. Load balancing support is available for TCP and TCP6 applications.

Demonstration

If you are interested in getting a demonstration of IncludeOS® Mothership, please get in touch with us. For demonstration purposes the system can be deployed in a cloud environment or on your local hypervisor.

You can reach Per Buer, our CEO, at perbu@includeos.com.

it as a "build to order" system. Building the image takes just a couple of seconds as all of the components are pre-compiled already.

When the image is built, it is transmitted to the virtual machine and executed. The virtual machines will from this moment on serving the given networking function in the network.

Disruption-free updates with LiveUpdate

When the Mothership updates an instance, this happens through the unique LiveUpdate mechanism. LiveUpdate allows the Mothership to "hot swap" the image running on the instance without downtime.

When a configuration change happens, the image is rebuilt and deployed. The instance stores its current state, connection tracking tables and the like, and then the new image is executed, the state is restored, and execution continues where it left off. There is no noticeable downtime.

Security

Perhaps the most significant advantage that IncludeOS NFV instances have is security. The security characteristics are:

No listening ports.

IncludeOS NFVs don't listen on the network. When they boot up, they make an outgoing TLS connection to their Mothership, and they only take input from this connection.

Minimal attack surface.

When the Mothership builds the image, it strips it of unused code. So an IncludeOS based NFV instance does not have a shell or any other software that might be exploited to gain or elevate privileges.

No support for host-initiated reconfiguration.

As mentioned above, an IncludeOS NFV cannot reconfigure itself. The only way to alter the instance behavior is to push a new image through Mothership. The ability for an application to alter itself doesn't exist in IncludeOS. An attack would have to deploy an entirely new image to alter the system. As most attack vectors have limited payload, this makes an attack very hard to pull off.

No system calls.

Applications running under IncludeOS interact with the operating system code using function calls, which are invoked using 64-bit pointers, in stark contrast to traditional Linux and Unix-based systems where the system calls are invoked using preset tables where each system call has an integer identifier. As most exploits rely on system calls to persist the attack, the attacker's task becomes very hard as the OS functionality is available at more or less random locations in memory. As the operating system itself does not reside in a filesystem, this makes it harder still.

Configuration

The Mothership allows you to define its configuration through a domain-specific programming language, NaCl. The NaCl configuration is translated to machine code and embedded into the image.

For a firewall, this means the firewall application is unable to alter its own rules, in contrast to Unix- or Linux-based systems, where a privileged user can execute shell commands to alter the local firewall rules. In the IncludeOS operating system, the system doesn't know how to modify itself. The topic is explored further in the section named “Security.”

In addition to security advantages, there are also performance advantages. A firewall rule-set created gets optimized for the processor during compilation and linking. It ensures the rules run almost instantaneous. Your firewall rule-set can contain thousands of rules without it significantly impacting performance.

Lastly, the NaCl rules are made to be humanly readable. Having security mechanisms that are easy to read and understand is crucial to maintaining a proper level of security.

```
Filter::IP firewallchain {
    if (ct.state == established) {
       syslog(INFO, “Packet on established connection (from ", ip.saddr, ")")
        accept
    }
    if (ip.saddr == bastion_host) {
        accept
    }
    Filter::TCP {
            if (ip.daddr in allowed_hosts and tcp.dport in allowed_services) {
                    syslog(INFO, "New connection established (from ", ip.saddr, ")")
                accept
            }
    } // end TCP

    syslog(WARNING, "Dropping packet from saddr ", ip.saddr," to ",ip.daddr)
    drop
}
```

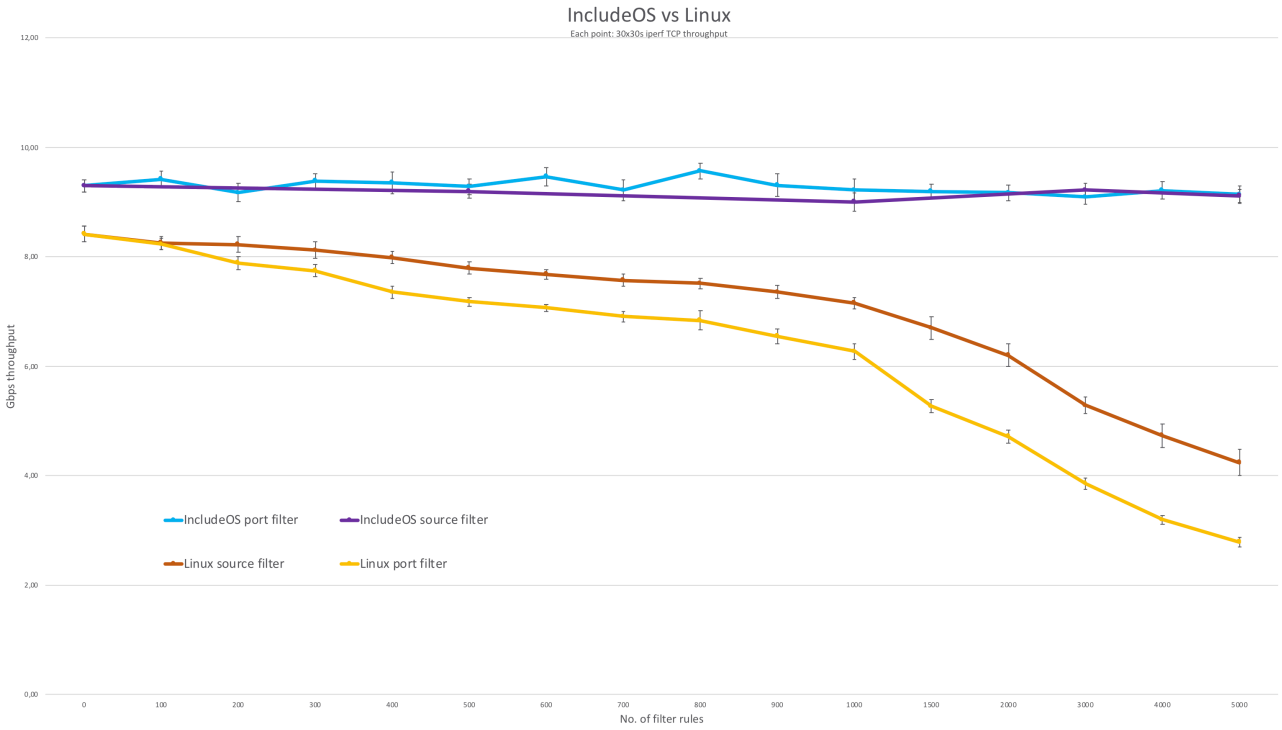
Performance

The Mothership takes the given configuration and compiles it to binary code. For configurations that consist of rules this gives a unique performance edge. The resulting code can take advantage of features such as speculative execution to parallelize operations. The resulting code-stream is very instruction cache friendly, as all the parameters for the operations get inlined into the binary stream.

Together with Oslo Metropolitan University we've run benchmarks on IncludeOS and compared it to the industry standard Iptables/Netfilter implementation in the Linux kernel. The observation is that the performance stays more or less consistent without regard to the complexity of the ruleset.

Oslo Metropolitan University ran benchmarks, and the results were clear. IncludeOS, which starts with approximately 15% higher throughput than a Linux-based router, is more or less unaffected by the addition of firewall rules. Even with as many as 5000 firewall rules, the performance degradation following such a complex ruleset is only a couple of percent.

In the graph below, you can see how IncludeOS maintains the throughput with the number of rules increasing from 0 to 5000, whereas Linux throughput drops as the number of rules increase.



Logging

IncludeOS supports logging through your conventional syslog mechanisms, allowing you to reuse your existing logging infrastructure.