Cloud Network Services, Nokia
Secure Systems Group, Aalto University

Vishal Verma, Tuomas Aura, Martin Peylo (Nokia), Jacopo Bufalino, Jose Luis Martin-Navarro

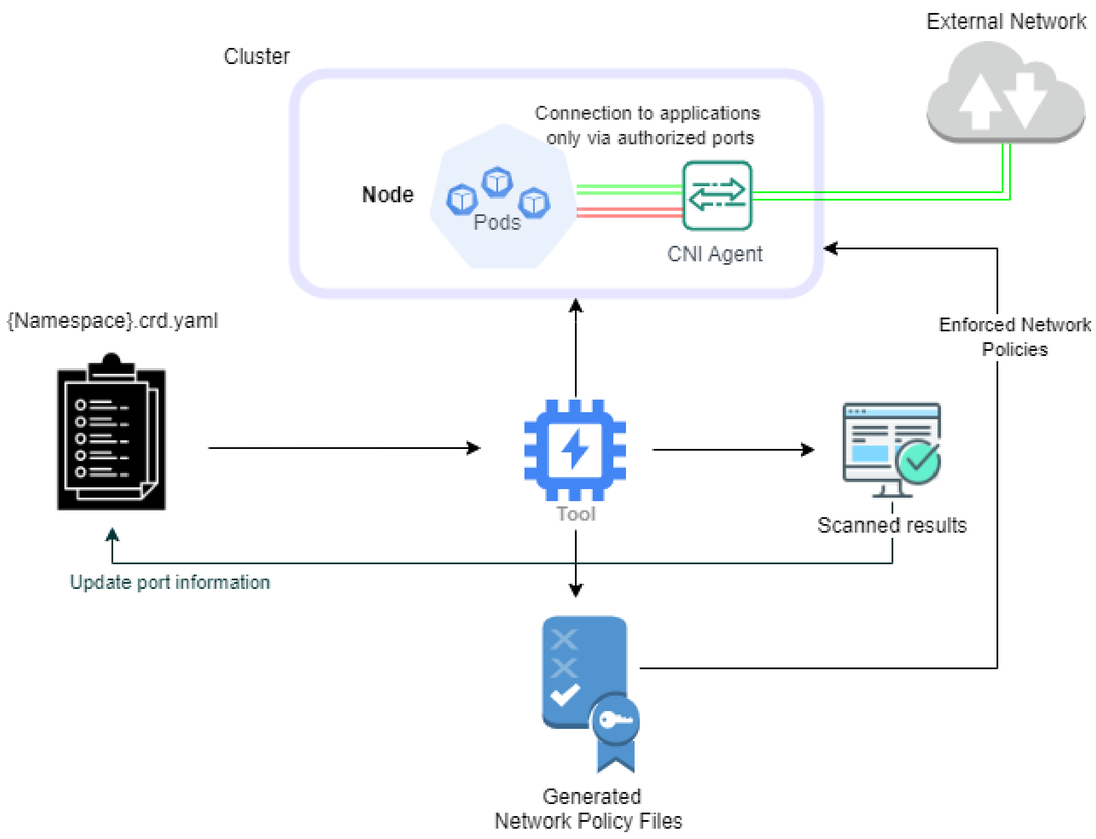
Network Security Policies for Containers in Cloud Applications

Problems

The existing solution does not offer adequate security measures such as proper port scanning and restriction features, leaving clusters susceptible to unauthorized access and compromising application security. Additionally, there is currently no comprehensive solution in the market that provides a complete overview of open ports and the ability to establish and enforce network policies using Container Network Interface (CNI) agents to restrict unauthorized ports and prevent tunneling into the cluster. This lack of functionality hinders the effective security of the cluster and puts all the deployed applications at risk of potential security threats.

Solution

The proposed solution addresses the need to limit unauthorized ports allowed to and from the cluster, which involves utilizing a standard template file created by the developer. This file is used to define all the permitted connections. Port detection is performed using tools such as lsof and nmap, which identify open ports within specified namespaces and pods. By comparing the detected ports with the ports documented in the template files, the authorization status of each port is determined. Subsequently, Kubernetes network policies are established and enforced through a CNI agent. Each namespace comprises two YAML files: "-block.yaml", which blocks all traffic, and "-allow.yaml", which only allows authorized ports.



Implementation

We propose a new tool that follows an order of events to address various network security challenges. Firstly, it performs scanning processes to identify compromised containers. Then, it creates network policy files using the information in the template file. Finally, it enforces these Kubernetes network policies through CNI agents, limiting unauthorized port access and mitigating security risks.

```
:::::Running function 'port_scan':::::
................................................
Namespace    Pod       Open Ports  Protocol    Authorized Ports
-----------  ------  ------------  ----------  -----------------
iperf        iperf           5000  UDP         ✓
iperf        iperf          50600  UDP         ✓
iperf        iperf           8000  TCP         X
nginx        nginx             80  TCP         ✓
nginx        nginx          12000  UDP         X
server       server          7777  TCP         ✓

:::::Running function 'create_policy_handler':::::
................................................
Generated network policy file: allowserver.yaml
Generated network policy file: blockall_server.yaml
Generated network policy file: allownginx.yaml
Generated network policy file: blockall_nginx.yaml
Generated network policy file: allowiperf.yaml
Generated network policy file: blockall_iperf.yaml

:::::Running function 'apply_yaml_files':::::
................................................
networkpolicy.networking.k8s.io/allowiperf created
networkpolicy.networking.k8s.io/blockalliperf created
networkpolicy.networking.k8s.io/allowserver created
networkpolicy.networking.k8s.io/blockallserver created
networkpolicy.networking.k8s.io/allownginx created
networkpolicy.networking.k8s.io/blockallnginx created

:::::Showing all network policy files in all namespaces:::::
................................................
NAMESPACE   NAME             POD-SELECTOR   AGE
iperf       allowiperf       <none>         3s
iperf       blockalliperf    <none>         3s
nginx       allownginx       <none>         1s
nginx       blockallnginx    <none>         0s
server      allowserver      <none>         2s
server      blockallserver   <none>         2s
```

By following this sequence, the tool ensures the efficient identification, verification, creation, and enforcement of network security policies for containerized applications within Kubernetes clusters.

Aalto University
School of Science

Contact: vishal.verma@aalto.fi, tuomas.aura@aalto.fi, martin.peylo@nokia.com, jacopo.bufalino@aalto.fi, jose.martinnavarro@aalto.fi