

istio proxy의 작동원리

istio는 쿠버네티스 클러스터 내에서 네트워크 흐름을 통제하는 오픈소스이다. istio는 Envoy 프록시를 사이드카(sidecar) 형태로 배포하여 트래픽을 관리한다. 프록시 컨테이너는 iptable을 조작하여 네트워크 패킷을 중간에 가로챈다. 가로채는 과정은 로그로 확인하기 어려워서, istio 사용 시 트래픽 흐름을 이해하지 못하는 경우가 많다.



사이드카 주입과 init 컨테이너

쿠버네티스 네임스페이스에 **istio-injection=enabled** 라벨을 붙이면 webhook을 통해 istiod에 /inject API를 호출한다. istiod는 파드에 프록시 컨테이너와 init컨테이너를 주입한다. init 컨테이너의 이름은 istio-init이다.

```
Init Containers:
  istio-init:
    Container ID:  docker://a44c572539d91e656dd44af7101850ef0a97b69416cb944f48e2fcdef73838f1
    Image:         docker.io/istio/proxyv2:1.18.1
    Image ID:      docker-
    Pullable:      //istio/proxyv2@sha256:7d1f3b4876e1652b344be6601ab9a62e454828503eeca59a94a406ea86be1fd
    Port:          <none>
    Host Port:     <none>
    Args:
      istio-iptables
      -p
      15001
      -z
      15006
      -u
      1337
      -m
      REDIRECT
      -i
      *
      -x
      -b
      *
      -d
      15090,15021,15020
      --log_output_level=default:info
```

init 컨테이너는 서비스 컨테이너가 생성되기 전에 먼저 작업을 수행한다. 작업의 내용은 istio-iptables 명령어이다. istio-iptables는 네트워크 트래픽을 제어하기 위한 iptable 작업을 수행한다.

- u: Redirect 하지 않을 UID. 보통 프록시의 UID를 지정한다.
- p: 모든 TCP 트래픽을 Redirect 할 Envoy 포트
- z: 모든 Inbound 트래픽을 Redirect 할 포트
- b: Redirect 할 Inbound Port
- d: Envoy로 Redirect 하지 않을 Inbound Port.
- i: Redirect 할 IP Range
- x: Envoy로 Redirect 하지 않을 IP Range

1. 프록시는 15006번 포트로 인바운드 트래픽을, 15001번 포트로 아웃바운드 트래픽을 수신한다.

2. 프록시는 1337 UID를 사용하고, 프록시의 패킷은 프록시로 라우팅 하지 않는다.프록시 -> 프록시로 패킷

3. 이 라우팅 되면 무한루프에 빠질 수 있다.

4. 프록시는 15090,15021,15020 포트를 목적지로 한 패킷을 제외한 모든 트래픽을 가로챈다.

이렇게 init 컨테이너를 통해 iptable에 규칙을 추가한 후에, istio-proxy 컨테이너를 생성한다. istio-proxy는 init 컨테이너가 라우팅에서 제외시켜 준 1337번 UID를 사용한다. 프록시가 주입된 파드의 템플릿을 보면 다음과 같이 runAsUser값이 1337임을 확인할 수 있다.

```
image: docker.io/istio/proxyv2:1.18.1
imagePullPolicy: IfNotPresent
name: istio-proxy
(...)
securityContext:
  allowPrivilegeEscalation: false
  capabilities:
    drop:
      - ALL
  privileged: false
  readOnlyRootFilesystem: true
  runAsGroup: 1337
  runAsNonRoot: true
  runAsUser: 1337
```

iptables 이해하기

istio는 iptable을 조작하여 프록시로 트래픽을 전달한다. 여기서 iptable은 파드의 네임스페이스 내 iptable을 의미한다. 각 파드는 독립된 프로세스이고, 파드별로 별도의 리눅스 네임스페이스를 가진다. 각 파드 내에서 프록시는 15001, 15006번 포트를 사용하고, 자신의 리눅스 네임스페이스에 정의된 iptable 규칙에 따라 패킷을 라우팅 한다.

특정 파드의 네임스페이스에서 init 컨테이너가 생성한 iptable을 살펴보면 다음과 같다.

```
root@minikube:~# iptables -t nat -L -v
Chain PREROUTING (policy ACCEPT 4094 packets, 246K bytes)
pkts bytes target      prot opt in     out    source      destination
4095 246K ISTIO_INBOUND tcp -- any   any    anywhere    anywhere

Chain INPUT (policy ACCEPT 4095 packets, 246K bytes)
pkts bytes target      prot opt in     out    source      destination

Chain OUTPUT (policy ACCEPT 321 packets, 29139 bytes)
pkts bytes target      prot opt in     out    source      destination
12 720 ISTIO_OUTPUT tcp -- any   any    anywhere    anywhere

Chain POSTROUTING (policy ACCEPT 321 packets, 29139 bytes)
pkts bytes target      prot opt in     out    source      destination

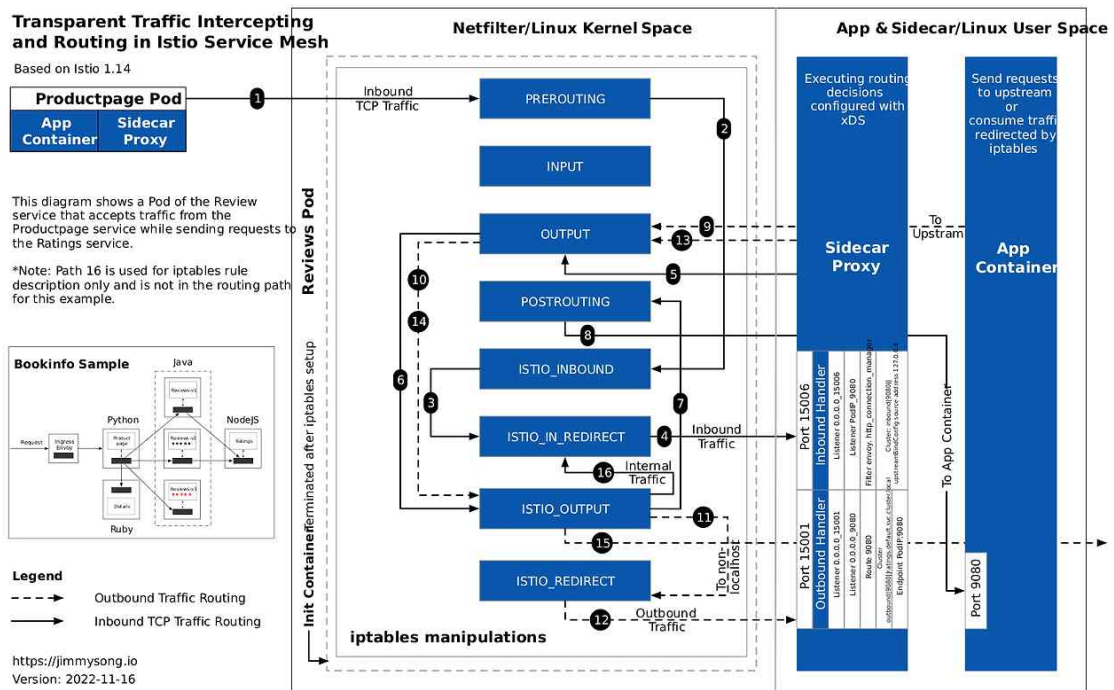
Chain ISTIO_INBOUND (1 references)
pkts bytes target      prot opt in     out    source      destination
0 0 RETURN      tcp -- any   any    anywhere    anywhere    tcp dpt:15008
0 0 RETURN      tcp -- any   any    anywhere    anywhere    tcp dpt:15090
4094 246K RETURN  tcp -- any   any    anywhere    anywhere    tcp dpt:15021
0 0 RETURN      tcp -- any   any    anywhere    anywhere    tcp dpt:15020
1 60 ISTIO_IN_REDIRECT tcp -- any   any    anywhere    anywhere

Chain ISTIO_IN_REDIRECT (3 references)
pkts bytes target      prot opt in     out    source      destination
1 60 REDIRECT   tcp -- any   any    anywhere    anywhere    redir ports 15006

Chain ISTIO_OUTPUT (1 references)
pkts bytes target      prot opt in     out    source      destination
1 60 RETURN      all -- any   lo     127.0.0.6  anywhere    anywhere
0 0 ISTIO_IN_REDIRECT tcp -- any   lo     anywhere    !localhost    tcp dpt:15008 owner UID match 1337
0 0 RETURN      all -- any   lo     anywhere    ! owner UID match 1337
11 660 RETURN    all -- any   any    anywhere    owner UID match 1337
0 0 ISTIO_IN_REDIRECT tcp -- any   lo     anywhere    !localhost    tcp dpt:15008 owner GID match 1337
0 0 RETURN      all -- any   lo     anywhere    ! owner GID match 1337
0 0 RETURN      all -- any   any    anywhere    owner GID match 1337
0 0 RETURN      all -- any   any    anywhere    localhost
0 0 ISTIO_REDIRECT all -- any   any    anywhere    anywhere

Chain ISTIO_REDIRECT (1 references)
pkts bytes target      prot opt in     out    source      destination
0 0 REDIRECT    tcp -- any   any    anywhere    anywhere    redir ports 15001
```

아래 그림을 통해서 이해해 보도록 하자.



트래픽이 외부에서 애플리케이션 컨테이너로 들어오는 경우(Port 9080)

1. 트래픽은 PREROUTING을 거쳐 ISTIO_INBOUND로 넘어간다. 왜냐하면 PREROUTING에서는 모든 트래픽을 ISTIO_INBOUND 타깃으로 전달하기 때문이다.

```
Chain PREROUTING (policy ACCEPT 4094 packets, 246K bytes)
pkts bytes target prot opt in out source destination
4095 246K ISTIO_INBOUND tcp -- any any anywhere anywhere
```

2. ISTIO_INBOUND에서 패킷은 ISTIO_IN_REDIRECT으로 넘어간다. 왜냐하면 "dpt:포트"로 지정된 규칙에 9080은 걸리지 않기 때문이다. 참고로 15008, 15090, 15021, 15020 은 모두 istio에서 공식적으로 사용하는 포트이다.

```
Chain ISTIO_INBOUND (1 references)
pkts bytes target prot opt in out source destination tcp dpt
0 0 RETURN tcp -- any any anywhere anywhere tcp dpt:15008
0 0 RETURN tcp -- any any anywhere anywhere tcp dpt:15090
4094 246K RETURN tcp -- any any anywhere anywhere tcp dpt:15021
0 0 RETURN tcp -- any any anywhere anywhere tcp dpt:15020
1 60 ISTIO_IN_REDIRECT tcp -- any any anywhere anywhere
```

3. ISTIO_IN_REDIRECT에서는 모든 패킷을 15006번 포트로 redirect 한다. 15006은 init 컨테이너가 지정한 프록시 포트이다. 프록시는 15006번 포트를 통해 모든 inbound 트래픽을 수신한다.

```
Chain ISTIO_IN_REDIRECT (3 references)
pkts bytes target prot opt in out source destination
1 60 REDIRECT tcp -- any any anywhere anywhere redir ports 15006
```

4. 15006번 포트를 통해 패킷을 전달받은 프록시는 아래의 규칙을 통해 트래픽을 전달한다. 복잡해 보이지만, 맨 밑에 **Addr: *:9080**만 보면 된다. 이 말은 9080번 포트에 가는 트래픽은 **Cluster: inbound|9080||**으로 보낸다는 의미이다.

```
$ istioctl proxy-config listener productpage-v1-7b4dbf9c75-v4z44 --port 15006
ADDRESS PORT MATCH
0.0.0.0 15006 Addr: *:15006
0.0.0.0 15006 Trans: tls; App: istio-http/1.0,istio-http/1.1,istio-h2; Addr: 0.0.0.0/0
0.0.0.0 15006 Trans: raw_buffer; App: http/1.1,h2c; Addr: 0.0.0.0/0
0.0.0.0 15006 Trans: tls; App: TCP.TLS; Addr: 0.0.0.0/0
0.0.0.0 15006 Trans: raw_buffer; Addr: 0.0.0.0/0
0.0.0.0 15006 Trans: tls; Addr: 0.0.0.0/0
0.0.0.0 15006 Trans: tls; App: istio,istio-peer-exchange,istio-http/1.0,istio-http/1.1,istio-h2; Addr: *:9080
0.0.0.0 15006 Trans: raw_buffer; Addr: *:9080
DESTINATION
Non-HTTP/Non-TCP
InboundPassthroughClusterIpv4
InboundPassthroughClusterIpv4
InboundPassthroughClusterIpv4
InboundPassthroughClusterIpv4
InboundPassthroughClusterIpv4
Cluster: inbound|9080||
Cluster: inbound|9080||
```

그러면 **Cluster: inbound|9080||** 은 무엇을 의미할까?

istioctl의 cluster 설정을 보면 다음과 같이 **inbound|9080||**의 상세 내용을 볼 수 있다. 여기서 **ORIGINAL_DST**란 원래 목적지인 파드 IP를 의미한다. 즉, 프록시는 해당 트래픽을 파드의 9080번 포트에 트래픽을 보낸다.

```
$ istioctl proxy-config cluster productpage-v1-7b4dbf9c75-v4z44 --port 9080 --direction inbound -o json
[
  {
    "name": "inbound|9080||",
    "type": "ORIGINAL_DST",
    "connectTimeout": "10s",
    "lbPolicy": "CLUSTER_PROVIDED",
    "circuitBreakers": {
      "thresholds": [
        {
          "maxConnections": 4294967295,
          "maxPendingRequests": 4294967295,
          "maxRequests": 4294967295,
          "maxRetries": 4294967295,
          "trackRemaining": true
        }
      ]
    },
    "upstreamBindConfig": {
      "sourceAddress": {
        "address": "127.0.0.6",
        "portValue": 0
      }
    },
    "commonLbConfig": {},
    "metadata": {
      "filterMetadata": {
        "istio": {
          "services": [
            {
              "host": "productpage.default.svc.cluster.local",
              "name": "productpage",
              "namespace": "default"
            }
          ]
        }
      }
    }
  }
]
```


여기서 한 가지 주목해야 할 사실은 **upstreamBindConfig.sourceAddress**이다. upstream으로 트래픽을 전송할 때 source 주소로 127.0.0.6을 지정하고 있다. 즉, 이제는 출발지가 **127.0.0.6**이 되는 셈이다. 이는 istio에서 지정한 **InboundPassthroughIP** 대역이다. 이 주소는 다음 라우팅 규칙에서 아주 중요한 역할을 하므로 반드시 기억하기 바란다.

5. 프록시는 트래픽을 파드의 9080 포트로 전달한다. 패킷은 OUTPUT을 통해 외부로 나갈 준비를 한다.

6. OUTPUT은 모든 트래픽을 ISTIO_OUTPUT으로 전달한다.

```
Chain OUTPUT (policy ACCEPT 321 packets, 29139 bytes)
pkts bytes target      prot opt in     out    source      destination
 12   720 ISTIO_OUTPUT tcp  --  any   any     anywhere    anywhere
```

7. ISTIO_OUTPUT은 source가 127.0.0.6인 규칙을 적용하여 lo(localhost) 인터페이스로 전달한다.

```
Chain ISTIO_OUTPUT (1 references)
pkts bytes target      prot opt in     out    source      destination
 1    60 RETURN      all  --  any   lo      127.0.0.6    anywhere
 0     0 ISTIO_IN_REDIRECT tcp  --  any   lo      anywhere     !localhost    tcp dpt:!!15008 owner UID match 1337
 0     0 RETURN      all  --  any   lo      anywhere     !owner UID match 1337
11   660 RETURN      all  --  any   anywhere  anywhere     owner UID match 1337
 0     0 ISTIO_IN_REDIRECT tcp  --  any   lo      anywhere     !localhost    tcp dpt:!!15008 owner GID match 1337
 0     0 RETURN      all  --  any   anywhere  anywhere     !owner GID match 1337
 0     0 RETURN      all  --  any   anywhere  anywhere     owner GID match 1337
 0     0 RETURN      all  --  any   anywhere  localhost
 0     0 ISTIO_REDIRECT all  --  any   any     anywhere    anywhere
```

8. 패킷은 POSTROUTING을 통해 애플리케이션 컨테이너로 전달된다.

트래픽이 애플리케이션 컨테이너에서 외부로 나가는 경우

9. 애플리케이션 컨테이너는 OUTPUT을 통해 트래픽을 외부로 내보낸다.

10. OUTPUT은 모든 트래픽을 ISTIO_OUTPUT으로 전달한다.

```
Chain OUTPUT (policy ACCEPT 321 packets, 29139 bytes)
pkts bytes target      prot opt in     out    source      destination
 12   720 ISTIO_OUTPUT tcp  --  any   any     anywhere    anywhere
```

11. ISTIO_OUTPUT의 규칙 중에 해당되는게 없기 때문에, 마지막에 나온 ISTIO_REDIRECT로 향한다. 참고로 UID/GID 1337번은 프록시로부터 트래픽이 나왔다는 의미이다.

```
Chain ISTIO_OUTPUT (1 references)
pkts bytes target      prot opt in     out    source      destination
 1    60 RETURN      all  --  any   lo      127.0.0.6    anywhere
 0     0 ISTIO_IN_REDIRECT tcp  --  any   lo      anywhere     !localhost    tcp dpt:!!15008 owner UID match 1337
 0     0 RETURN      all  --  any   lo      anywhere     !owner UID match 1337
11   660 RETURN      all  --  any   anywhere  anywhere     owner UID match 1337
 0     0 ISTIO_IN_REDIRECT tcp  --  any   lo      anywhere     !localhost    tcp dpt:!!15008 owner GID match 1337
 0     0 RETURN      all  --  any   anywhere  anywhere     !owner GID match 1337
 0     0 RETURN      all  --  any   anywhere  anywhere     owner GID match 1337
 0     0 RETURN      all  --  any   anywhere  localhost
 0     0 ISTIO_REDIRECT all  --  any   any     anywhere    anywhere
```

12. ISTIO_REDIRECT는 15001번 포트로 패킷을 전달한다. 즉, 지금까지 애플리케이션 컨테이너에서 프록시 컨테이너로 패킷이 전달된 셈이다.

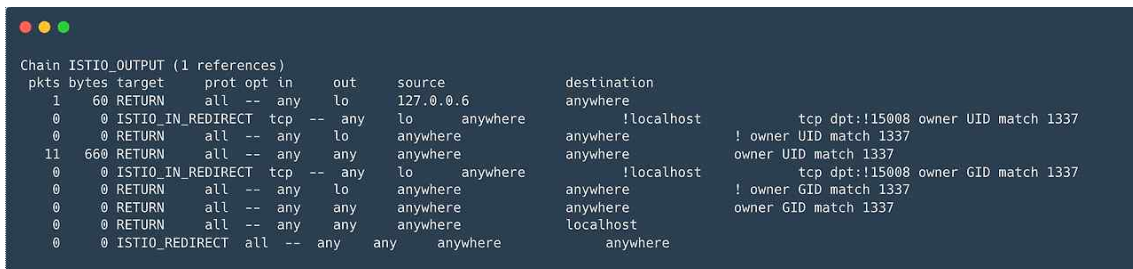


```
Chain ISTIO_REDIRECT (1 references)
pkts bytes target prot opt in out source destination
0 0 REDIRECT tcp -- any any anywhere anywhere redir ports 15001
```

13. 프록시는 OUTPUT을 통해 패킷을 외부로 내보낼 준비를 합니다.

14. OUTPUT은 모든 트래픽을 ISTIO_OUTPUT으로 전달한다.

15. 지금은 프록시가 내보낸 패킷이기 때문에 owner UID가 1337번이 된다. 따라서 다시 프록시로 돌아가지 않고, 바로 외부로 나간다.



```
Chain ISTIO_OUTPUT (1 references)
pkts bytes target prot opt in out source destination
1 60 RETURN all -- any lo 127.0.0.6 anywhere
0 0 ISTIO_IN_REDIRECT tcp -- any lo anywhere !localhost tcp dpt:15008 owner UID match 1337
0 0 RETURN all -- any lo anywhere ! owner UID match 1337
11 660 RETURN all -- any any anywhere owner UID match 1337
0 0 ISTIO_IN_REDIRECT tcp -- any lo anywhere !localhost tcp dpt:15008 owner GID match 1337
0 0 RETURN all -- any lo anywhere ! owner GID match 1337
0 0 RETURN all -- any any anywhere owner GID match 1337
0 0 RETURN all -- any any anywhere localhost
0 0 ISTIO_REDIRECT all -- any any anywhere anywhere
```

지금까지 트래픽이 파드 외부에서 내부로, 내부에서 외부로 전달되는 과정을 살펴보았다.

여기서 꼭 짚고 넘어가야 할 부분은 iptable을 거치는 순간 owner UID/GID가 1337인 경우이다. 이는 프록시가 트래픽을 내보냈다는 의미이므로, 다시 프록시의 15006번 포트로 들어가지 않는다. 이 부분이 없다면 트래픽은 영원히 제자리를 맴돌 것이다.