Oracle® Communications

Security Edge Protection Proxy (SEPP) Cloud Native User's Guide

Release 1.3.0
F32141-02
June 2020

Oracle Communications Security Edge Protection Proxy (SEPP) Cloud Native User's Guide, Release 1.3.0

F32141-02

Contents

List of Figures

List of Tables

My Oracle Support

My Oracle Support (https://support.oracle.com) is your initial point of contact for all product support and training needs. A representative at Customer Access Support can assist you with My Oracle Support registration.

Call the Customer Access Support main number at 1-800-223-1711 (toll-free in the US), or call the Oracle Support hotline for your local country from the list at http://www.oracle.com/us/support/contact/index.html. When calling, make the selections in the sequence shown below on the Support telephone menu:

1. Select **2** for New Service Request.
2. Select **3** for Hardware, Networking and Solaris Operating System Support.
3. Select one of the following options:
    - For Technical issues such as creating a new Service Request (SR), select **1**.
    - For Non-technical issues such as registration or assistance with My Oracle Support, select **2**.

You are connected to a live agent who can assist you with My Oracle Support registration and opening a support ticket.

My Oracle Support is available 24 hours a day, 7 days a week, 365 days a year.

Whats New In This Guide

This is the initial release of this document for Oracle Communications Security Edge Protection Proxy (OCSEPP) 1.3.0.

1
Overview

Oracle Communication Security Edge Protection Proxy (OCSEPP) is a proxy network functions (NF) which is used for secured communication between inter-Public Land Mobile Network (PLMN) messages. This document provides a brief overview of the recommended methods for configuring SEPP.

The OCSEPP supports the following functions:

1. Supports inter PLMN traffic for both SEPP MNO and IPX Proxy modes.
2. Supports containerized deployment based on micro service based architecture on cloud native environment.

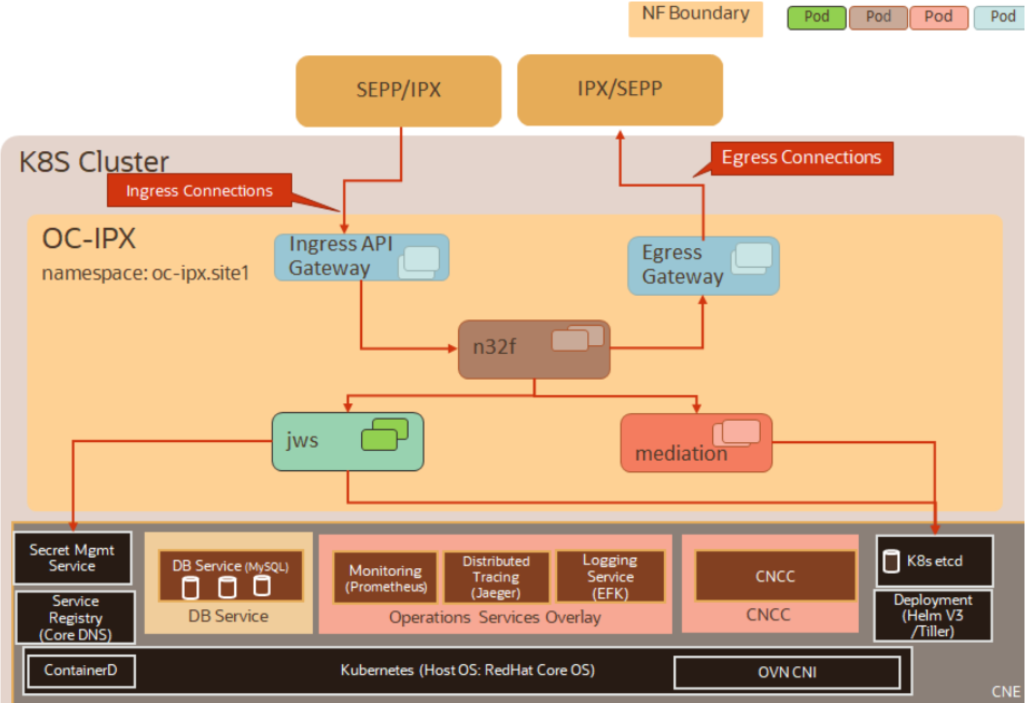
Security Edge Protection Proxy (SEPP) Architecture

This section explains the Security Edge Protection Proxy (SEPP) architecture.

Oracle SEPP is deployed by both Mobile Network Operators (MNO) and Internetwork Packet Exchange (IPX) providers. MNOs deploy OCSEPP acting as 3GPP 5GC SEPP NEF which enables inter PLMN communication between two networks via N32 interface.

IPX providers deploy OCSEPP acting as IPX Proxy enabling inter PLMN communication between MNO's through N32 interface supporting Application Layer Security (ALS) as specified in TS 33.501. IPX provider is able to deploy OCSEPP in either of one mode i.e. functionality of 5GC SEPP NF or IPX proxy. IPX providers are able to host 5GC SEPP NF functionality for few of the MNO's and host IPX proxy functionality for rest of the MNOs.

Figure 1-1   Security Edge Protection Proxy Architecture



The above architecture diagram shows an overview of SEPP deployment and functionality:

- **Ingress API Gateway :** Access point for incoming traffic. Provides TLS.

  n32f : forwards changes done by mediation and jws.

  jws : Adds java web service. Reads JWS key from secrets. Reads JWS attributes from k8s config map

  mediation : Applies modification based on rules. Read rules from k8s config map

  Egress Gateway: Egress traffic origination point. Provides TLS.

For information on installing SEPP, see the *OCSEPP Cloud Native Installation Guide*.

Acronyms

The following table provides information about the acronyms used in the document.

Table 1-1   Acronyms

Fields	Description
CNE	Cloud Native Environment
DNS	Domain Name System
FQDN	Fully Qualified Domain Name
NF	Network Function
OHC	Oracle Help Center

Table 1-1 (Cont.) Acronyms

Fields	Description
OSDC	Oracle Software Delivery Cloud
PLMN	Public Land Mobile Network
SEPP	Security Edge Protection Proxy
SVC	Services
TLS	Transport Layer Security

Reference

- CNE Installation Document
- Security Edge Protection Proxy (SEPP) Cloud Native Installation Guide

2 Configuring SEPP using rules

NF-Mediation Reloading ConfigMap, if in case any updates made in the Rules.

- Tool NF Mediation Download Tool can be used to download the rule config map in a folder (folder name will be configmap name). It needs namespace as well as required configmap name. These rules then can be changed accordingly.
- Tool NF Mediation Upload Tool can be used to upload the rule config map from the existing config map folder. It needs namespace as well as required configmap name to be uploaded.
- If rules changed on NF active mediation then use **"<release-name>-nf-mediation-config-active**" as the name of the configmap.
- If rules changed on NF test mediation then use **"<release-name>-nf-mediation-config-test**" as the name of the configmap.

Configure SEPP using rules

Following is the procedure to configure SEPP using rules:

1. Login to the node where you want to update the SEPP config map.
2. Execute rules download script to download the existing config map. Refer to NF Mediation Download Toolfor download script.
3. Edit the config map with new values.
4. Execute rules upload script to upload the new config map. Refer to NF Mediation Upload Tool for upload script. The modifcation in the new rules can be in the headers or the body of the request or response sent to mediation.

NF Mediation Download Tool

Following is the rule for downloading NF mediation:

```
#!/bin/sh

read -p "Enter a valid Namespace name: " in_namespace
read -p "Enter a valid rules ConfigMap Name: " in_config

if ( [ -z "$in_namespace" ] || [ -z "$in_config" ] )
then
      echo "Namespace/ConfigMap can not be empty"
else
     #default_namespace="seppsvc"
    #namespace=${in_namespace:-$default_namespace}

    mkdir -p $in_config
    cd $in_config

    kubectl get cm $in_config -n $in_namespace -o json > rule_file
```

```
     for i in $(jq -r '.data | keys | .[]' rule_file);
     do
         j=$(echo $i | sed 's/\./\\./g')
         var="{.data."$j"}"
         kubectl get cm $in_config -n $in_namespace -o jsonpath=$var > $i
     done
     rm -rf rule_file

     cd ../../../
fi
```

NF Mediation Upload Tool

Following is the rule for uploading the NF mediation:

```
#!/bin/sh

read -p "Enter a valid Namespace name: " in_namespace
read -p "Enter a valid rules ConfigMap Name: " in_config

if ( [ -z "$in_namespace" ] || [ -z "$in_config" ] )
then
      echo "Namespace/ConfigMap can not be empty"
else
     #default_namespace="seppsvc"
    #namespace=${in_namespace:-$default_namespace}

    #mkdir -p $in_config
    #cd $in_config

    folder_name="$in_config/"
    #echo $folder_name
    kubectl create configmap $in_config -n $in_namespace  --dry-run -o
yaml --from-file=$folder_name| kubectl replace -f -
fi
```

NF-Mediation Rule Configuration

Following are the sample mediation rules.

Roaming Partner

Following is the sample rule to configure roaming partner:

```
rule "Roaming-partner-1"
when
    req :
Request(body.get("$.dataToIntegrityProtectBlock.requestLine.path") ==
"nnrf-disc/v1/nf-instances")
then
   req.headers.add("x-destination-uri", "http://psepp-stub-service.default:
```

```
8084/sepp/ipx/testing/server")
end
```

Modification operation

Following is the sample rule to add Modification Oparation:

```
rule "Modification-policy"
when
    rsp : Response(body.has("$..metaData.authorizedIpxId", 
"ipx.oracle.com") && body.has("$..statusLine","HTTP/2 200 OK"))
then
    String iePath1 = rsp.body.absPath("$..payload","iePath", "/
nfInstances/0/fqdn")
    String iePath2 = rsp.body.absPath("$..payload","iePath", "/
nfInstances/0/ausfInfo/supiRanges/0/start")
    rsp.body.put("$.modifications","operations",operation("ADD", 
"dataToIntegrityProtectBlock/payload", null, "['/validity','BODY','400']"))
    rsp.body.put("$.modifications","operations",operation("REMOVE", 
"dataToIntegrityProtectBlock"+iePath1,null, null))
    rsp.body.add("$.modifications.operations",operation("REPLACE", 
"dataToIntegrityProtectBlock"+iePath2,null, "20"))
end
```

Common functions

Following is the sample rule to add the mediation rules:

```
function Map<Object, Object> addObject() {
        return new HashMap<Object, Object>();
}

function ArrayList<Object> addArray() {
        return new  ArrayList<Object>();
}

function Map<String,Object> operation(String op,
                                                  String path,
                                                  String from,
                                                  Object value) {
        Map< String,Object> operationObj = new HashMap< String,Object>();
        operationObj.put("op", op);
        operationObj.put("path", path);
        operationObj.put("from", from);
        operationObj.put("value", value);
        return operationObj;
}
```

Common rules

Following is a common rule sample file.

```
rule "Ipx-Add-Modification-block-If-Not-Found"
      salience 3
```

```
when
    req : Request(body.has("$.modifications") == false)
then
        req.body.put("$", "modifications", addObject())
        req.body.put("$.modifications", "operations", addArray())
end


rule "Ipx-Add-Operations-Array-If-Not-Found"
      salience 4
when
    req : Request(body.has("$.modifications.operations") == false)
then
        req.body.put("$.modifications", "operations", addArray())
end

rule "Ipx-Add-Modification-block-If-Not-Found-Rsp"
      salience 5
when
    rsp : Response(body.has("$.modifications") == false)
then
        rsp.body.put("$", "modifications", addObject())
        rsp.body.put("$.modifications", "operations", addArray())
end

rule "Ipx-Add-Operations-Array-If-Not-Found-Rsp"
      salience 6
when
    rsp : Response(body.has("$.modifications.operations") == false)
then
        rsp.body.put("$.modifications", "operations", addArray())
end
```

Custom rules

Following is the custom rule sample file.

```
rule "IPX-Sample Rule Behave1"
when
    rsp : Response(body.has("$..metaData.authorizedIpxId",
"ipx.oracle.com") && body.has("$..statusLine","HTTP/2 200 OK"))
then
    String iePath1 = rsp.body.absPath("$..payload","iePath", "/
nfInstances/0/fqdn")
    String iePath2 = rsp.body.absPath("$..payload","iePath", "/
nfInstances/0/ausfInfo/supiRanges/0/start")
    rsp.body.put("$.modifications","operations",operation("ADD",
"dataToIntegrityProtectBlock/payload", null, "['/validity','BODY','400']"))
    rsp.body.put("$.modifications","operations",operation("REMOVE",
"dataToIntegrityProtectBlock"+iePath1,null, null))
    rsp.body.add("$.modifications.operations",operation("REPLACE",
"dataToIntegrityProtectBlock"+iePath2,null, "20"))
end

rule "IPX-Sample Rule Behave2"
```

```
when
    rsp : Response(body.has("$..metaData.authorizedIpxId",
"ipx.oracle.com") && body.has("$..statusLine","HTTP/2 200 OK") &&
body.has("$..headers[*].header","content-type"))
then
    String iePath = rsp.body.absPath("$..headers","header", "content-type")
    rsp.body.add("$.modifications.operations",operation("ADD",
"dataToIntegrityProtectBlock/headers", null, "['Content-
type','charset=UTF-8']"))
    rsp.body.add("$.modifications.operations",operation("ADD",
"dataToIntegrityProtectBlock/headers",null, "['Content-type','text/
plain']"))
    rsp.body.add("$.modifications.operations",operation("REMOVE",
"dataToIntegrityProtectBlock"+iePath,null, null))
end

rule "IPX-Sample Rule Behave4"
when
    rsp : Response(body.has("$..metaData.authorizedIpxId",
"ipx.oracle.com") && body.has("$..statusLine","HTTP/2 200 OK") &&
body.has("$..payload[*].iePath","/nfInstances/0/ausfInfo/supiRanges/0/
end"))
then
    String iePath = rsp.body.absPath("$..payload","iePath", "/
nfInstances/0/ausfInfo/supiRanges/0/end")
    rsp.body.add("$.modifications.operations",operation("REMOVE",
"dataToIntegrityProtectBlock"+iePath, null, null))
end
```

3

Metrics and KPIs

Metrics

The following table provides information for Metrics for SEPP.

Table 3-1 Metrics

Metrics Details	SEPP microservice	Metrics
Number of total requests which is received to JwsSvc	JwsSvc	sepp_jwssvc_rx_total
Number of successful responses sent from JwsSvc	JwsSvc	sepp_jwssvc_rsp_success
Number of failure responses sent from JwsSvc	JwsSvc	sepp_jwssvc_rsp_failure
Number of total requests received by N32fIpx	N32fIpx	sepp_n32fipx_rx_total
Number of total requests getting successfully implemented	N32fIpx	sepp_n32fipx_rx_success
Number of total requests getting failed while during implementation	N32fIpx	sepp_n32fipx_rx_failure
Number of total response received by N32fIpx	N32fIpx	sepp_n32fipx_rsp_total
Number of total response getting successfully implemented	N32fIpx	sepp_n32fipx_rsp_success
Number of total response getting failed while during implementation	N32fIpx	sepp_n32fipx_rsp_failure
Number of total requests/response from mediation successfully implemented	N32fIpx	sepp_n32fipx_med_success
Number of total requests/response from mediation failed during implementation	N32fIpx	sepp_n32fipx_med_failure
Number of total requests/response from jws successfully implemented	N32fIpx	sepp_n32fipx_jws_success
Number of total requests/response from jws failed during implementation	N32fIpx	sepp_n32fipx_jws_failure

> **Note:**
>
> Currently, the dashboard must be created manually.

KPIs

The following table provides information for KPIs for SEPP.

Table 3-2 KPIs

KPI Details	Microservice	Service Operation	Response Code	Notes
Requests/sec	All	All	Note Applicable	KPI templates are not made as deliverable yet.