IPsec Certificate-based Authentication

Design review

Xu Liu <xliu2@vmware.com>

Motivation

- The current Antrea IPSec support can only use preshared shared key (PSK) authentication with static manually created keys. This is too limited for serious use in enterprise networks.
- OVS supports authenticating tunnel endpoints using x509 version 3 certificates. Antrea only needs to manage the certificates and load them to each Node. OVS toolkits will be responsible for monitoring and configuring the IKE daemon.

Certificate format

CA based self-signed certificates for Nodes

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            4f:fb:fe:f4:c2:e5:52:b0:9a:c5:2c:32:05:26:66:1c:8e:a0
        Signature Algorithm: sha512WithRSAEncryption
        Issuer: CN = antrea-ipsec-ca
        Validity
            Not Before: Apr 26 00:24:38 2022 GMT
            Not After : Apr 23 00:24:39 2032 GMT
        Subject: O = antrea.io, CN = k8s-node-control-plane
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                RSA Public-Key: (2048 bit)
                Modulus: <REDACTED>
                Exponent: 65537 (0x10001)
        X509v3 extensions:
            X509v3 Extended Key Usage:
                IPSec Tunnel
            X509v3 Basic Constraints: critical
                CA:FALSE
            X509v3 Authority Key Identifier: <REDACTED>
            X509v3 Subject Alternative Name:
                DNS:k8s-node-control-plane
    Signature Algorithm: sha512WithRSAEncryption
       <REDACTED>
```

OVS configurations

Certificate configurations

```
# ovs-vsctl set Open_vSwitch . \
other_config:certificate=/etc/ipsec.d/certs/k8s-node-control-plane-cert.pem \
other_config:private_key=/etc/ipsec.d/private/k8s-node-control-plane-privkey.pem \
other_config:ca_cert=/etc/ipsec.d/cacerts/cacert.pem
```

Tunnel configurations

```
# ovs-vsctl show
9fef812b-d2f1-477a-bffb-d7f492bb42f9
    Bridge br-int
        datapath_type: system
        Port worker-1-ac11df
            Interface worker-1-ac11df
                type: gre
                options: {remote_ip="192.168.77.101", remote_name=k8s-node-worker-1}
        Port worker-2-4a2272
            Interface worker-2-4a2272
                type: gre
                options: {remote_ip="192.168.77.102", remote_name=k8s-node-worker-2}
        Port antrea-gw0
            Interface antrea-gw0
                type: internal
        Port antrea-tun0
            Interface antrea-tun0
                type: gre
                options: {key=flow, remote_ip=flow}
    ovs_version: "2.15.1"
```

Certificate management

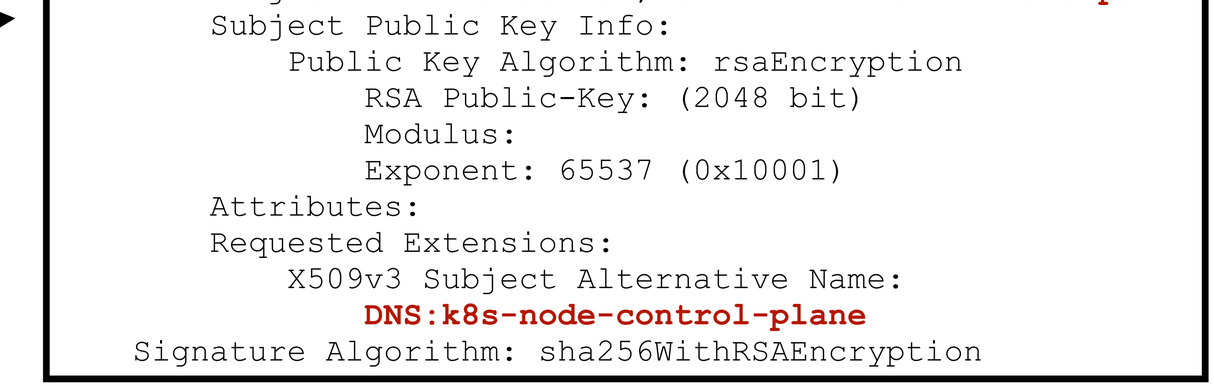
Request and issue certificates

- Antrea-agent Pods cannot mount their own certificates individually easily as they are managed by Daemonset. It is not secure to store all the issued certificates and private keys in one Secret.
- It is trivial to issue new certificates without persisting them in Kubernetes as, in most cases, they can be self-signed.
- Kubernetes > v1.19 provides stable CertificateSigningRequest APIs, which fit nicely with the controller and agent pattern of Antrea.
- Upon first running, antrea-controller can generate a self-signed root certificate and save the certificate and its private key as a Secret. Meanwhile, it will also save the certificate in a Configmap so that antrea-agents can mount or read it.

Certificate management

Integrate with Kubernetes CSR API

```
apiVersion: certificates.k8s.io/v1
kind: CertificateSigningRequest
metadata:
 name: k8s-node-control-plane-ipsec
spec:
  request: <PEM encoded CSR>
  signerName: antrea.io/signer
 usages:
  - ipsec tunnel
status:
  certificate: <signed certificate>
  conditions:
  - message: Automatically approved by antrea.io/
signer
    reason: AutoApproved
    status: "True"
    type: Approved
```



* Approved/denied/failed requests will be automatically deleted after 1 hour by the garbage collector of Kubernetes

Certificate management

RBAC

```
antrea-controller:
- apiGroups:
  - certificates.k8s.io
  resources:
  - certificatesigningrequests
  verbs:
  - get
  - list
  - watch
- apiGroups:
  - certificates.k8s.io
  resources:
  - certificatesigningrequests/approval
  - certificatesigningrequests/status
  verbs:
  - update
- apiGroups:
  - certificates.k8s.io
  resources:
  - signers
  resourceNames:
  - antrea.io/signer
  verbs:
  - approve
  - sign
```

```
antrea-controller:
- apiGroups:
    - ""
  resources:
    - configmaps
    - secrets
  resourceNames:
    - antrea-ipsec-ca
  verbs:
    - get
    - update
- apiGroups:
    - ""
  resources:
    - configmaps
    - secrets
  verbs:
    - create
```

```
antrea-agent:
- apiGroups:
  - certificates.k8s.io
  resources:
    - certificatesigningrequests
  verbs:
    - get
    - watch
    - list
    - update
    - patch
    - create
```

Certificate management

Certificate renewal

- The signed certificate is about to expire.
- Node reboots (If we store the private key and signed certificates to `/var/run/antrea` on the Node).
- Root certificate changed. (Root certificate expired or the Secret/Configmap is deleted)

* Currently, the script `ovs-monitor-ipsec` watches on OVS databases changes instead of the file content changes of certificates. For certificates reloading, we can choose the following for a workaround.

1. Generate a random file name or suffix for each signed certificate and update the `other_configs` section in OVS DB.
2. Use static names for certificate files and add another field to `other_configs`. e.g, other_config:**certificate_hash**.

Questions

- Does the `antrea-ipsec` container need to be responsible for requesting CSRs and managing certificates? If not, it can be handled in `antrea-agent` container by mounting the same folders on the Node.
- Other questions?