TLS Origination Best Practices

Kenan O'Neal ASPEN MESH

IstioCon

#IstioCon

whoami

- Istio open source contributor for close to a year
- Current release manager for Istio 1.12
- Worked on introducing VERIFY_CERTIFICATE_AT_CLIENT environment variable

TLS Origination Security

Istio is designed to be flexible, but at times this comes at a cost. One of these costs happens when configuring a DestinationRule.

Things that aren’t done by default:

1. Checking that the certificate is signed by a CA that your system trusts
2. The certificate’s SAN DNS name of the host is not verified
3. The SNI isn’t sent to the intended server

CA (Certificate Authority) Certificate Bundle

- Specify CA certificate bundle to use in validating certificates
- Most Operating Systems have a built-in CA certificate bundle

```
spec:
  host: self-signed.badssl.com
  trafficPolicy:
    tls:
      mode: SIMPLE
      caCertificates: /etc/ssl/certs/ca-certificates.crt
```

SAN (Subject Alternative Name)

- Hostname that was issued the certificate
- Relies on CA certificate being specified

```
spec:
  host: wrong.host.badssl.com
  trafficPolicy:
    tls:
      mode: SIMPLE
      caCertificates: /etc/ssl/certs/ca-certificates.crt
      subjectAltNames:
      - "wrong.host.badssl.com"
```

SNI (Server Name Indication)

- Hostname the service is requesting to talk to

```
spec:
  host: self-signed.badssl.com
  trafficPolicy:
    tls:
      mode: SIMPLE
      caCertificates: /etc/ssl/certs/ca-certificates.crt
      subjectAltNames:
      - "self-signed.badssl.com"
      sni: self-signed.badssl.com
```

ServiceEntry Behavior

Important things to note:

- A ServiceEntry can also specify SANs (Subject Alternative Name)
- A SAN in a DestinationRule overwrites all ServiceEntry SANs
- In order for SANs to be verified against the host, a certificate must be specified in the DestinationRule

Verify Certificate At Client

- Automatically set CA certificate bundles
- System is checked for the first certificate bundle detected from the list below
- Can be overridden by explicitly setting the CA certificate
- Istio version > 1.12

```
"/etc/ssl/certs/ca-certificates.crt",                // Debian/Ubuntu/Gentoo etc.
"/etc/pki/tls/certs/ca-bundle.crt",                  // Fedora/RHEL 6
"/etc/ssl/ca-bundle.pem",                            // OpenSUSE
"/etc/pki/tls/cacert.pem",                           // OpenELEC
"/etc/pki/ca-trust/extracted/pem/tls-ca-bundle.pem", // CentOS/RHEL 7
"/etc/ssl/cert.pem",                                 // Alpine Linux
"/usr/local/etc/ssl/cert.pem",                       // FreeBSD
```

Verify Certificate At Client

Only works in SIMPLE or MUTUAL TLS modes, ***NOT*** in ISTIO_MUTUAL

https://istio.io/latest/docs/reference/commands/pilot-discovery/#envvars

```
spec:
  host: self-signed.badssl.com
  trafficPolicy:
    tls:
      mode: SIMPLE
      subjectAltNames:
      - "self-signed.badssl.com"
      sni: self-signed.badssl.com
```

Setting “Verify Certificate At Client”

- Helm yaml file for istiod needs VERIFY_CERTIFICATE_AT_CLIENT = true

```
pilot:
  ...
  env:
    VERIFY_CERTIFICATE_AT_CLIENT: “true”
```

Testing

- Use a host that you trust but serves bad certificates (badssl.com)
  - Untrusted root certificate
  - Self-signed certificate
  - Certificate for a different host

Work Currently Being Done

This is not easy enough for the user. Faseela K. is working to improve VERIFY_CERTIFICATE_AT_CLIENT to include auto_san and auto_sni.

If you would like to see more changes like these to improve security and ease of use for users, I’m sure Istio is happy to have more contributions.

Thank you!

@Kenan O’Neal (Aspen Mesh)
https://aspenmesh.io

#IstioCon