Injecting Security into the Cloud

Susan Hinrichs
shinrich@verizonmedia.com
Verizon Media

verizon media

ONES 2020

About me

Work with Edge Team at Verizon Media

- **Support Apache Traffic Server. Committer, PMC member**
- **Work on web proxies, routing, networking tools**

Practice and teach yoga in my spare time

Reach me at

- **shinrich@verizon.media, shinrich@apache.org**
- **https://www.linkedin.com/in/shinrich/**

verizon media

Goal of Talk

Present TLS scenarios we encountered while adapting services to hybrid cloud scenarios

Describe how we use a proxy (Apache Traffic Server) and an access control system (Athenz) to secure new traffic patterns

- **Proxy supporting Application Delivery Network (ADN)**

Work performed with my colleagues at Verizon Media and members of the Apache Traffic Server and Athenz communities

- **Persia Aziz, Zeyuan Yu, Alan Carroll, Vinith Bindiganavale, and Henry Avetisyan**

verizon media

Apache Traffic Server: Web Cache and Proxy

Apache Traffic Server (ATS) has been used for many years (open source for over 10 years). Primarily for the traditional web serving model.

> **Allows injection of business specific logic via "plugins" triggering on hooks during the lifecycle of a HTTP transaction and session.**

We have spent the past three years enhancing TLS configuration to better support our evolving requirements to support hybrid cloud and fine-grained mutual TLS based authentication

https://trafficserver.apache.org/

Athenz: a Role Based Access Control (RBAC) System

Athenz is a open source RBAC system that leverages TLS mutual authentication

Contributed to open source by Yahoo/Verizon Media

Athenz has been the basis for much of our Verizon Media authentication and authorization work

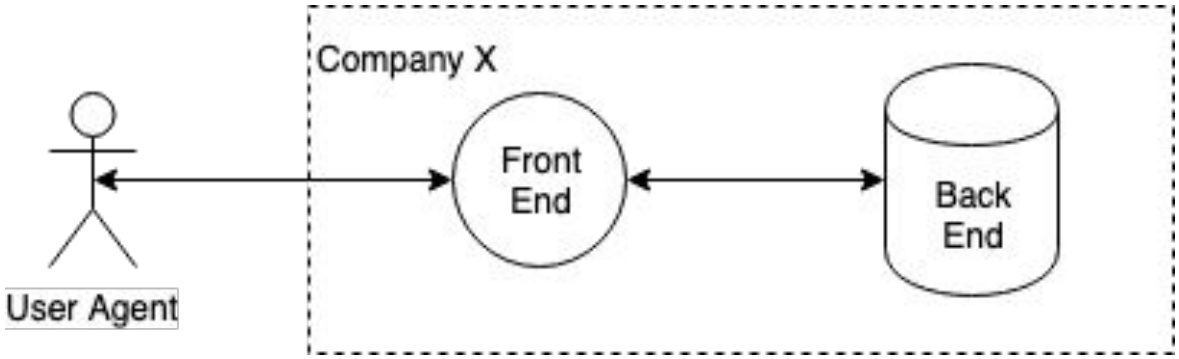
- https://github.com/yahoo/athenz

Motivation

verizon media

The Old Days, Corporate Data Centers

Dedicated resources to a single organization

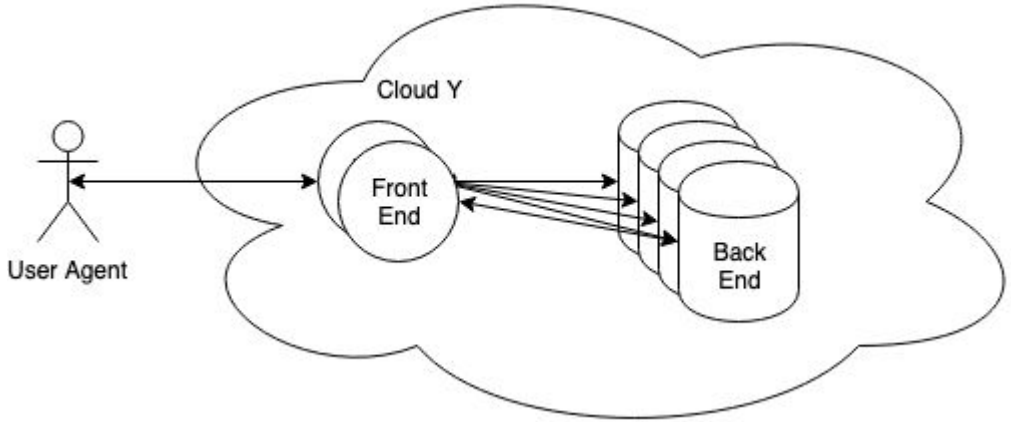


Recent Past, Public Cloud

Each organization rents virtual resources from a 3rd party cloud service to run company apps

No employee investment in machine care and feeding. Quick to scale up and down.

But expensive for large, dedicated resource usage

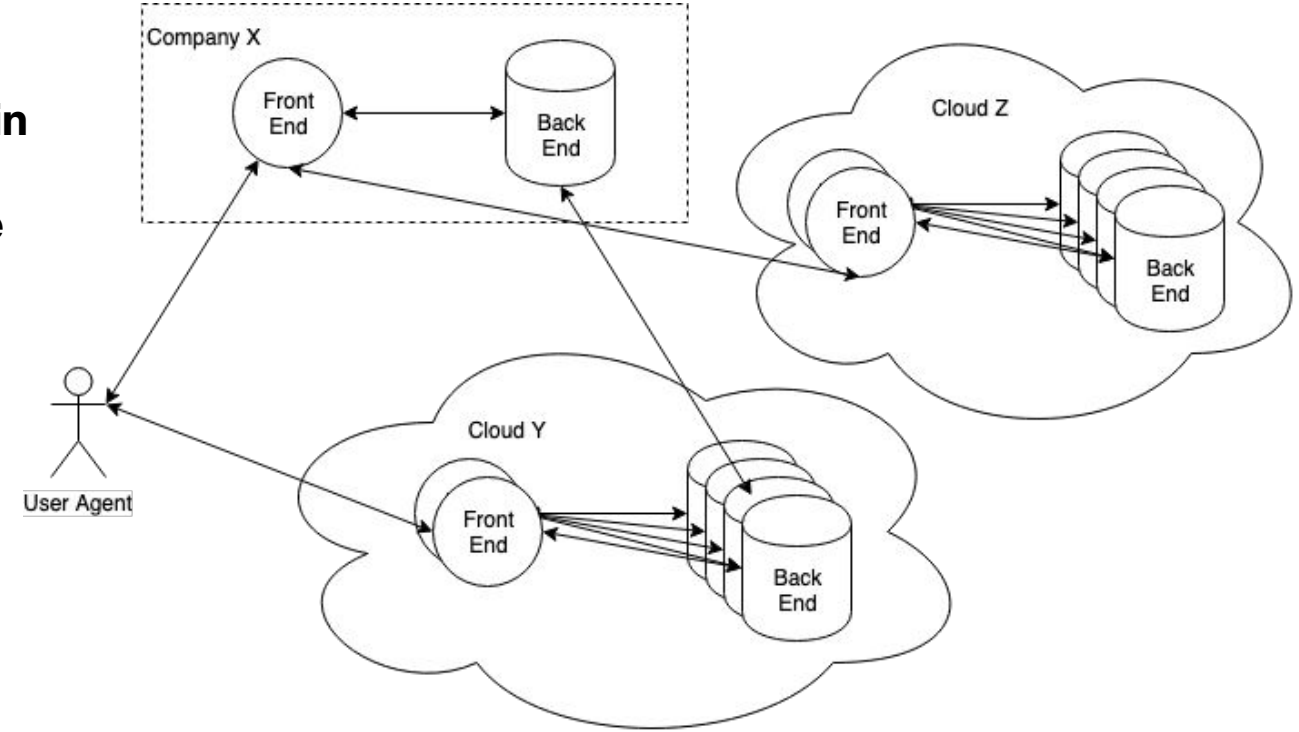


verizon media

Now, Hybrid Cloud

Run applications in combination of public and private resources

Dynamic and complex trust relationships



verizon media

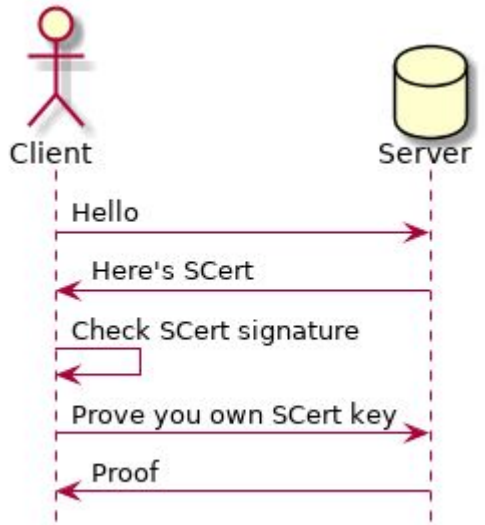
Injecting TLS Authentication

verizon media

Classic Client-Server TLS Authentication Model

Only server proves identity via certificate

Client authentication (if any) is through mechanism independent of TLS, e.g. password or 2FA



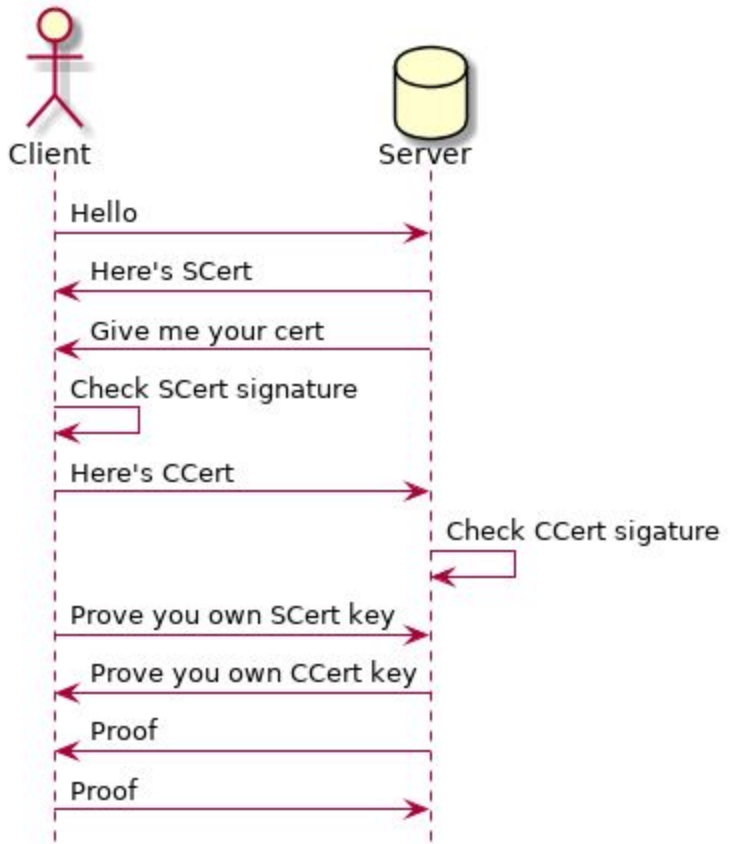
verizon media

Mutual TLS Authentication Model

Client provides certificate too

Both sides verify the other's certificate

Athenz provides a Role Based Access Control (RBAC) system on top of mutual TLS authentication

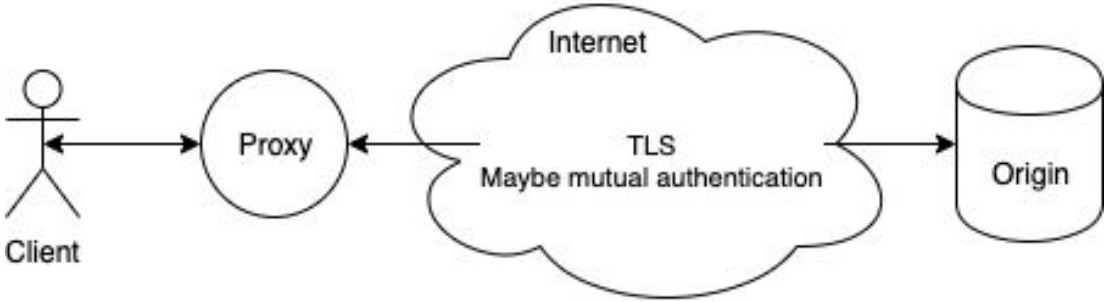


Retrofitting Clients for TLS

Process must connect to service using TLS

Client is not capable of TLS or maybe not capable of mutual TLS

- **Put a proxy like ATS in front of the client**
- **Configure ATS to provide the same certificate for all outgoing service requests.**
- **What if client only should mutual TLS to some services?**



verizon media

Fine Grained Client Certificate Selection

Via conf_remap.so in remap.config

Specify overrides for proxy.config.ssl.client.cert.filename

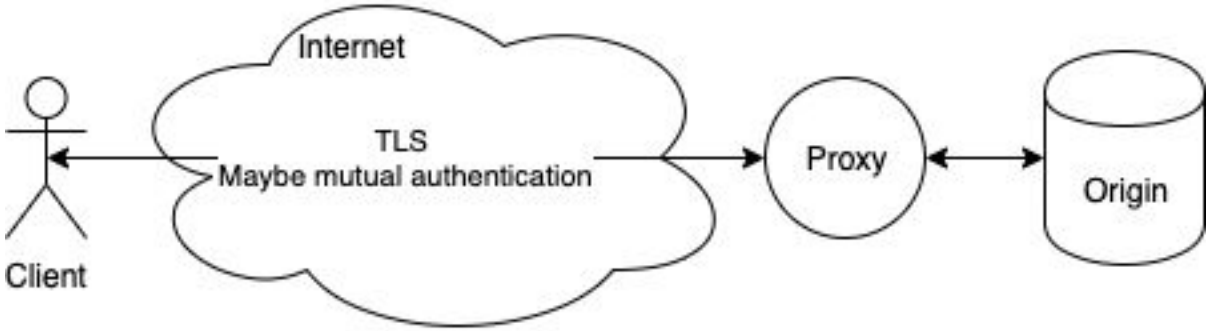
```
map https://bank.yahoo.com/pci https://pci.server.com/
@plugin=conf_remap.so
@param=proxy.config.ssl.client.cert.filename=pci.pem

map https://yahoo.com/datastore
https://datastore.server.com/ @plugin=conf_remap.so
@param=proxy.config.ssl.client.cert.filename=datastore.pem
```

Retrofitting Server for TLS

Server must provide a TLS interface for clients

Server is not capable of TLS or maybe not capable of requiring certificates from clients or doing adequate certificate verification

- **Put a proxy like ATS in front of the server**
- **Configure ATS to require certificates from all clients**
- **What if only some client requests require mutual TLS?**



Using SNI to Drive TLS requirements

In Client Hello, user agent can set Server Name Indication (SNI) in the TLS Extensions

- **Modern clients do this**

Server can adjust requirements for TLS handshake based on SNI value

- **Which HTTP protocols to offer via ALPN**
- **Which TLS protocols to allow**
- **Which ciphers to offer**
- **Whether to require a client certificate**

SNI and Misbehaving Clients

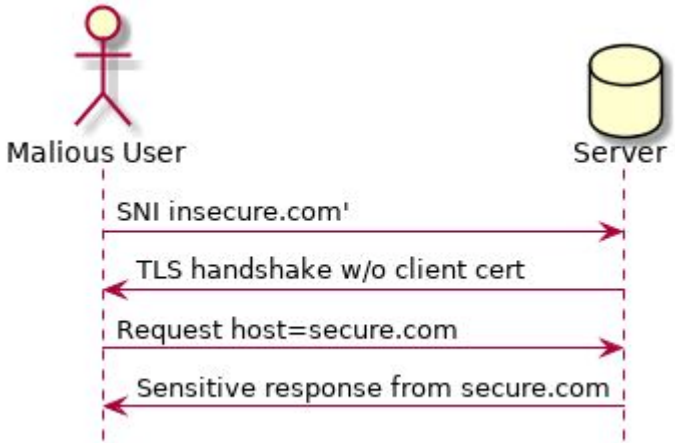
verizon media

What If User Agent Does Not Play By The Rules?

When making a request we assume at least the first request on the TLS connection SNI == Host header value. But this is not required.

```
curl -H 'host: secure.com' https://insecure.com/sensitive-data
```

SNI == insecure.com,



verizon media

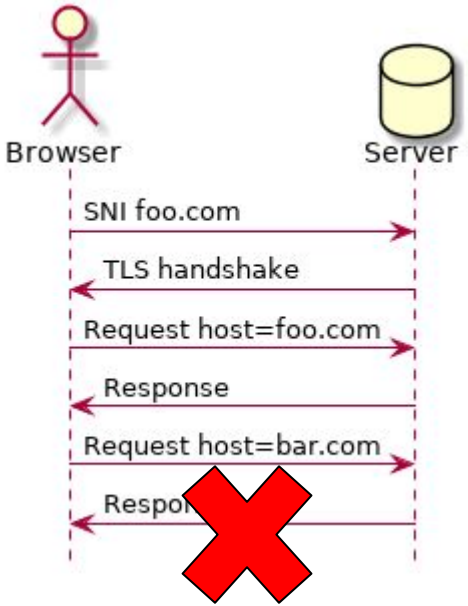
Cannot solely rely on SNI for security policy decisions

Option 1: Dedicated proxies that always require client cert.

- **May be other SNI-tied actions that need to be addressed**

Option 2: Fail any transaction where Host header value != connection's SNI

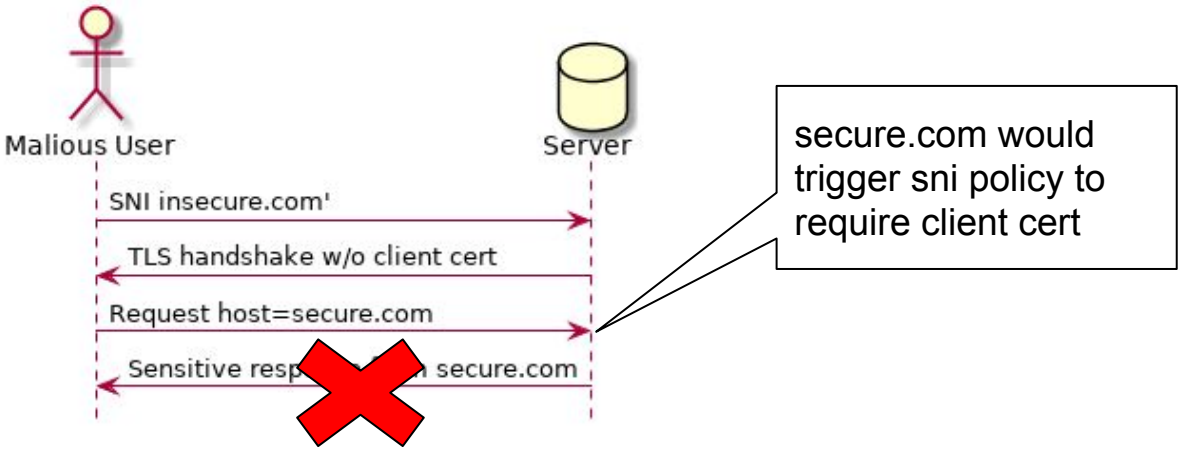
- **Too onerous. If the SNI is not used for policy selection, doesn't matter. With browser connection/session reuse name differences may be innocuous.**



ATS Solution

In transaction processing, use Host header value and see if that value would have selected a security-sensitive SNI policy.

If so, fail transaction (return 403) if SNI != Host header value



Other Options To Address Host/SNI Mismatch

A draft IETF RFC

Secondary Certificate Authentication in HTTP/2

Rather than failing the transaction, attempt to request additional client certificates on the fly. More friendly and better able to use existing connections.

- **Supporting implementations exist for HTTP/1.1**
- **Need updated implementation for HTTP/2**
- **We're keeping an eye on this work and may move there in the future**

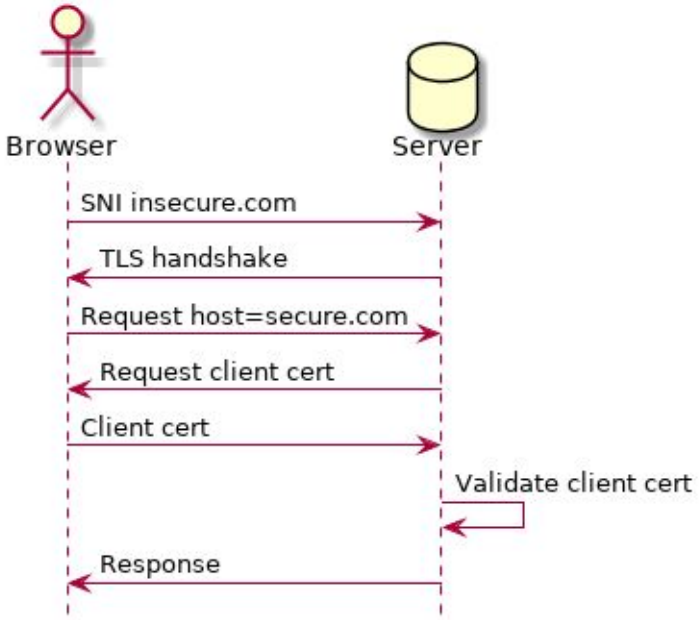
Renegotiation for client certificates

HTTP/1.x requests are made in series over a connection

- **Can incrementally add client certificates via TLS renegotiation**

HTTP/2 many requests can be operating in parallel

- **Cannot just update the “current” client certificate for the underlying TLS connection**
- **Must add HTTP/2 level frames**



Secure Edge: Proxy TLS

verizon media

Proxying TLS via the Secure Edge

Much like retrofitting TLS Server. However, motivation is addressing client in one cloud and server in another cloud (e.g. hybrid cloud case).

Don’t want to expose cloud servers directly.

- **May not trust security of client/server implementations**
- **May find it more secure to enforce security standards on a fixed set of Edge machines.**

Three options

- **TLS Delegation**
- **Tunnel**
- **Bridge**

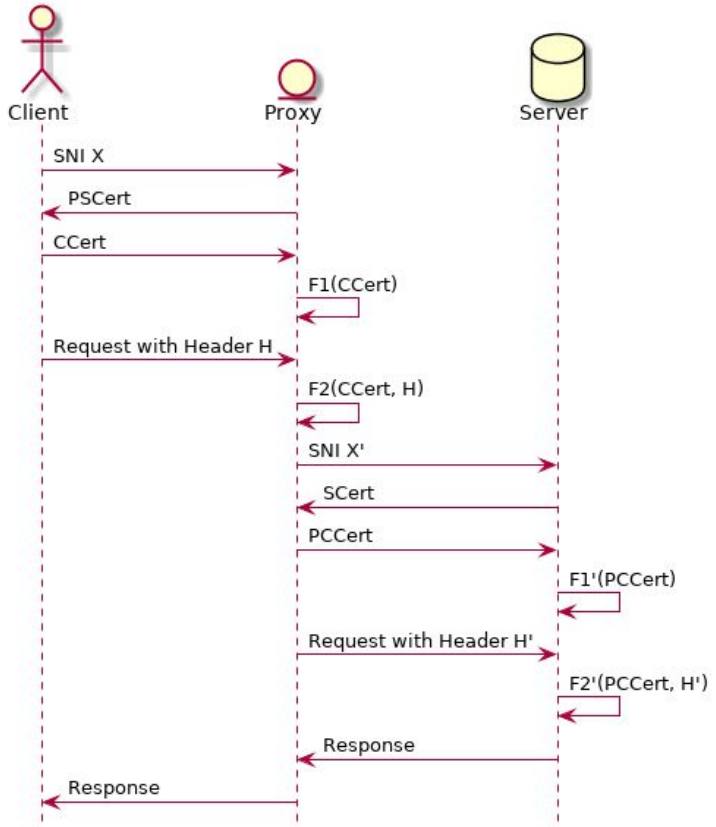
verizon media

TLS Delegation - General Case

Proxy validates CCert and header H

Proxy presents PCCert and header H'

- **CCert cannot be used for TLS to Server**
- **H' is related to H**



verizon media

Athenz Authentication and Authorization

Athenz builds on top of mutual TLS authentication

- **Client provides a x509 client cert signed by the Athenz CA**
- **The valid client cert authenticates an Athenz principle**

Athenz token expresses authorization

- **Athenz OAuth2 access token identifies the principle and roles that principle is authorized for**
- **Token has limited lifetime and is signed**
- **Passed in header of client request**

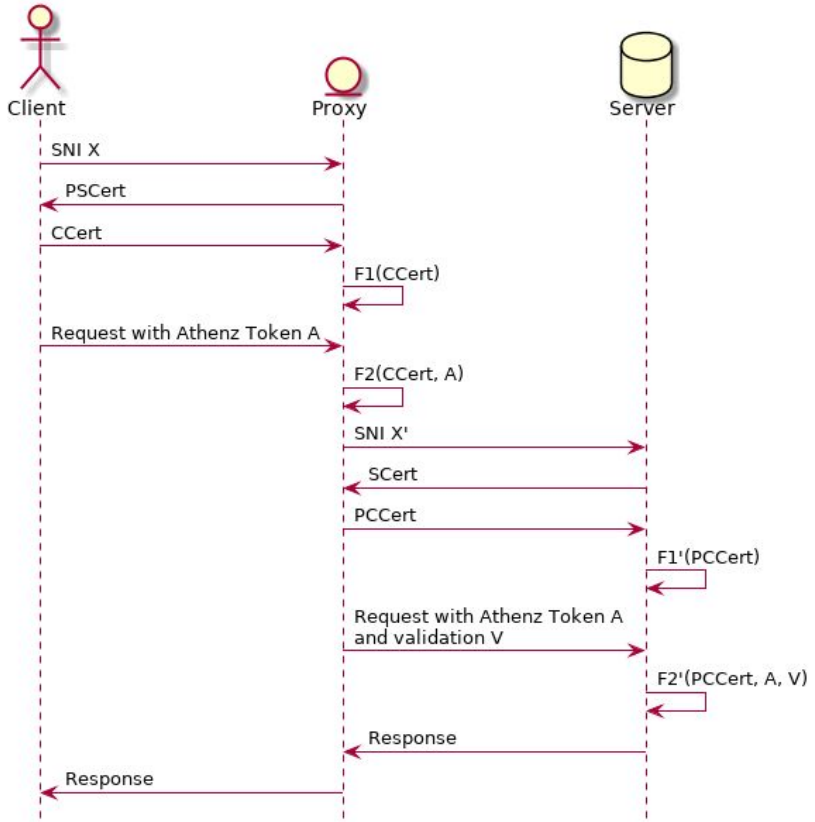
TLS Delegation - Athenz Case

Proxy validates CCert and Athenz token A

- **Validates that token A is signed and bound to cert identity**

Proxy presents PCCert and forwards Athenz token A along with validation header V

Must protect against malicious user presenting bogus A and V directly to Server



verizon media

Attacking TLS Delegation

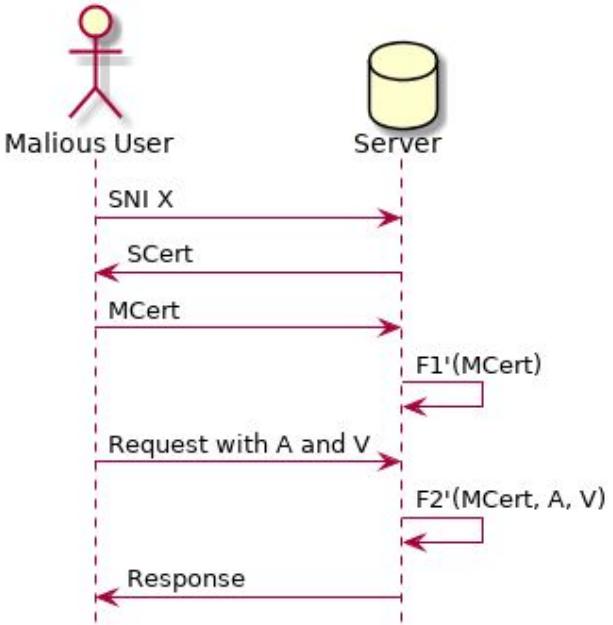
Request header A and V only checked if initial mutual TLS passes

A is signed but could be stolen

If Malicious User can present and verify a certificate trusted by Server, all is lost.

Buying a cert signed by DigiCert or Symantec could be a problem

Server must trust a limited set of CA



TLS Tunnel

Proxy does not terminate TLS connection

Looks at SNI and decides where to forward the TLS bytes

- **Actual TLS termination is end-to-end**

Proxy just acts as a SNI-based router in this case

```
sni:
  -fqdn: supersensitive.example.com
   tunnel_route: supersensitive.origin.example.com
```

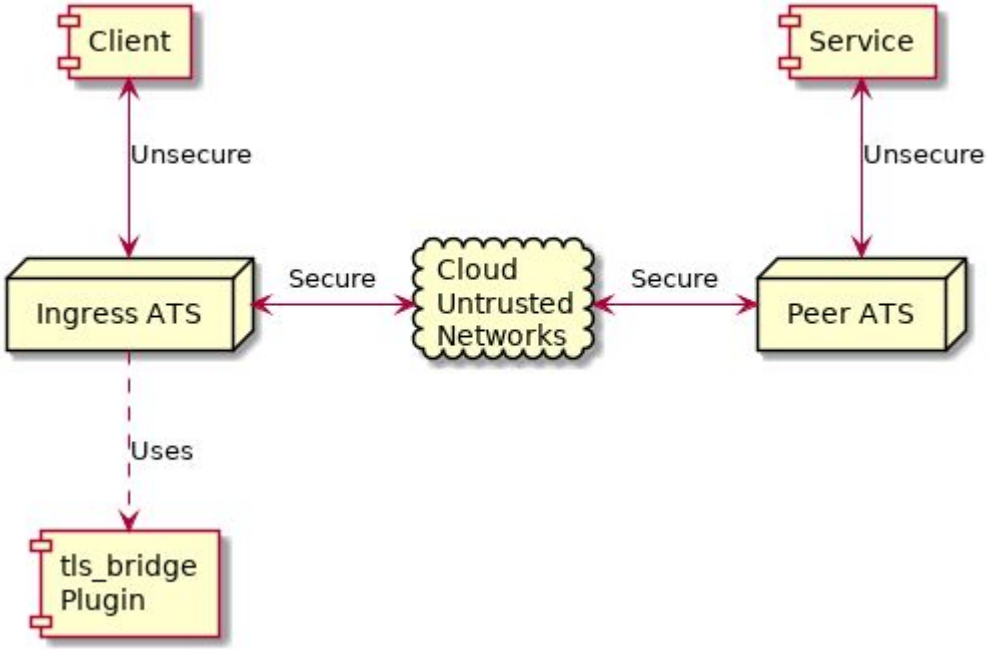
TLS Bridge

Avoid decryption on Proxy, but make some guarantees on strength of TLS connection independent of the user agent and/or server stack

Build a separate Tunnel between gateways. Double tunnel.

https://docs.trafficserver.apache.org/en/latest/developer-guide/plugins/example-plugins/tls_bridge.en.html

TLS Bridge



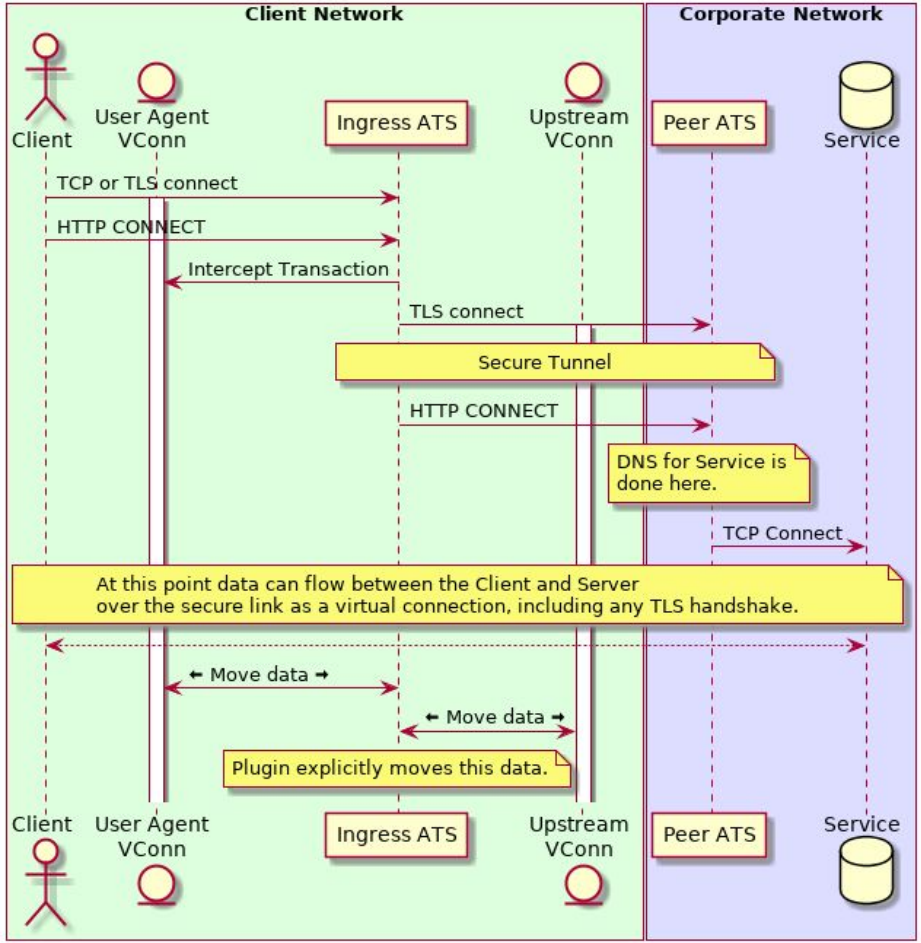
verizon media

TLS Bridge

Client connects to Ingress ATS

Proxy negotiates new TLS to Peer ATS

Proxies blind tunnel original traffic between client and service



verizon media

Wrapping It Up

The Secure Edge approach has been very instrumental into move to Hybrid Cloud

- **Proxy provides one place to ensure security policy is applied consistently**
- **All teams do not need to become mTLS experts**
- **Clients and servers can be migrated between clouds**

Q&A

shinrich@verizonmedia.com, shinrich@apache.org

Slides at https://github.com/shinrich/presos/blob/master/ones20.pdf

verizon media