TLS State of the Onion

Susan Hinrichs

Fall ATS Summit 2018

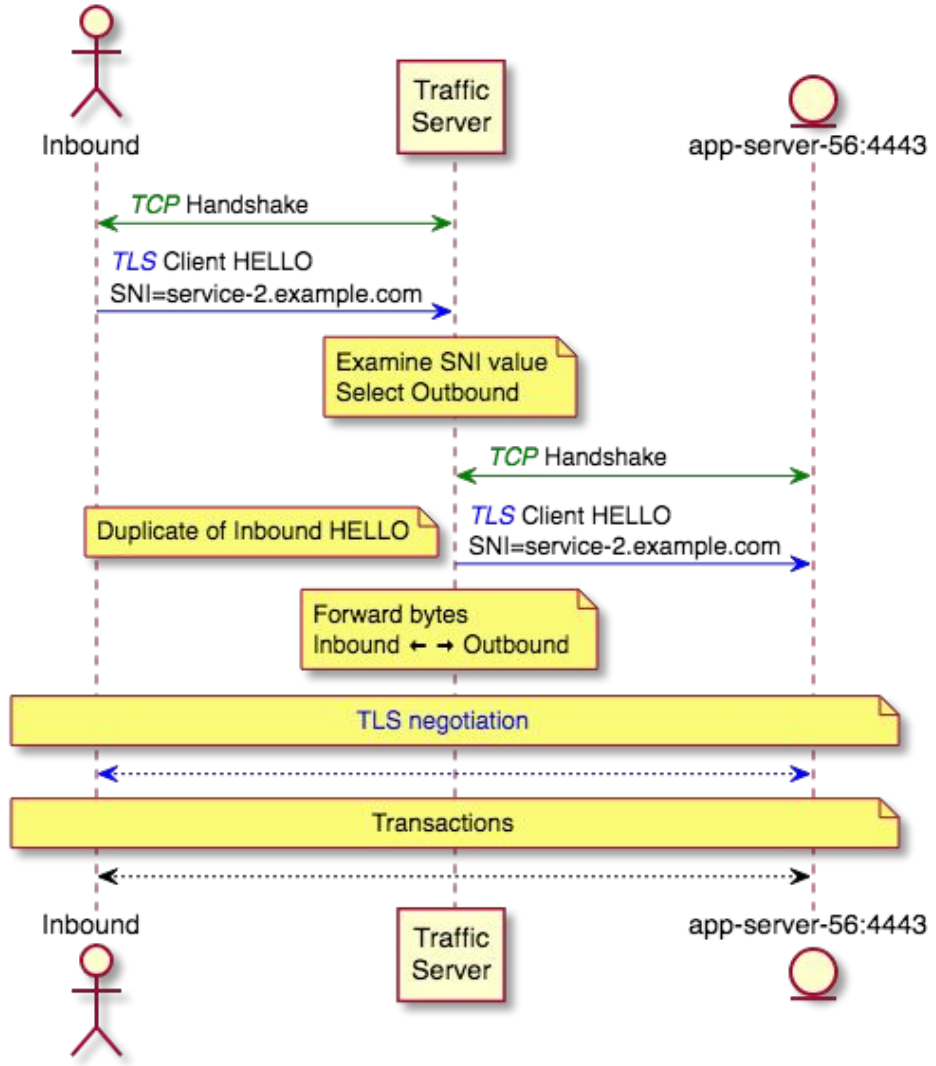
SNI/Layer 4 Routing

- A big motivator for a lot of Oath/Yahoo work in the TLS area this past year.
- Alan wrote some nice documentation for the use cases.
  - https://docs.trafficserver.apache.org/en/latest/admin-guide/layer-4-routing.en.html
- Using Traffic Server to route and securely tunnel traffic between regions
  - SNI-based control for mutual TLS
  - Tunnel to peer: Direct to origin or to another gateway (nested tunnels)
  - Do not decrypt on the Traffic Server
- Persia did most of the implementation

SNI Routing Example

- ```
  tunnel_route: app-server-29:443
    fqdn: service-1.example.com

  - tunnel_route: app-server-56:4443
    fqdn: service-2.example.com
  ```


  ```

SSL overrides - ssl_server_name.yaml

- Persia's work for over a year https://docs.trafficserver.apache.org/en/latest/admin-guide/files/ssl_server_name.yaml.en.html?highlight=ssl_server_name%20config
- Allows for TLS config overrides
  - On client side based on SNI
    - verify_client
    - disable_h2
    - tunnel_route
  - On server side based on host name (SNI to origin)
    - verify_origin_server
    - client_cert
- Considering adding overrides for the CA file (both sides)
- Can specify the FQDN exactly or prefix wildcard
- Can be constrained by IP list

Openssl 1.1.0/1.1.1

- Openssl 1.1.0/1.1.1 required numerous code tweaks
  - The community had been updating Traffic Server during pre-releases
- Some new features came with openssl 1.1.*
  - ASYNC_*_JOB (which I'll discuss more in a future slide)
  - Better performance scaling (eliminating the pesky lock in the error lookup code)
  - New ciphers, especially CHACHA20_POLY
- Traffic Server (and others) have problems with the first final version of openssl 1.1.1
  - Shared_sigalgs got reset when a new cert/CTX is set in the ssl_cert_callback. That value is never reset, so no matching signature algorithm is found and the handshake fails.
  - For Traffic Server only handshakes that use the default certificate would work.
  - Patch in place
    - https://github.com/openssl/openssl/issues/7244
  - Also new crashes on shutdown

Openssl 1.1.1 and TLSv1.3

- Masaori added a config to specify TLSv1.3 ciphers
  - proxy.config.ssl.server.TLSv1_3.cipher_suites
- Probably want to add a disable/enable config too
  - proxy.config.ssl.client.TLSv1_3
- Currently always on. A default set of ciphers in place if you don't change the configuration.
- Initial measurements
  - On pre-release 1.1.1 we were seeing around 0.7% of connections negotiating pre-release versions of TLSv1.3
  - On final 1.1.1 (with final TLSv1.3) the rate is much, much lower. Around 0.002%
    - At this point beta/alpha versions of Firefox and Chrome are negotiating TLSv1.3
    - Should improve as the installed base updates.

Next Steps for TLSv1.3

- You can compile against 1.1.0 and run against 1.1.1
  - The TLSv1.3 specific configs won't be present.
- Even compiling against 1.1.1 will not get you 0-RTT handshake
  - Client can send "early data" before the handshake completes
    - Concerns about replay attacks https://tools.ietf.org/html/draft-thomson-http-replay-01
  - Server decides
    - Can reject early data. Return 425 - too early
    - Can gather and not process data until handshake completes.
      - Deal with multiple early data packets?
- Relevant openssl calls
  - https://www.openssl.org/docs/man1.1.1/man3/SSL_get_max_early_data.html

TLS Dynamic Record Size

- Fixed error that would occasionally cause crashes if write had to be retried and block size changed. TS-4424,
  - Can now safely set proxy.config.ssl.max_record_size to -1
  - We haven’t rolled this out yet
  - Should improve TLS performance by adapting to optimal record size for the situation.

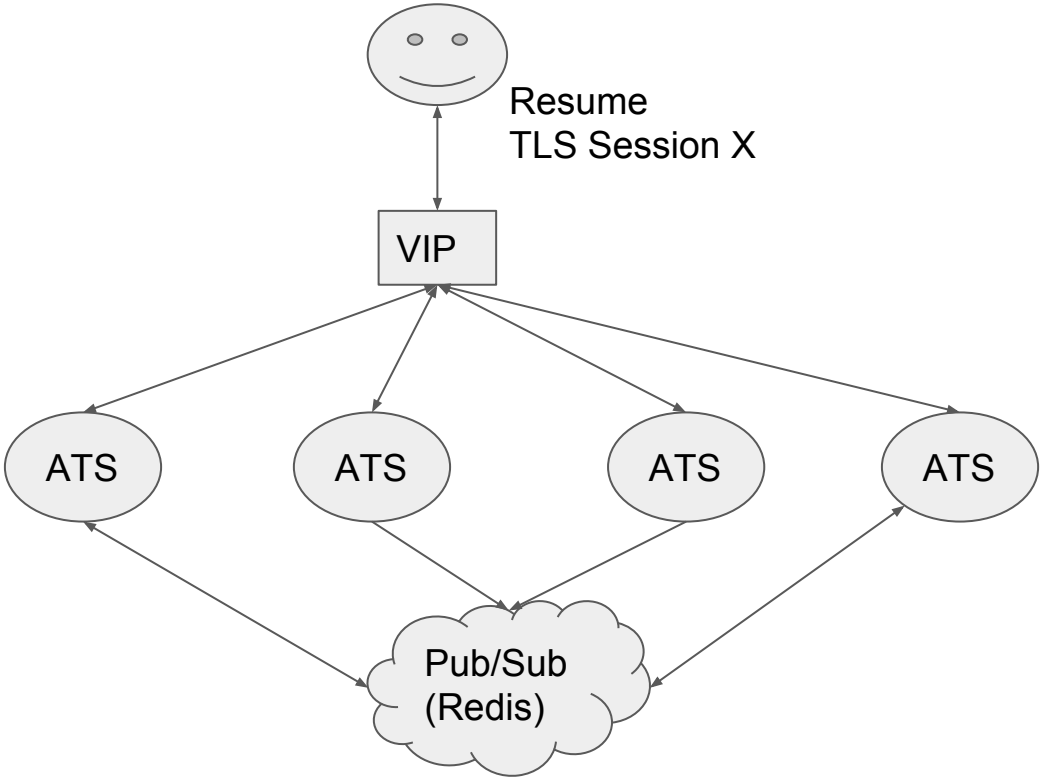
TLS Half Open Disable

- Recently added for Peter and Jeremy
  - They reported having a TLS client send a client-notify and FIN but ATS would continue sending data back.
  - This sounded like a bad side effect of the half-open logic in HttpSM. If the underlying protocol is just TCP it could be legitimate to sending back traffic after the client sends a FIN. The client may still be listening.
- It is not reasonable to continue reading over a TLS socket once the peer has shutdown
  - Augmented the allow_half_open() methods to always return false if the underlying network connection is TLS.

Certifier Plugin

- New plugin that supports SSL bump and dynamic cert loading
  - Zeyuan will discuss in detail

Session Reuse Plugin



SSL Session Reuse Plugin

- In PR #4125. This is a plugin that Yahoo/Oath has been using for 4 years to allow for TLS session sharing over a pod
  - Had to remove Yahoo support library, and add TLS Session APIs
- Uses Redis as a cross-pod pub/sub communication mechanism. When a server creates a new session, it shares it with its peers via Redis
  - If the session is restarted on another server in the pod, it should be known

SSL Session Reuse Plugin

- Also supports auto-updating tickets (STEK) and keeping them up to date.
  - Added by Dave Thompson
  - Protocol to identify master.
  - Master will periodically update STEK and publish to peers
  - Use TSSslTicketKeyUpdate to keep the last two ticket encryption keys active
- Eventually the benefit of session reuse will likely decrease
  - TLS 1.3 discourages session reuse
  - HTTP/2 inherently uses a connection for longer.
  - Seeing around 34% TLS session resumption (based on a really small sample set)
    - Around 22% ticket and 12% session ID
    - 97% of presented tickets are good. 40% of session IDs presented are good.

Crypto Proxy Rides Again!

- Integrated ASYNC_*_job support from openssl 1.1.0
- Will be open sourcing a repo that includes a reference Crypto-Proxy server and an initial openssl engine that works with Traffic Server
- Configure Traffic Server box to load openssl engine to take over RSA private key operations
  - Set up openssl.conf file to load engine and identify support files.
- Crypto Proxy listens for requests
  - TLS mutual auth. Client certs and IP addresses must be whitelisted
  - Keys identified by hash of public key
  - Only Crypto Proxy has the private key. Performs private key operations on behalf of trusted client.
- Currently just support RSA. Will add support for Elliptic Curve.

Crypto Proxy Engine Loading

- Using openss.cnf file to identify engine file and arguments

```
[engine_section]
async = async_section

[async_section]
dynamic_path = /home/shinrich/crypto-proxy-engine/proxy-engine.so
engine_id = proxy-engine
client_cert = /home/shinrich/tlstestkeys/client.pem
client_key = /home/shinrich/tlstestkeys/client.key
server_ca = /home/shinrich/tlstestkeys/signer.pem
crypto_proxy_address = 10.0.0.1
crypto_proxy_port = 9999
num_crypto_proxy_threads = 3
init = 1
default_algorithms = RSA
```

RSA/ECDSA Dual Certificates

- It does work.
  - Set up parallel certs in ssl_multicert.config
  - Shown in Examples
- Is anyone running in this production?
  - We are starting to experiment with rolling this out.

ssl_multicert.config moving forward

- Should it be yaml-ized?
- Should it be merged into ssl_server_name.yaml?
  - No fqdn would be needed in that case since the fqdn’s could be pulled from the certificate?
- Merge both into a third new file?