YAHOO!

State of the SSL Onion

Susan Hinrichs – ATS Summit Fall 2016

SSL Ticket KeyBlock Simplification

- Addressed by Persia for 7.0
  - TS-3528: **Create a global for the ticket_key_name from ssl_multicert.config**
  - TS-3371: **Should create a global session ticket disable**
- In addition to the ssl_multicert.config, specify the ticket key block file and enable/disable in records.config
  - In records.config add proxy.config.ssl.server.session_ticket.enable and proxy.config.ssl.server.ticket_key.filename
  - The ssl_multicert.config overrides the global setting in records.config.

SNI and Server Session Reuse

- TS-4468: **http.server_session_sharing.match = both unsafe with HTTPS**
  - Identified by Jered Floyd and original patch from him.
- Problem with SNI names and server session reuse of SSL connections
  - Exposed by additional checking with 2.4 version of Apache Httpd
  - Returns 400 Bad Request on mismatch

SNI and Server Session Reuse

- Consider remaps
  - map https://example.com/ https://origin.example.com/
    map https://www.example.com/ https://origin.example.com/
- Pristine Headers enabled
- ATS will use the host field of server request to set SNI for origin connection.
- Scenario
  - Client makes request to https://example.com/bob
  - ATS negotiates SSL connection to origin.example.com with SNI=example.com
  - Client makes request to https://www.example.com/dave
  - ATS reuses existing connection to origin.example.com. Request host field=www.example.com, but original SNI was example.com. Apache Httpd is mad.

SNI and Server Session Reuse

- After much debate came up with the following fix
  - In addition to testing on port and server FQDN, check that the SNI (if set) matches
- May want to make a “lax” mode for origins that don’t care
  - TS-4839

Revisit Session ID Hooks

- Proposed a while back (April 2015) but got push back and then got distracted by other issues.
- Motivation: Add points for plugins to share Session ID information. When running ATS in a cluster must share the Session information to make Session ID based reuse useful
  - TS-3527
  - API proposal at http://network-geographics.com/ats/docs/ssl-session-api.en.html
  - Had a branch with the API sketched out.
- From Yahoo it looks like Session ID based reuse is still 40-50% of our TLS reuse (tickets being the other part).

SSL Negotiation Refinement

- Use Cases for finer grained decision making about what attributes are available during initial session negotiation
  - Degree of certificate verification
  - Set of protocols we are willing to accept (via NPN/ALPN)
  - Set of ciphers to negotiation
- Can implement most (all?) of this via plugins.
  - Are there standard cases to be set up via static configuration
  - Add to ssl_multi_cert.config? Or another configuration paradigm?

SSL Negotiation Refinement – An actual use case

- Don’t want to advertise Http/2 in some cases
- Solution: Use sni/cert callback to adjust the NPList using TSSslVConnNPListSet
  - Hacky though. Cannot easily adjust default NPList
  - Just overwriting with a fixed list.
  - Need plugin API access to the Server Port object for a cleaner solution.

Client Certificate Support (TS-5022)

- Only global controls
  - Can only specify one client certificate to be supplied to all origin servers
  - Can only specify one set of requirements for requiring certificates from clients
- Seems like there ought to be use-cases that require different client certs for different origin servers
  - Question seems to be coming up more often.
  - Should this be a plugin-specific solution?
    - What hook? Should we add a cert selection callback for the ATS to origin connection?
  - Or is there a common use case that requires finer granularity client certificate specification?
- More Complexity for server session reuse.

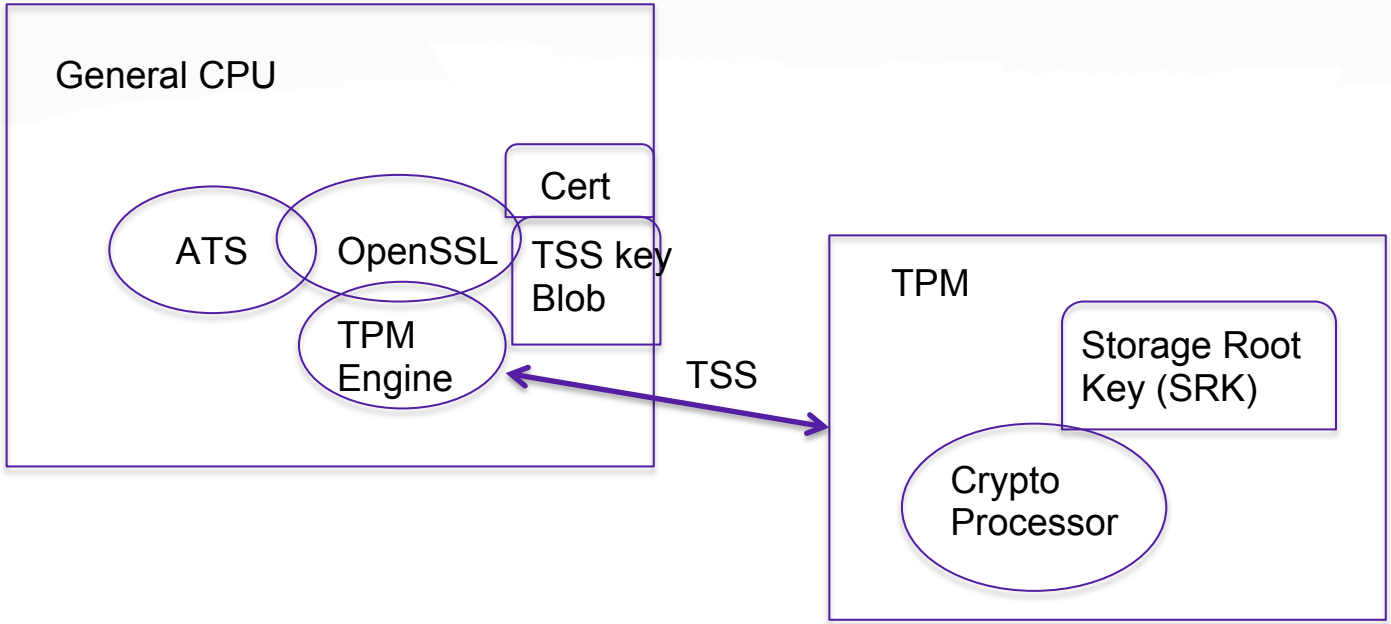
SSL Handshake threads

- Introducing off-box elements to the SSL handshake (CryptoProxy) will cause idle blocking during SSLAccept
  - Even if socket is non-blocking
  - Greatly harms performance
- Previous attempt spawned Ethreads for each SSLAccept call
  - Dave Thompson fixed up into pools
  - Still performance problems (maybe memory leaks)
- Planning on doing a much narrower async thread
  - Native OS thread that only calls SSL_accept and signals READ_READY (or similar) back to original thread when done.

Any News on Alternate SSL Implementations?

- BoringSSL
- LibreSSL
- Others
- Is it time to start looking at TLS 1.3?
  - Not yet, but soon. Beta implementations becoming available

Trusted Platform Module (TPM) Proof of Concept



TPM Tools

- Tpm-tools - tools to manage your TPM
  - Initialize, take ownership, create keys
  - Yum package
- Tpm-emulator
  - Used an emulator rather than starting with hardware
  - Allowed me to test on openstack box
  - https://github.com/PeterHuewe/tpm-emulator
- Libengine-tpm-openssl
  - Creates an openssl engine called “tpm”.
  - Load into your openssl application and it uses TSS to communicate to TPM

TPM Flow

- Load tpm driver
- Initialize the TPM card and “take ownership”
  - Resets Storage Root Key (SRK)
- Create new TPM key
  - Creates TSS Key blob in file
  - Can only be used in conjunction with TPM
- Update ATS
  - To load openssl TPM engine and use engine to load private key
  - 16 lines of code changed – Pushed to tpm-poc branch on my fork. https://github.com/shinrich/trafficserver/tree/tpm-poc
- At this point, SSL handshakes against ATS will use TPM to perform RSA private key operations

Questions?

YAHOO!