Item 281
git_comments:

1. sk_OPENSSL_STRING_pop

git_commits:

1. **summary:** TS-2367: Couple fixes to make builds happy
   message: TS-2367: Couple fixes to make builds happy

github_issues:

github_issues_comments:

github_pulls:

github_pulls_comments:

github_pulls_reviews:

jira_issues:

1. **summary:** Add OCSP (Online Certificate Status Protocol) Stapling Support
   description: RFC: http://tools.ietf.org/html/rfc6066 Overview: https://wiki.mozilla.org/Security/Server_Side_TLS#OCSP_Stapling http://en.wikipedia.org/wiki/OCSP_stapling There is support for this added into openssl 0.9.8g.

jira_issues_comments:

1. Marking for v5.0.0, move back if someone is working on it for v4.2.0.
2. proxy.config.ssl.stapling.enabled: Enable stapling of OCSP responses. Disabled by default. proxy.config.ssl.stapling.cache_timeout: Number of seconds before an OCSP response expires in the stapling cache. 3600s (1 hour) by default. proxy.config.ssl.stapling.request_timeout: Timeout for queries to OCSP responders. 10s by default. proxy.config.ssl.stapling.update_period: Update period for stapling caches. 60s (1 min) by default. When OCSP Stapling is enabled, ATS spawns a new thread to send OCSP request and get OCSP response. The response will be cached for 3600s(1 hour) in server. For details, the request has a timeout when try to connect to CA's OCSP responder, and it's 10s by default. ATS keeps checking for cached response with an update period. In the SSL module, ATS does not send OCSP request in openssl's callback functions. It just try to get response from stapling caches. In this way, the connection won't hang ATS event system.
3. **body:** Some quick comments on style: - this is large enough to have a separate file, {{OCSPStapling.cc}} - looks like you don't need {{HAVE_OPENSSL_OCSP_STAPLING}}, since it is always true - can you remove {{MAX_STAPLING_DER}} and just allocate the size you need? - {{struct certinfo}} doesn't need to be in a header, and should be called {{ocsp_stapling_info}}, or something. If it ends up being needed in the header, follow the naming conventions used there/ - the new settings need to be documented - is there any was we can add regression tests for this?
   label: code-design
4. [~ffcai] Is there any updates on this?
5. Hi [~jamespeach], thanks for review! I made the following changes according to your comments: 1. separate the ocsp stapling's code from iocore/net/SSLUtils.cc, into iocore/net/P_OCSPStapling.h and iocore/net/OCSPStapling.cc 2. move struct certinfo into OCSPStapling.cc, since it's not needed in header file 3. add the new settings' description in mgmt/RecordsConfig.cc HAVE_OPENSSL_OCSP_STAPLING is defined when SSL_CTX_set_tlsext_status_cb is defined in openssl library, in case an old version openssl is used. MAX_STAPLING_DER is 10K, so each certificate will use ~10K size to store ocsp's info. I think it's not too large for common use cases (in my test case, it's about 2K). Do we need to take large amount of certificates into consideration?
6. [~bcall] Any luck with running this patch in production?
7. **body:** It seems to me that OCSP stapling should be enabled by default rather than disabled by default. Current statistics from NetCraft show that 98% of Apache boxes do not use stapling, whereas 98% of Microsoft boxes use stapling. The differentiator is whether stapling is on or off by default--IIS uses OCSP

stapling by default. From a policy perspective, OCSP Stapling is superior for privacy-enhancing and performance reasons because clients do not have to seek a response from a third party - it comes directly from the server, which is why it is also a more efficient mechanism. Also, all major browser platforms support stapling, it is provided in mod-ssl, and because of these reasons, the number of demands for OCSP stapling "out-of-the-box" are likely to grow substantially over the next several months.
label: code-design

8. [~manjeshnilange] Yes, it has been running successfully in production and I will commit it today. [~benwilson] I will ask the other committers what they think about enabling it by default. I would be in favor of that. Here is a SSL Labs test of a server with it enabled in production: https://www.ssllabs.com/ssltest/analyze.html?d=r13.ycpi.sjb.yahoo.net&ignoreMismatch=on
9. **body:** I think {{MAX_STAPLING_DER}} should be removed. The DER copy in {{stapling_get_cached_response}} looks strange; can {{d2i_OCSP_RESPONSE}} just use the DER response in ghee {{certinfo}} struct? I don't know about the blocking {{select}} loop to hit the responders. We can land the change with that, but would you be able to look into using the ATS core HTTP APIs to fetch the responses? {{proxy.config.ssl.stapling.update_period}} isn't really a check periodicity, it's a sleep period between checks. To implement an update period, you could {{schedule_every}}, using a lock to make sure that you don't get concurrent updates. This also saves another background thread. All functions that return 1 or 0 should be declared {{bool}}.
label: code-design
10. After talking to [~jpeach@apache.org] on the IRC: 1. Move sleep from stapling_update() to StaplingUpdateContinuation::mainEvent sleep(SSLConfigParams::ssl_stapling_update_period); 2. Keep the updates in its own thread since they are blocking updates. 3. All functions that return 1 or 0 should be declared bool. 4. The DER copy in stapling_get_cached_response looks strange; can d2i_OCSP_RESPONSE just use the DER response in ghee certinfo struct? 5. Change configuration options with stapling in the name to ocsp. I think users would be able to understand what it does better.
11. **body:** Thanks to [~jpeach@apache.org] and [~bcall]! I made some updates as following: # Use schedule_every() to check/update response. # Use spawn_event_threads() to spawn a thread for OCSP thread. # All functions that return 1 or 0 should be declared bool: Done. # d2i_OCSP_RESPONSE is a heavy conversion, so I do copy first, then release the lock to cinf as soon. I removed stapling_get_cached_response in callback function, and leave the conversion/check to OCSP update thread. # Change configuration options with stapling in the name to ocsp: Done. # I change query_responder() to use openssl's API, OCSP_sendreq_nbio, to implement the unblocking query with timeout option.
label: code-design
12. Thanks! I'll review tomorrow.
13. Commit 562179c50eae3422ac9b4fe50a1b41ea09712ad1 in trafficserver's branch refs/heads/master from [~ffcai] [https://git-wip-us.apache.org/repos/asf?p=trafficserver.git;h=562179c] TS-2367: Add OCSP (Online Certificate Status Protocol) Stapling Support
14. Commit 2621e676c2b3880c5f3d822bf6e0e5cf30c021fc in trafficserver's branch refs/heads/master from [~ffcai] [https://git-wip-us.apache.org/repos/asf?p=trafficserver.git;h=2621e67] TS-2367: Add OCSP (Online Certificate Status Protocol) Stapling Support
15. Commit 17ae8069acb908fece508e323c791e52a8c91a3c in trafficserver's branch refs/heads/master from [~bcall] [https://git-wip-us.apache.org/repos/asf?p=trafficserver.git;h=17ae806] TS-2367: Couple fixes to make builds happy
16. Commit 8bed36eed7819912723a583d2ceeef7cd19ea4b7 in trafficserver's branch refs/heads/master from [~bcall] [https://git-wip-us.apache.org/repos/asf?p=trafficserver.git;h=8bed36e] TS-2367: A better way to declare an argument as unused
17. Commit f5a3d5a2ff8ada63015748532f6904d6d94ed48e in trafficserver's branch refs/heads/master from [~bcall] [https://git-wip-us.apache.org/repos/asf?p=trafficserver.git;h=f5a3d5a] TS-2367: Don't define OCSPContinuation if there is no OCSP support