Main page
Recent changes
Server admin log (Prod)
Server admin log (RelEng)
Deployments
SRE/Operations Help
Incident status

Cloud VPS & Toolforge
Cloud VPS documentation
Toolforge documentation
Request Cloud VPS project
Server admin log (Cloud VPS)

Tools
What links here
Related changes
Special pages
Permanent link
Page information
Cite this page

Print/export
Create a book
Download as PDF
Printable version

Incident documentation/20150820-OCSP

< Incident documentation

Contents [hide]

Summary

On Thursday Aug 20th, we had an OCSP Stapling issue that started around 14:37 UTC. This broke HTTPS connectivity for some browsers. Most reports were from Firefox users, whereas Chrome seemed unaffected. The bulk of affected users reported issues resolved circa 15:10 (~33 mins in), but there were still isolated/rare/intermittent failures until 16:42 (~2h in).

Timeline

14:37 - First browser errors begin appearing to users

14:40 - First Mentioned in ops channel: < marktraceur> Invalid signing cert error on mediawiki.org, just me or...

14:47 - Alex pings Brandon in another channel, Brandon starts investigating

15:00 - Brandon still hasn't fully figured out the issue, but decides to just disable OCSP Stapling for now (https://gerrit.wikimedia.org/r/#/c/232734)

15:10 - Merge -> Deploy process completed across the cache clusters disabling OCSP Stapling. At this point, reporters indicated the issue was gone from the browser POV, although I could still see intermittent full-failures on direct queries for OCSP to GlobalSign of the "trylater" form, which we think browsers don't mind.

... investigating deeper and with less urgency during this window ...

16:23 - Saw an intermittent failure in a browser, and got some confirmation this was still happening (rarely), in spite of our disable of OCSP Stapling. Likely due to bad intermittent-but-infrequent responses directly from GlobalSign (as in expired, not "trylater")

16:42 - Finished deploying a workaround that should hold us for ~12 hours, and give some breathing room to step back and fix this right. We can re-up this workaround manually as necessary if we don't have real fixes in place before then. The workaround is that I've retried GlobalSign manually until I obtained a valid response with a good lifetime window, and then disabled all of our usual automatic OCSP updating mechanisms, pushed the known-good update everywhere, and turned OCSP Stapling back on, so that we send this good update and suppress direct-to-GlobalSign failures in the browser.

Conclusions

There are multiple layers of issue going on here. First, there's the root GlobalSign OCSP response issues:

1. GlobalSign was sending OCSP responses with expired and/or very-nearly-expired signatures (as opposed to OCSP validity windows), at various rates vs other responses. The bad responses would have an 12-hour OCSP validity window starting at the time of the request, with a signature expiry happening in under an hour in the future (or even, in the past, as this event went on). These may still be happening intermittently as I write this, but needs further confirmation.

2. GlobalSign was also sending some "trylater" responses with no real content, at various rates vs other responses, at various times during all of the above. These are still continuing as I write this.

On top that, our process for fetching OCSP from GlobalSign and stapling it to responses (which avoids the browser contacting GlobalSign directly) was designed to not use invalid/faulty responses, but it failed at preventing this issue. We've only been validating the OCSP Window defined by the NotBefore/NotAfter fields, which tend to be short (12 hours); we've implicitly assumed the signature on the response itself (which tends to have a 3 month expiry) would be rotated sufficiently ahead of expiry, which clearly isn't a valid assumption. We still would've eventually run into the same failure, but if the checks had been in place and failing, the cron spam and icinga alerts would've given us a much earlier heads-up about the impending doom, perhaps allowing us to avoid the fallout.

Also, in the "trylater" response case, openssl writes a malformed 5-byte output file and warns of the issue to the terminal, but exits with status code zero (success). Our OCSP script assumed any status-zero would produce a legitimate certificate file. These still ultimately failed at a later verification step and were not moved into place for runtime usage, but this (combined with intermittent failures vs successes in general) added a lot of confusion to the debugging process and slowed everything down.

Also, there's been a long-idle task to make our OCSP Updater more robust in general. We may or may not have noticed the signature expiry issue while working on that, but the intended design changes would still have potentially helped limit the fallout and speed recovery in this case. We should prioritize this more.

Actionables

Short-term:

- ✓ **Done** ocsp updater: handle openssl "trylater" and similar more-gracefully
- ✓ **Done** ocsp updater: validate the signature expiry lifetime
- ✓ **Done** ocsp updater: re-enable automatic updates

Medium-term:

- ↻ **In progress** Make OCSP Stapling support more generic and robust

Category: Incident documentation

This page was last edited on 24 August 2015, at 19:07.