Summary of Services for SOW with Harvey Nash:

- Automate verification of CA test websites per the CA/Browser Forum Baseline Requirements.
- Automate processing of revoked intermediate certs that are reported in CCADB, verification of the updates before committing to OneCRL, and then updating CCADB when OneCRL has been updated.
- Data Integrity Checks for comparing CCADB with OneCRL and ExtendedValidation.cpp.
- Reports/tools for identifying discrepancies in intermediate certificate records in CCADB, such as https://crt.sh/mozilla-disclosures
- Automate CA Mis-issuance checks, such as those found by searching crt.sh for compliance issues when reviewing inclusion requests and performing investigations.

---- DETAILS - not intended for the SOW ----

- Automate Verification of the Test Websites per the BRs
    - Need to make sure that the sites produce the correct results (valid, expired, revoked), and that the SSL cert actually chains up to the indicated root cert.
    - Section 2.2 of the BRs says: “The CA SHALL host test Web pages... At a minimum, the CA SHALLhost separate Web pages using Subscriber Certificates that are (i) valid, (ii) revoked, and (iii) expired.”
    - CAs are required to maintain these 3 test websites for each root cert in Mozilla’s program. Currently Kathleen manually tests these while processing Audit Cases. Verification of these 3 test websites should be automated, so it can be done with the click of a button in Salesforce.
- Automate processing of revoked intermediate certs that are reported in CCADB, verification of the updates before committing to OneCRL, and then updating CCADB when OneCRL has been updated.
    - Currently this is done manually by getting the CRL URL out of the intermediate cert, and using OpenSSL to find the serial number, revocation date, and revocation reason. To make sure the data as reported matches what is in the CRL. It is surprising how frequently there are problems, such as the CRL not having been updated etc. This task would automate the tests that are currently done manually.
    - Also automate verification of the files attached to the “CCADB entries generated…” Bugzilla Bugs. Currently Kathleen copy-pastes the contents of the “Intermediates to be revoked” attachment into https://mozkeeler.github.io/blocklister/index.html then manually compares the output to the list of certs in the CCADB that are “Ready to Add” to OneCRL, and to the excel output of using onecrl2csv.go on the “existing and new revocations in the form of a revocations.txt file” attachment in the bug, which is version of revocations.txt with the changes.
    - Then, after OneCRL has been updated, Kathleen manually checks the revocations.txt file on her system using onecrl2csv.go, and updates the

corresponding CCADB records to indicate that the cert has been “Added to OneCRL” and the Bugzilla Bug number that caused the updates.

- Data Integrity Checks
    - Currently we have a data integrity check that compares what is in CCADB with what is in certdata.txt in Firefox Beta, and generates a report listing the deltas. We need more data integrity checking to compare CCADB with the following.
        - OneCRL
        - ExtendedValidation.cpp in Firefox Beta
        -
- Reports/tools for identifying discrepancies in intermediate certificate records in CCADB
    - Currently we rely on https://crt.sh/mozilla-disclosures to determine when CAs have intermediate certs that have not been disclosed. It would be helpful to have similar reports in the CCADB, to make it easier to identify the problems that we need to file ca-compliance bugs for, or other areas in which we need to make updates. For example, the following sections on the mozilla-disclosures page:
        - Unconstrained Trust: Disclosure is required
        - Disclosed (as Not Revoked), but Revoked via CRL
- Integration between Bugzilla and Root Inclusion Cases in CCADB
    - We will be moving towards having the CAs directly input and maintain their root inclusion data in the CCADB, so will need to automate (as much as possible) sharing the data with Bugzilla. This will require investigation into what sort of integration with Bugzilla is possible.
- Automate Misissuance checks
    - We currently search crt.sh for compliance issues when reviewing inclusion requests and performing investigations. The crt.sh UI is not ideal for our uses, and the command-line SQL is cumbersome.
    - Should be able to specify a root or set of roots and return a nicely formatted list of all revoked and/or non-revoked certificates grouped by the errors identified by certlint, x509lint, and cablint.
    - Ideally, we should be able to filter out OCSP signing certs because they often trigger false errors. Also should be able to filter out intermediate certificates with EKUs not recognized by our program.