#### SWEN90010 - SWEN90010\_2019\_SM1 High Integrity Systems

Assignment\_1

Student:

Bowen Bai 969899

Zeming Yao

Answer:

Question1:

When the log proofs are used, one thing we need to confirm is that the root hash of the Merkle tree should be reliable for the user to verify. If the attack could trick the user to accept a fake root hash, they may make modifications on the leaves as well. For Mozilla’s solution, the certificate which is included in the CT log not only signs a domain, the head of the Merkle tree which summarizes the SHASUM256 files will also be attached to that domain. So, this makes it possible for the verifier to compare the special domain’s first two 16 bytes code to the root hash to confirm that it is the correct one.

Question2:

The whole process of Mozilla’s System is based on the irreversibility of SHA256 or SHA512 which means that the attackers cannot easily work out the binaries from the SHA256SUMS files at least impossible now. What’s more, another basic of this system is that there should not be one even more collisions, they could trick the verifier to accept fake leaves made by them.

Question3

When the verifier is trying to verify an update, the first step is to obtain the certificate for that update and check whether it is trustworthy. For each release, the certificate will contain the Merkle tree head for that release and an SCT as well. So that the verifier could compare the SCT he received and the SCT in the trusted log to confirm this is release or update which has been added to the log. This is not enough, cause the verifier is still not able to confirm that the metadata comes from Firefox, so the next step is to compute the Merkle tree head with the receiving file, and then compare it with the one in the certificate. Without the second step, if the attacker cheats to get an untrustworthy certificate of their fake Merkle tree head, the verifier might also get a malicious release.

Question4

Question5

Question6

Question7