Devin Saverline FullStory Take Home Test Answers

1. How does the FS recording get initialized?

    There appears to be 3 scripts that are used: fs, integrations, and latest.
    As far as I can tell by using breakpoints, fs comes first. In that script it fetches the integrations script and injects it. In the integrations script it fetches the latest script and injects that script.

2. What network traffic is being sent? What do you think each call is doing?

    There is a POST to https://rs.fullstory.com/rec/page which I think sets up the recording by gathering settings. I think this because it is only called when visiting the main page or reloading the main page. It is not called when navigating to other pages or navigating back to the main page from other pages. Also the request has some basic info about the page and the response has things like "OrgSettings".

    The brunt of the calls are POSTs to https://rs.fullstory.com/rec/bundle with the following parameters: OrgId, UserId, SessionId, PageId, Seq, PageStart, PrevBundleTime, and LastActivity. The payload for these calls includes events and arguments for those events.

3. Playback of FullStory recording shows a user journey from page to page in one contiguous "session" (for example, a user lands on the home page, adds items to a cart, then goes through the checkout). How does the recording technology "stitch together" this journey?

    Session info is stored in window['localStorage'].
    I found the following actions:
    Save: window['localStorage'].setItem(key, session)
    Load: var lastSession = window['localStorage'].getItem(key);
        if (!lastSession) {
          lastSession = FS._cookies()[key];
        }
        return lastSession
    The calls to /rec/bundle include a sessionId, so all the calls will be attributed to the same user within the same session.

4. What else can you infer from the script? What would you improve?
    I see that FullStory uses bugsnag.
    I also noticed that FullStory has integrations with at least: Google Analytics, Intercom, Mixpanel, and Qualtrics.
    I see there is an ability to target different environments like staging and local.

As far as improvements, I'm not sure why 3 scripts are needed (unless it is just to adhere to the theory of separation of concerns.)

The process is very noisy (there are a lot of network calls) even when the user is not doing any actions. It does seem there is a limit (after some time of no action the calls stop until another action is performed) but perhaps this could be limited more.

The scripts have a few comments but there could be more comments to make it easier to figure out what is going on.

Also the information sent in the calls to https://rs.fullstory.com/rec/bundle is not really human readable which can make debugging harder. I would suggest looking into making that info more readable.