Comment Template for: NIST SP 800-63-4 Suite (Initial Public Draft)

Please submit responses to dig-comments@nist.gov by March 24, 2023

Organization:	*Intercede*
Name of Submitter/POC:	*Andrew Atyeo*
Email Address of Submitter/POC:	[REMOVED]

Comment #	Publication (Base, 63A, 63B, 63C)	Section	Page #	Line #	Comment (Include rationale for comment)	Suggested Change
1	63B	7.1 Session Binding	48	1817-18	7.1. Session Bindings, lines 1817-1830 "A session secret SHALL be shared between the subscriber's software and the service being accessed. The nature of a session depends on the application, such as: • a web browser session with a "session" cookie, or • an instance of a mobile application that retains a session secret Session secrets SHALL NOT be persistent (retained across a restart of the associated application or a reboot of the host device)" There is a bit of a mixed message as to whether a session cookie counts as a session secret, but I believe the intent is that it does (even though only the 2nd bullet mentions session secret, both bullets are defining examples of a session secret). Assuming a session cookie does count as a session secret, then how to enforce the requirement that "session secrets SHALL NOT be persisted?" Since a web application runs in a browser, the persistence of the cookie is controlled by the browser. The website/webserver/application can set headers to tell the browser about cookie expiry etc, but it is the browser that is in control. To prevent a cookie being retained during a restart/reboot, such a cookie must not have an expires date (so it will be a session cookie). Since a browser has no knowledge of a reboot of host device, a cookie with an expiration date could persist beyond a reboot/restart. However, many browsers have a "session restore" feature that persists and restores session cookies (ie cookies without an expires date), which may prevent session cookies meeting the requirement (of session secrets/cookies not being persisted). – See the note about "session restore" in https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Set-Cookie This could mean that cookies cannot meet the requirements to be a session secret – but I know that is not the intent – the whole push towards federated identity essentially is a push towards (browser) cookie based sessions	I am not sure how to resolve this, but one outcome could be to work with the w3c to define an additional attribute that can be set for a cookie that marks it as never being persisted?. However unless browser vendors are on board it would not be implemented in practice. An alternative may be to change the "Session secrets SHALL NOT be persisted" to a SHOULD NOT. Or a clause relating to browsers (indicating that browser cookies are marked to prevent persistence, since as we know, a webserver can only set attributes on a cookie to instruct a browser, it cannot actually control the browsers behaviour): "Session secrets managed in code SHALL NOT be persisted. Session secrets implemented as browser cookies SHALL instruct the browser to not persist the cookie"
2	63B	5.1.1.2 Memorized	15	711-724	"Passwords obtained from previous breach corpuses" – it is unclear whether this applies specifically to the user, or general. If in general, what would be the criteria for including passwords from previous breach corpuses especially when considering line 722+ "Excessively large blocklists SHOULD NOT be used because they frustrate subscribers' attempts to establish an acceptable memorized secret and do not provide significantly improved security". We can see that this could be taken as some kind of user specific list, frequency of occurrence, or how recent the breach has been. "Excessively large blocklists SHOULD NOT be used because they frustrate subscribers' attempts to establish an acceptable memorized secret and do not provide significantly improved security" – we disagree with these statements. We currently use a blocklist of 1.4 billion known breached passwords and have had no feedback that users find it hard to choose a new password / have excessive rejections of passwords as long as the feedback to the user is specific enough that this is why their password is being rejected. We also feel that large blocklists are in fact the only way in which blocklists can be effective, because it is not possible to know all of the users' online identities when they are choosing a new password. A small general blocklist will just ignore the very relevant set of known breached passwords for a specific individual, and often those passwords are not directly tied to the username or email for which the new password is being created. In fact when legitimately recovering passwords for users (in order to maintain breach blocklists), the first thing we will do is manually build a list of known passwords across all their online identities using Open Source Intelligence techniques and matching lastnames and location information, linked in profiles etc.	[REMOVED]
3	63-Base	5.1.3 Identify Poten	28	1100	missing heading? (I think the "low/moderate/high" bullets on line 1101-1109 are meant to be for "loss of sensitive information" But the heading is missing. This may be just a formatting issue in the output of this draft as pdf.	Review formatting on that section, it seems (at least on the PDF) that a heading is missing.