### Finding: X.com vulnerable to Session Fixation attack

**DREAD Score Summary**

**STATUS: verified**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Remediation Effort | | | undefined | | | | |
| Damage Potential | Reproducibility | Exploitability | | Affected Users | Discoverability | Total | DREAD Risk |
| 6 | 8 | 8 | | 7 | 7 | 36 | Moderate |

**Affects:**

undefined

**Description:**

Rapid7 discovered the existence of a session fixation vulnerability on knowledge.elmentk.com. By viewing the session ID before and after a successful login, it was determined that the sessionID (JSESSIONID) is indeed the same, and has not been regenerated. This would allow an attacker to generate a valid session (before logging in) and present this to a victim. Once the victim clicked (or was made to browse to the link without their knowledge), the attacker would have access to their session.

**Proof:**

Viewing the session ID before and after a successful login shows that the sessionID is indeed the same, and has not been regenerated.

**Recommendations:**

Regenerate JSESSIONID after valid login:In Java, there is not well-supported programmatic access to the JSESSIONID cookie. You can't change the name of the cookie or set cookie properties (such as HttpOnly) very easily. However, in Java, it is possible to invalidate and recreate the session (not the session ID, necessarily) using the following code:{{{session.invalidate();session=request.getSession(true);}}}This is also the OWASP recommendation, although it isn't explained in much detail on the site. Note I said that this does not regenerate the session ID necessarily. Looking at the comment thread for this blog, it appears JBoss doesn't regenerate the JSESSIONID using this code. I haven't confirmed this myself, but there is no guarantee in the Java specification that the session.invalidate will actually regenerate the session ID if it is recreated using request.getSession during the same request. So be careful and test if you use this method - I tried the code with Tomcat, and it worked, but I can't vouch for the other Java application servers.In addition, you have to manually copy over all of the current data in the session. This may be necessary when designing an experience like Amazon.com, where you can add things to your cart before you log in. You would have to make sure that this information is persisted in the new session after authentication. This is an inconvenience, but is certainly doable.

**References:**

\* Session Fixation \* ACROS Security: http://www.acrossecurity.com/papers/session\_fixation.pdf \* Chris Shiflett: http://shiflett.org/articles/session-fixation \ \* http://keepitlocked.net/archive/2007/12/26/preventing-session-fixation-through-session-id-regeneration-in-java-and-asp-net.aspx