**A5 Broken Access Control**

* **Leicht Verwundbar, wenn**
  + nicht geprüft wird, ob alle Daten- und Funktionsreferenzen angemessene Verteidigungsmechanismen haben. D.h. wird überprüft, ob ein bestimmter Nutzer für den Zugriff auf bestimmte Daten oder Funktionen autorisiert ist
  + ein System zulässt, dass Zugriffskontroll-Überprüfungen umgangen werden können, indem URL, interner App-Zustand oder die HTML-Seite modifiziert werden
  + ein System erlaubt, dass der Primärschlüssel zu dem Eintrag eines anderen Nutzers geändert wird, sodass der Account dieses Nutzers eingesehen und verändert werden kann
* **Beispiel Attacken**
  + Angreifer kann mit einem SQL-Aufruf, bei dem auf Account-Informationen mit nicht verfizierten Daten zugegriffen wird, Zugriff auf jeden Account erhalten
  + Angreifer erhält durch Eingabe entsprechender URLs z.B. Zugriff auf Adminseite
* **Wie verhindert man Attacke auf verwundbare Stellen der/des Software/Betriebssystems**
  + Zugriffskontrolle nur in vertrauten, serverseitigen Code oder serverlosen API durchführen, wo der Angreifer nicht die Zugriffskontrollen-Überprüfungen oder Metadaten modifizieren kann
  + Einmaliges Implementieren der Zugriffskontroll-Mechanismen und Verwenden dieser in der gesamten Applikation
  + Zugriffskontrollen sollten die Eigentümer der Einträge prüfen, anstatt zu akzeptieren, dass der Nutzer jeden Eintrag erstellen, lesen, aktualisieren oder löschen kann
  + Deaktivieren der Webserver-Verzeichnis-Auflistung und sicherstellen, dass sich Datei-Metadaten (z.B. .git) nicht innerhalb Webroots befinden
  + Loggen der Zugriffskontroll-Fehlschläge und Alarmieren der Admins wenn angemessen (z.B. bei wiederholten Fehlschlägen)
* **Ziele eines Angreifers (Umgebung und Nutzen)**
  + unautorisierte Inhalte einsehen, verändern oder löschen
  + nicht autorisierte Funktionen ausführen
  + Administration der Seite übernehmen
* **Gefährdete Personengruppen**
  + Unternehmen mit Web-Präsenz
  + Personen mit Web-Konten