LDAP & OAuth 2.0

LDAP ist ein offenes und plattformübergreifendes Protokoll für die Verzeichnis-Dienste-Authentifizierung. LDAP stellt eine Kommunikationssprache bereit, die Anwendungen verwenden, um mit Verzeichnisdiensten zu kommunizieren. Verzeichnisdienste speichern die Kennwörter für Computer- und Benutzerkonten und geben diese an andere Netzwerkdienste weiter.

Die LDAP-Benutzer-Authentifizierung ist ein Prozess zur Validierung einer Kombination aus Benutzernamen und Passwort in Verbindung mit einem Verzeichnisserver, wie z.B. dem Microsoft Active Directory.

Das Microsoft Active Directory ist der weitverbreitetste Verzeichnisdienst, unterstützt Kerberos sowie LDAP und stellt Single-Sign-On (SSO) zur Verfügung.

Es gibt 2 Optionen für die LDAP v3 Authentifizierung.

1. Einfache Authentifizierung
    a. Anonyme Authentifizierung – gewährt einem anonymen Benutzer Zugriff
    b. nicht authentifizierte Authentifizierung – für Logging Zwecke falls dem Client kein Zugriff gewährt wurde
    c. Benutzername/Kennwort Authentifizierung – gewährt einem Zugriff auf Basis der Anmeldeinformationen
2. SASL Security Layer
    a. Die SASL Authentifizierung bindet den LDAP Server an einen Authentifizierung-Mechanismus wie z.B. Kerberos

LDAP = Lightweight Directory Access Protocol
OAuth = Open Authorization v2

OAuth ist ebenfalls ein offenes aber Token basiertes standarisiertes Protokoll. OAuth erlaubt eine sichere Autorisierung und Authentifizierung beim Einsatz von Desktop-, Web- und API- basierten Anwendungen.

OAuth wird eingesetzt, um einen entfernten Dienst den Zugriff auf Daten über eine Anwendung zu ermöglichen.

„Der Benutzer erlaubt einer Drittanwendung den Zugriff auf Daten die bei einem anderen Dienst gespeichert sind."

OAuth arbeitet mit 4 verschiedenen Rollen.

1. **Resource Owner** = Ressourcenbesitzer
    a. Der Resource Owner gewährt der Drittanwendung Zugriff auf seine Ressourcen
2. **Resource Server** = Ressourcenserver
    a. Der Resource Server hält die schützenswerten Daten vor
3. **Client** = Drittanwendung / Web Service, lokale oder mobile Anwendung
    a. Die Drittanwendung wünscht den Zugriff auf die geschützten Daten
4. **Authorization Server** = Autorisierungsserver
    a. Der Authorization Server authentifiziert den Resource Owner und stellt einen Zugriffstoken aus

Hinweis: Der Authorization Server und Resource Server werden in der Regel zusammen betrieben. Ein Zugrifftoken hat eine zeitlich begrenzte Gültigkeit.

LDAP & OAuth 2.0

Das Konstrukt funktioniert wie folgt:

Das Ziel ist die Unterbindung der Herausgabe von Benutzernamen und Passwort eines Anwenders/Resourcen Owner an den Client/Drittanwendung.

Mal ganz ehrlich, gibst Du einem Händler deine EC-PIN?!

Ich bin Kunde bei Fotobuch.de und habe dort einen Account. Ich melde mich an und möchte meine Bilder über einen angeschlossen Fotobuch.de Druckdienst ausdrucken lassen. Der angeschlossene Dienst ist eine Druckerei. Damit die Druckerei Zugriff auf meine hochgeladenen Fotos bei Fotobuch.de bekommt, benötigt die Druckerei meine Zugangsdaten von Fotobuch.de. Natürlich geben wir diese nicht heraus!

Damit die Druckerei meine Fotos aber drucken kann, muss sich die Druckerei gegen Fotobuch.de authentifizieren und fordert so eine Autorisierung über den Resourcen Owner an. Nach der Authentisierung bekommt die Druckerei über OAuth vom Authorization Server einen einmaligen Zugriffstoken, der nur für den Zugriff auf die zu druckenden Fotos gültig ist.

In der Regel ist es so, dass ich die Webseite der Druckerei öffne, dort auf einen Hyperlink klicke, der mich wiederum zu Fotobuch.de weiterleitet. Melde mich nun bei Fotobuch.de mit meinem Benutzernamen und Kennwort an. Die Druckerei bekommt über die zustanden gekommene Session einen Zugriffstoken. Der Zugriff auf die Resource ist nun möglich.

Authentisierung = Ist der Nachweis einer Identität
Authentifizierung = Bestätigung der Identität
Autorisierung = ist eine Berechtigung

Was wird benötigt um das erste Zugriffstoken und Aktualisierungstoken zu erhalten?

- User-ID
- User-Password
- Client-ID
- Client-Secret (zufällige Zeichenfolge) *

Was wird benötigt um ein aktualisiertes Zugriffstoken zu erhalten?

- Client-ID
- Client-Secret (zufällige Zeichenfolge)
- Aktualisierungstoken

*Die zufällige Zeichenfolge ist dem Client und dem Autorisierungsserver bekannt.

Hier wird deutlich, dass der Unterschied darin liegt, dass der Client die Berechtigung zum Aktualisieren von Zugriffstoken unter Verwendung des ihm bekannten Client-Secret bekommt. Dies ermöglicht die Aufrechterhaltung einer aktiven Verbindung ohne erneut die User-ID und das User-Password eingeben zu müssen.

Dies zeigt auch, dass der Verlust eines Aktualisierungs-Tokens kein Problem sein sollte, weil die Client-ID und das Client-Secret nicht bekannt sind. Es zeigt auch, dass es wichtig ist, die Client-ID und das geheime Client-Secret geheim zu halten.

Somit wird klar, dass das Access-Token die Anmeldeinformationen und Daten zur Identifizierung eines Benutzers enthält und als sehr schützenswert anzusehen ist.

Erstellt von Jörn Walter **08.12.2018**

Die Client-Secret (zufällige Zeichenfolge) wird als Klartext benötigt und sollte auch Secret bleiben. Leider kann das Client-Secret nicht wie ein Passwort gehasht werden, und aus diesem Grund sollte es außerhalb des WebRoot-Verzeichnisses gespeichert werden.