www.bt-magazin.de 2.2015 Heft 21

€ 4,90

S&S

Business Technology

Business Technology

Lean Innovation & IT Leadership

SSO mit SAML in der Praxis

Implementierung von SAML

Viele Unternehmen nutzen bereits SAML, um eine bessere Benutzererfahrung für ihre Kunden und Mitarbeiter zu schaffen. Allerdings gibt es ebenso viele Unternehmen, die vor der vermeintlichen Komplexität von SAML zurückschrecken. Dieser Artikel zeigt, wie SAML SSO funktioniert, und beschreibt zwei verschiedene Methoden der Umsetzung. Für jede dieser Methoden werden Vor- und Nachteile diskutiert.

AUTOREN: MAMOON YUNUS UND DR. DIRK KRAFZIG

Was ist SAML? Die Security Assertion Markup Language (SAML) [1], [2], [3] bietet zahlreiche Vorteile für Unternehmen. Eine der größten Fähigkeiten ist Single Sign-On (SSO), die Möglichkeit, Benutzer sicheren Zugriff auf mehrere Anwendungen mit einem einzigen Satz von Anmeldeinformationen zu ermöglichen.

SAML ist ein XML-basiertes Framework für die Autorisierung und Authentifizierung sowie für die Kommunikation von Attributen und Berechtigungen eines Benutzers. Um SAML SSO zu verstehen, muss man zunächst die drei Teilnehmer betrachten: den Benutzer, den Identity-Provider (IdP) und den Service-Provider (SP). Der Benutzer ist diejenige Partei, die versucht, eine Anwendung aufzurufen. Der Service-Provider stellt die Anwendung zur Verfügung. Der Identity-Provider ist der Inhaber der Anmeldeinformationen der Benutzer, wie Benutzername, Passwort, x509, Rollen usw.

Es gibt zwei unterschiedliche Methoden, um SAML SSO zu implementieren: Service-Provider-initiiert und Identity-Provider-initiiert.

Artikelserie

Teil 1: Anwendungssicherheit SAML – Eine Einführung
Teil 2: Implementierung von SAML SSO in der Praxis

Beim SP-initiierten SSO versucht ein Benutzer, den Service-Provider direkt aufzurufen. **Abbildung 1** zeigt den Ablauf der Anmeldung:

1. Der Benutzer versucht, auf eine Anwendung vom Service-Provider per URL zuzugreifen.
2. Der Service-Provider leitet den Benutzer im Browser auf den Identity-Provider für die Authentifizierung.
3. Der Benutzer liefert dem Identity-Provider Anmeldeinformationen. Wenn die Anmeldeinformationen gültig sind, erstellt der IdP ein SAML-Token, signiert es und bettet es in einen HTTP-Redirect-Befehl ein, der den Benutzer wieder an die Anwendung zurückleitet.

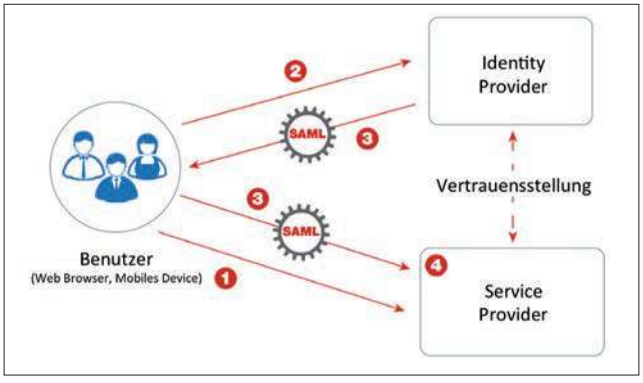

Abb. 1: Service-Provider-initiiertes SSO

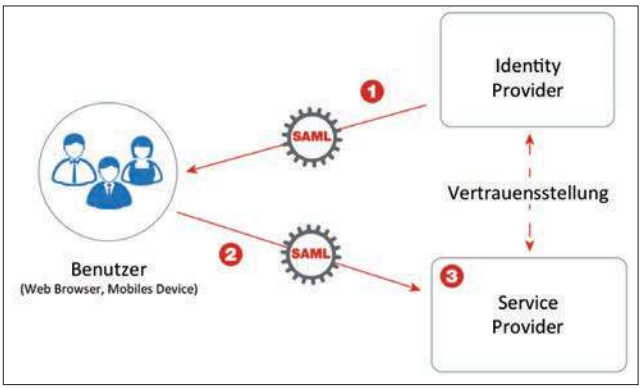

Abb. 2: IdP-initiiertes SSO

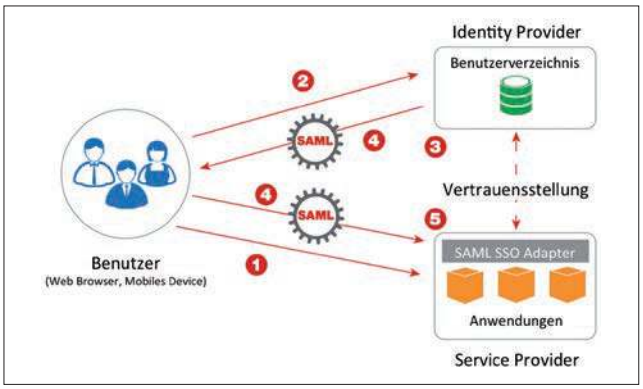

Abb. 3: SAML-Fähigkeit in die Endpunkte integriert

4. Der Service-Provider überprüft das eingehende SAML-Token aus der Weiterleitung. Dazu validiert er die Signatur des Tokens und prüft die SAML-Inhalte. Wenn das Token gültig ist, wird ein Session-Cookie erstellt, um die Nutzung der SSO-Sitzung für eine definierte Zeitdauer ohne erneute Authentifizierung zu erlauben.

Bei IdP-initiiertem SSO ruft der Benutzer zunächst den Identity-Provider auf, um sich zu authentifizieren. Der Identity-Provider leitet erfolgreich authentifizierte Benutzer zum Service-Provider weiter. Den Ablauf sieht man in **Abbildung 2**.

1. Der Benutzer klickt auf einen Link zu einem Identity-Provider. Der Benutzer wird durch den Identity-Provider authentifiziert. Der IdP gibt einen signierten SAML-Token an den Benutzer heraus.
2. Der Browser des Benutzers wird zum Service-Provider umgeleitet.
3. Der Service-Provider überprüft das eingehende SAML-Token aus der Weiterleitung, validiert es durch eine Signaturprüfung und prüft die SAML-Inhalte. Wenn das Token gültig ist, wird für den Benutzer ein Session-Cookie vergeben, um die Nutzung der SSO-Sitzung für eine festgelegte Zeitdauer ohne erneute Authentifizierung zu gestatten.

Die beiden Beispiele wurden zur Verdeutlichung ein wenig vereinfacht und die technischen Details, die mit jedem Schritt verbunden sind, wurden weggelassen. Der nächste Abschnitt geht nun weiter ins Detail [4].

IMPLEMENTIEREN VON SAML SSO

Die Umsetzung von SAML SSO ist für Anfänger sicherlich eine Herausforderung. Sowohl die Erzeugung als auch die Auswertung von SAML-Token kann komplex sein und Wissen über Public-Key-Infrastrukturen (PKI), digitale Signaturen (DSIG) und SAML-Vertrauensstellungen erfordern sowie über entsprechende Programmierframeworks und zugrunde liegende OASIS-Standards.

Eine Einführungsstrategie von SAML ist von mehreren Faktoren abhängig – z. B. fachliche Architektur, Technologie von Altsystemen, Kommunikationsmuster, Erfahrung des Entwicklungsteams und Kosten für den Bau individueller SAML-Adapter.

Dabei lassen sich zwei sehr unterschiedliche Einführungsstrategien zur Umsetzung von SAML SSO unterscheiden. Die erste Option ist der Bau oder Kauf eines SAML-SSO-Adapters für jeden Service-Provider, der am SSO teilnehmen soll (manchmal auch als SAML-Agenten bezeichnet). Die zweite Möglichkeit ist es, ein API-Gateway einzusetzen, mit dem die meisten Teile des SAML-Protokolls zentralisiert abgedeckt werden. Dadurch werden die Auswirkungen auf die Backend-Dienste weitgehend minimiert. Für beide Einführungsstrategien ist allerdings zu beachten, dass die Benutzeridentitäten in einem geeigneten Identity-Provider zentralisiert sein müssen.

EINFÜHRUNGSSTRATEGIE 1: SAML-SSO-ADAPTER

Entscheidet man sich dafür, die SSO-Logik in jeden einzelnen Endpunkt zu legen, bietet SAML den Vorteil, durch seine offenen Standards einen gewissen Grad an Herstellerunabhängigkeit zu erreichen. Dies gilt sowohl für den Umgang mit Identitäten als auch für den Ablauf des SSO an sich. Hiermit wird ein wesentlicher und sehr positiver Beitrag zu jeder Modernisierungsstrategie geleistet. Für den praktischen Erfolg dieser Einführungsstrategie stehen zwei Faktoren: a) die Fähigkeit, SAML technisch in den unterschiedlichen Ablaufumgebungen der vorhandenen Legacy-Komponenten umzusetzen und b) die dazugehörigen Entwicklungs- und Betriebsteams für den SAML-Ansatz zu begeistern und mit dem nötigen SAML-Know-how zu versorgen.

Der Zweck der Adapter ist es, SAML-fähige, standardisierte Komponenten – wie in **Abbildung 3** dargestellt – in Applikationen einzubetten.

1. Der Benutzer versucht, auf eine Anwendung zuzugreifen.
2. Der Benutzer ist nicht authentifiziert und wird vom SAML-SSO-Adapter zum Identity-Provider zur Authentifizierung umgeleitet.
3. Der IdP ermittelt für den Benutzer die Anmeldeinformationen und authentifiziert den Benutzer.
4. Nach erfolgreicher Authentifizierung erstellt der IdP ein signiertes SAML-Token und leitet den Benutzer zurück zum Service-Provider.
5. Der SAML-SSO-Adapter im Service-Provider prüft das SAML-Token. Wenn das Token gültig ist und der Benutzer zugreifen darf, wird ein Session-Cookie erstellt, um die Nutzung der SSO-Sitzung für eine festgelegte Zeitdauer ohne erneute Authentifizierung zu erlauben.

In der Praxis gibt es bei dieser Einführungsstrategie einige Herausforderungen zu berücksichtigen:

- Das Bauen eigener Adapter kann Zeit und Ressourcen Ihres Entwicklungsteams binden, die dann für fachliche Weiterentwicklung verlorengeht.
- Die fertigen Adapter erfordern weitere Bandbreite für laufende Wartung und Fehlersuche.
- Jeder neue oder aktualisierte SP muss eine gründliche Sicherheitsüberprüfung durchlaufen, da jeder SP eigene Angriffsfläche bietet.
- Adapter beschützen den Service-Provider nicht vor Gefahren der zugrunde liegenden Plattform wie z. B. unsicheren SSL-Implementierungen.
- Für jeden einzelnen Service-Provider müssen die Auswirkungen der SSO-Implementierung auf Performance und Skalierbarkeit berücksichtigt werden.
- Die dezentrale Architektur erfordert eine strikte Governance, oder sie kann zu potenziellen Unstimmigkeiten in verschiedenen Umgebungen führen und erhebliche Aufwände in der Betreuung nach sich ziehen.

EINFÜHRUNGSSTRATEGIE 2: SAML SSO MIT EINEM API-GATEWAY

In der zweiten Einführungsstrategie wird SAML mit einem API-Gateway in die Unternehmensarchitektur integriert. Anstelle der Adapter übernimmt das API-Gateway die Verantwortung für den Umgang mit SAML. **Abbildung 4** veranschaulicht den Aufbau der Architektur mit einem API-Gateway:

1. Der Benutzer versucht, auf eine Anwendung zuzugreifen.
2. Da der Benutzer nicht authentifiziert ist, leitet das API-Gateway den Benutzer im Browser zur Authentifizierung zum Identity-Provider.
3. Der IdP ermittelt für den Benutzer die Anmeldeinformationen und authentifiziert den Benutzer.
4. Wenn die Anmeldeinformationen gültig sind, erstellt der IdP ein signiertes SAML-Token und leitet den Benutzer zurück zum API-Gateway.
5. Das API-Gateway prüft das SAML-Token und dessen Inhalt. Wenn das Token gültig ist, räumt das API-Gateway den Benutzerzugang ein und erstellt ein

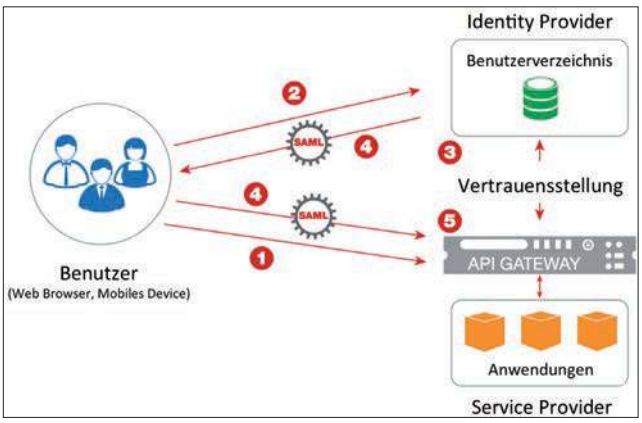

Abb. 4: SAML-Fähigkeit in einem API-Gateway integriert

Session-Cookie, um die Nutzung der SSO-Sitzung für eine festgelegte Zeitdauer ohne erneute Authentifizierung zu erlauben.

Die Verwendung eines API-Gateways lindert viele der im vorherigen Szenario genannten Herausforderungen:

- Es ist weniger Kodierung erforderlich. Ihr Entwicklungsteam muss nicht alle Änderungen an allen Service-Providern durchführen. Die Verwaltung und Wartung von Gatewaykonfigurationen erfordert weniger Zeit und Ressourcen.
- Ein API-Gateway bietet typischerweise einen höheren Grad an Sicherheit, als Sie mit realistischem Budget mit einem internen Entwicklungsteam erreichen können.
- Skalierbarkeit und Performance von SSO berühren die Anwendungen nicht mehr.
- Das API-Gateway zentralisiert Testen, Debuggen, Überwachung, Prüfung, Wartung, Änderungen der Sicherheitsregeln an einem zentralen Ort.

BENUTZERFREUNDLICHE AUTHENTIFIZIERUNG

Neben der Zentralisierung der Benutzerinformationen und Standardisierung der Technologie möchte man durch den Einsatz von SAML SSO auch den Komfort für die Anwender erhöhen. In vielen Anwendungsszenarien möchte man daher erreichen, dass ein Benutzer, der sich bereits erfolgreich an seinem Endgerät authentifiziert hat, ohne weitere Rückfrage für Anwendungen berechtigt wird, die am SSO teilnehmen. Um dies zu ermöglichen, gibt es zahlreiche technische Möglichkeiten:

- Einsatz von Benutzerzertifikaten, die auf dem Endgerät abgelegt werden.
- Verwendung des SPNEGO-Protokolls, mit dem der Browser das zugrunde liegende Betriebssystem nach den Benutzerdaten (KERBEROS-Identität) fragen kann.
- Einmalige Log-in-Maske am IdP und Vergabe eines SSO-Tokens.

ZUSAMMENFASSUNG

Zusammenfassend können wir folgende Empfehlung geben: Wir raten zum Ansatz mit SAML-SSO-Adaptern, wenn Ihr Unternehmen nur wenige Anwendungen betreibt, eine homogene Technologielandschaft besitzt und Geschäftslogik sowie Sicherheitsregeln nur einer geringen Änderungsrate unterworfen sind. Auf der anderen Seite empfehlen wie den Ansatz mit einem API-Gateway, wenn Ihr Unternehmen sehr viele Anwendungen betreibt, Ihre Technologielandschaft heterogen ist und Änderungen in Geschäftslogik und Sicherheitsregeln häufig auftreten.

Links & Literatur

[1] Security Assertion Markup Language (SAML) 2.0 Technical Overview: https://www.oasis-open.org/committees/download.php/11511/sstc-saml-tech-overview-2.0-draft-03.pdf
[2] SAML Executive Overview: https://www.oasis-open.org/committees/download.php/11785/sstc-saml-exec-overview-2.0-draft-06.pdf
[3] Advantages of SAML: http://saml.xml.org/advantages-saml
[4] Weitere Informationen zu SAML-Sicherheit in „Anwendungssicherheit mit SAML“, Yunus und Krafzig, BT 1.2015 Seite 56, und unter: https://jaxenter.de/anwendungssicherheit-saml-eine-einfuehrung-18066

Mamoon Yunus

ist CEO von Crosscheck Networks, einem führenden Technologieanbieter für Cloud- und Web-Services-Infrastrukturen. Als SOA Pionier und Gründer von Forum Systems hat er wichtige Techniken für XML Appliances patentieren lassen. Er besitzt zwei Abschlüsse vom MIT. InfoWorld hat ihn 2004 als einen von vier „Up and coming CTOs to watch“ ausgezeichnet.

Dr. Dirk Krafzig

ist Gründer von SOAPARK. Als Sprecher auf Konferenzen und Autor von Artikeln und Büchern gilt Dr. Krafzig als ein Protagonist der serviceorientierten Architektur (SOA) und hat maßgeblich zu der Begriffsbildung in diesem Bereich beigetragen. Insbesondere die SOA-Fallstudien mit der Deutschen Post, Credit Suisse, Halifax Bank of Scotland und Winterthur Versicherung in seinem Bestseller „Enterprise SOA“ haben viel Aufmerksamkeit auf sich gezogen. Derzeit arbeitet Dr. Krafzig in einem strategischen SOA-Programm bei einem Mobilfunkanbieter an dem Thema Security.