Bảo mật API cho ứng dụng web

Ngày nay, Internet được coi là một kênh truyền thông quan trọng giúp các doanh nghiệp dễ dàng tiếp cận và cung cấp thông tin cho khách hàng. Việc xây dựng các ứng dụng web để kết nối là điều cần thiết, tuy nhiên có rất nhiều mối quan tâm cạnh tranh mà chúng ta cần phải cân bằng. Tìm ra cách bảo mật ứng dụng và quản lý người dùng là rất quan trọng, nhưng hiếm khi những tính năng này có thể tạo nên sự khác biệt cho ứng dụng và thu hút được nhiều khách hàng sử dụng. Hiện nay, khách hàng mong đợi hơn rất nhiều cho vấn đề trải nghiệm đăng ký và đăng nhập liền mạch, nếu tự mình viết tất cả mã để hỗ trợ hầu hết các chức năng thì cực kỳ tốn thời gian và tốn kém trong khi chúng ta chỉ có nguồn lực hạn chế và thời gian hạn hẹp. Việc tìm cách chuyển gánh nặng phát triển này sang các công cụ hiện có là điều bắt buộc. Dịch vụ API Gateway của Amazon là một trong những lựa chọn đáng để chúng ta tham khảo.

API Gateway là một dịch vụ mới giúp dễ dàng tạo và xuất bản API RESTful trên nền tảng đám mây. Dịch vụ này cung cấp rất nhiều tính năng để quản lý API nhưng ở bài viết này, chúng ta chỉ giới thiệu và thảo luận các vấn đề bảo mật liên quan đến việc cho phép người dùng truy cập vào các API hỗ trợ công khai (Public APIs). Cách tiếp cận hiển thị và bảo mật các điểm cuối RESTful qua Internet bằng API Gateway giúp dễ dàng phát triển nhiều ứng dụng khách chuyên biệt cho các nền tảng và thiết bị khác nhau.

Quy trình xác thực và ủy quyền (Authentication and Authorization)

Ví dụ bạn đang điều hành một cửa hàng mua bán thú cưng và đang phát triển dịch vụ web để quản lý cửa hàng. Dịch vụ này truy xuất hiển thị hiển thị danh sách vật nuôi để bán theo phương thức GET và thêm vật nuôi mới vào danh sách theo phương thức POST. Bạn muốn đảm bảo rằng chỉ quản trị viên cửa hàng của bạn có thể thêm danh sách mới, nhưng bất kỳ ai đã đăng nhập đều chỉ có thể xem danh sách vật nuôi được rao bán.

Khi lập kế hoạch cho một hệ thống bảo mật như thế này, chúng ta cần xem xét hai mối quan tâm cơ bản: xác thực và ủy quyền. Xác thực là cách xác nhận rằng người dùng đúng là họ đúng như những gì họ nói và ủy quyền là cách xác định những gì người dùng đó được phép làm. Tiếp tục với ví dụ về cửa hàng thú cưng, giả sử Alice là quản trị viên (có thể đăng thông tin trên trên cổng dịch vụ) và Bob là khách hàng hợp pháp (chỉ xem thông tin). Nếu một người khách cố gắng giả làm Alice để đăng danh sách giả mạo, người khách đó sẽ thất bại ở bước xác thực. Nếu Bob đăng nhập với tư cách là chính mình và cố gắng đăng danh sách, anh ấy sẽ không thực hiện được ở bước ủy quyền vì khách hàng không ủy quyền cho phép làm điều đó.

Trở lại việc xây dựng dịch vụ bảo mật các phương thức API riêng lẻ thông qua API Gateway của Amazon, chúng ta có thể quản lý cả hai vấn đề xác thực và ủy quyền thông qua dịch vụ định danh và quản lý truy cập (IAM - Identity and Access Management) của AWS (Amazon Web Services). Quy trình xác thực về việc cho phép người dùng trao đổi tên (username) và mật khẩu (password) của họ để lấy một bộ thông

tin đăng nhập IAM được sử dụng để xác thực các yêu cầu API riêng lẻ. Sau đó, API Gateway sẽ xử lý quy trình ủy quyền bằng cách sử dụng các chính sách IAM được liên kết với các thông tin xác thực đó.

Xác thực người dùng (User Authentication)

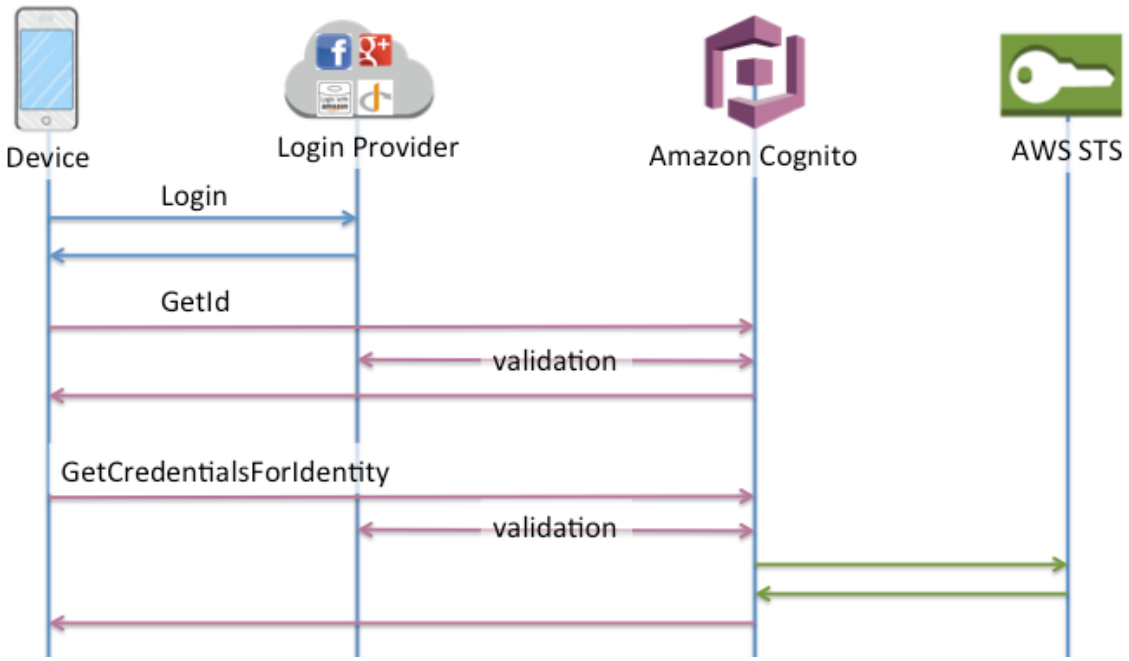
Để truy cập một API được lưu trữ trên API Gateway, người dùng cần phải có một bộ thông tin xác thực IAM. Tuy nhiên, việc cung cấp thông tin xác thực IAM tồn tại lâu dài cho tất cả người dùng - đây là một phương pháp bảo mật rất tệ, rất khó quản lý việc cung cấp và thu hồi, chưa kể trải nghiệm người dùng sẽ khá rắc rối. Thay vào đó, cách tiếp cận tốt hơn là sử dụng nhà cung cấp định danh bên ngoài để quản lý xác thực người dùng và tìm nạp thông tin xác thực IAM tạm thời để thực hiện lệnh gọi API. Nhà cung cấp định danh cũng có thể xử lý tất cả các tác vụ quản trị liên quan như cấp phép người dùng, đặt lại mật khẩu, v.v. Đối với các dịch vụ Internet hướng tới công chúng, việc tích hợp với nhà cung cấp như Amazon hoặc Facebook thường là hợp lý, nhưng chúng ta cũng có thể sử dụng các nhà cung cấp OpenID Connect khác hoặc viết dịch vụ xác thực của riêng mình được hỗ trợ bởi cơ sở dữ liệu độc quyền.

Bất kể lựa chọn tích hợp với nhà cung cấp định danh nào, chúng ta cũng sẽ cần một cơ chế để trao đổi danh tính được xác thực bên ngoài của người dùng để lấy một bộ thông tin xác thực IAM. Có nhiều cách khác nhau để thực hiện điều này, nhưng ở đây sẽ tập trung vào 3 dịch vụ: Amazon Cognito, SAML Providers (SAML- Security Assertion Markup Language), giải pháp đối tác bên thứ ba (third-party partner solutions)

Amazon Cognito

Amazon Cognito cho phép chúng ta dễ dàng tích hợp với nhiều nhà cung cấp xác thực bên thứ ba và đảm nhiệm việc tích hợp với dịch vụ cung cấp mã bảo mật (STS - Security Token Service) của AWS để tìm nạp thông tin xác thực IAM tạm thời. Khi sử dụng Amazon Cognito, chúng ta quản lý người dùng cho ứng dụng của mình bằng nhóm nhận dạng. Điều này cho phép xác định một hoặc nhiều nhà cung cấp định danh sẽ đảm nhiệm việc xác thực người dùng cũng như vai trò IAM mà bất kỳ người dùng được xác thực nào cũng có thể đảm nhận.

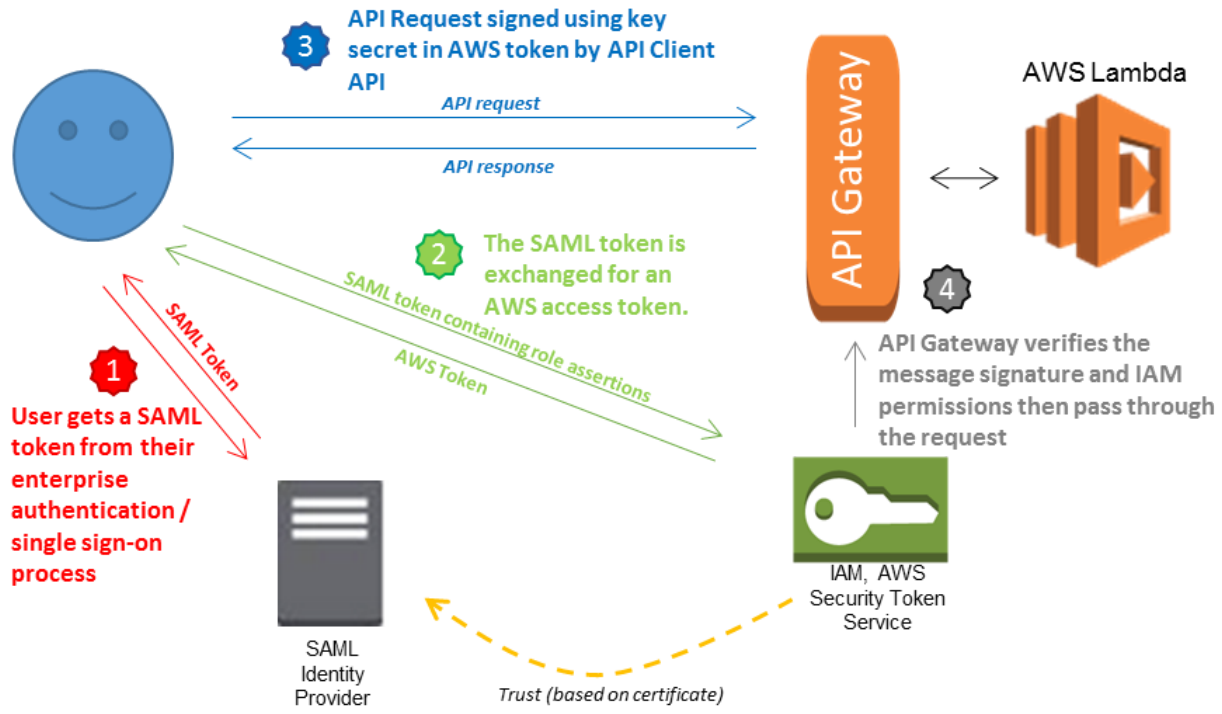
Ví dụ: bạn có thể chọn cho phép người dùng đăng nhập vào ứng dụng của mình Đăng nhập bằng Amazon, Đăng nhập bằng Google+ hoặc tên người dùng và mật khẩu dành riêng cho ứng dụng của bạn. Để triển khai điều này với Amazon Cognito, bạn tạo nhóm nhận dạng mới và định cấu hình Amazon, Google hoặc chương trình hỗ trợ tùy chỉnh làm nhà cung cấp xác thực. Sau đó, bạn có thể sử dụng các phương thức GetId và GetCredentialsForIdentity từ API Amazon Cognito để trao đổi danh tính từ một trong các nhà cung cấp xác thực lấy thông tin đăng nhập IAM cần thiết để thực hiện lệnh gọi API, như thể hiện trong sơ đồ minh họa sau.



SAML Providers

Trong khi Amazon Cognito cho phép chúng ta ánh xạ một vai trò duy nhất tới tất cả người dùng được xác thực trong nhóm nhận dạng, SAML Providers cho phép ánh xạ chi tiết hơn giữa các lớp người dùng và vai trò đó. Giả sử chúng ta có vai trò IAM riêng biệt cho từng người dùng truy cập API của mình và chúng ta cũng có thể duy trì cả danh tính người dùng và vai trò mà họ có quyền truy cập trong chính nhà cung cấp định danh đó.

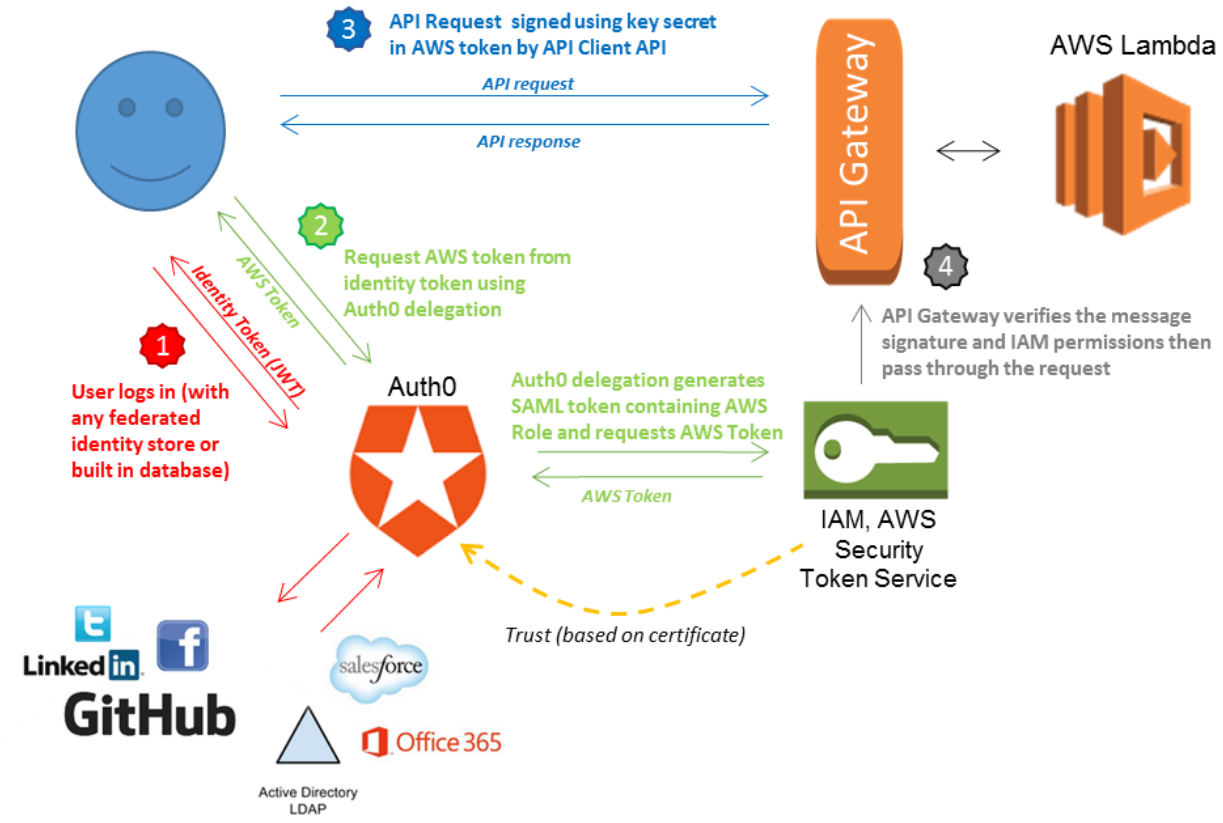
Khi người dùng đăng nhập vào ứng dụng của chúng ta, trước tiên họ sẽ xác thực với SAML Providers và được cấp một SAML Token. SAML Token này bao gồm thông tin về danh tính của người dùng, danh sách các tính năng IAM mà người dùng được phép đảm nhận và chữ ký mật mã chứng minh tính hợp lệ của Token. Sau đó, Token này được đổi lấy thông tin xác thực IAM tạm thời bằng cách sử dụng phương thức AssumeRoleWithSAML từ API AWS STS. Cuối cùng, các yêu cầu được thực hiện đối với API của bạn bằng cách sử dụng các thông tin xác thực này, như thể hiện trong sơ đồ minh họa sau.



Tìm hiểu thêm về SAML 2.0-based , cách tạo tích hợp SAML với AWS Creating IAM Identity Providers và cấu hình xác thực SAML Configuring SAML Assertions for the Authentication Response .

Giải pháp của bên thứ ba (Third-party Solutions)

Ngoài các dịch vụ như Amazon Cognito do AWS cung cấp, còn có một số dịch vụ của bên thứ ba trong hệ sinh thái đối tác AWS rộng lớn hơn có thể giúp bạn quản lý các hoạt động tích hợp phức tạp trên nhiều nhà cung cấp định danh và vai trò IAM. Một đối tác như vậy (Auth0) gần đây đã xuất bản một hướng dẫn hiển thị các phương pháp tích hợp bảo mật khác nhau với API Gateway, bao gồm cả việc sử dụng các quy tắc tùy chỉnh ánh xạ giữa định danh và vai trò của IAM. Điều này cho phép chúng ta cung cấp các cấp truy cập khác nhau cho các lớp người dùng khác nhau dựa trên nhiều thuộc tính. Sơ đồ sau đây cho thấy quá trình tích hợp bằng cách sử dụng ủy quyền Auth0 cho AWS mà hướng dẫn minh họa.



Ủy quyền (Authorization)

Khi đã thiết lập cơ chế cho người dung đăng nhập và lấy thông tin xác thực IAM, chúng ta vẫn phải xác định những thao tác mà người dùng được phép thực hiện.

Ủy quyền cấp phương thức (Method-level Authorization)

Để quản lý quyền truy cập vào các phương pháp riêng lẻ trên API, API Gateway sử dụng thông tin xác thực IAM có được trong giai đoạn xác thực để ủy quyền các yêu cầu đối với từng phương pháp. Mỗi bộ thông tin đăng nhập được liên kết với một bộ chính sách xác định các quyền được cấp cho người dùng. Bằng cách sửa đổi các chính sách này, chúng ta tác động đến các phương thức trên API mà mỗi người dùng có thể truy cập.

Ví dụ có một dữ liệu "product" mà tất cả người dùng đều có thể đọc được nhưng chỉ quản trị viên mới có thể ghi được. Trong trường hợp này, chúng ta có hai vai trò IAM - một cho quản trị viên và một cho người dùng thông thường và sẽ chỉ định các chính sách cho từng vai trò bằng các câu lệnh sau:

```
For regular users:{
    "Effect": "Allow",
    "Action": [ "apigateway:GET" ],
    "Resource": [ "arn:aws:apigateway:us-east-1::my-api-id:/stage/myapi/product" ]
}
```

```
For administrators:{
    "Effect": "Allow",
    "Action": [ "apigateway:GET", "apigateway:POST" ],
    "Resource": [ "arn:aws:apigateway:us-east-1::my-api-id:/stage/myapi/product" ]
}
```

Trong các chính sách IAM kiểm soát quyền truy cập vào API, các phương thức HTTP khác nhau (GET, POST, PUT, v.v.) là các hành động và mỗi tài nguyên RESTful trong API là một tài nguyên có ARN riêng. Bằng cách kết hợp các câu lệnh cho phép hoặc từ chối một tập hợp các phương thức HTTP trên từng tài nguyên API, chúng ta có thể tạo các chính sách cấp quyền truy cập chính xác vào tập hợp các phương thức tối thiểu cần thiết cho mỗi lớp người dùng.

Cấp quyền chi tiết (Fine-grained Permissions)

Đặt quyền ở cấp phương thức cho phép chúng ta kiểm soát bất cứ hành động (actions) nào mà người dùng được phép truy cập dựa trên vai trò của người dùng, nhưng trong một số trường hợp, vẫn cần quản lý kiểm soát chi tiết hơn đối với dữ liệu cụ thể mà người dùng được phép thao tác trên đó . Chẳng hạn, chúng ta có thể cho phép tất cả người dùng được xác thực ĐĂNG (POST) vào tài khoản để sửa đổi chi tiết thông tin, tuy nhiên vẫn hạn chế mỗi người dùng chỉ được phép sửa đổi thông tin tài khoản của chính họ chứ không phải của người khác. Để thực hiện điều này, các dịch vụ chúng ta viết cần có quyền truy cập vào danh tính của người gọi ban đầu. Chúng ta không thể dựa vào một giá trị ban đầu (giá trị thô) trong yêu cầu để xác định người gọi vì bất kỳ người dùng nào có quyền gọi một phương thức đều có khả năng giả mạo danh tính của người khác. Thay vào đó, một giải pháp phổ biến là chuyển Token thu được từ nhà cung cấp xác thực thông qua API tới dịch vụ phụ trợ. Dịch vụ có thể xác thực Token với nhà cung cấp xác thực, sau đó đưa ra quyết định về việc có cho phép hoạt động hay không. Với cách tiếp cận này, chúng ta phải nhúng các điều khiển xác thực chi tiết vào ứng dụng của mình.

Nhìn chung, bảo mật các ứng dụng và API là một chủ đề phức tạp. Hy vọng bài viết này cung cấp một cái nhìn tổng quan tốt về một số phương pháp phổ biến để quản lý quyền truy cập của người dùng vào API bằng API Gateway và IAM. Khuyến khích chúng ta khám phá các tài nguyên khác có sẵn để quản lý bảo mật trên AWS, bao gồm blog bảo mật cá nhân cũng như tài liệu IAM và AWS STS để đảm bảo tuân theo các phương pháp tốt nhất khi nói đến khía cạnh quan trọng của thiết kế và triển khai ứng dụng.

Theo https://aws.amazon.com