ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ

BÀI TẬP NHÓM SỐ 8

BỘ MÔN: KIẾN TRÚC HƯỚNG DỊCH VỤ

LỚP MÔN HỌC: INT3505 21

CHỦ ĐỀ:

Tìm hiểu về an ninh cho API

Version 1.1 Rev. 05

Ngày nộp: 24 tháng 04, năm 2021.

DANH SÁCH THÀNH VIÊN NHÓM:

- LÊ ĐÌNH HOÀNG (MSSV: 17021254)

- NGUYỄN ĐÌNH NHẬT MINH (MSSV: 17021298)

I. Giới thiệu:

- Qua các bài trước, chúng ta đã tìm hiểu về kiến trúc vi dịch vụ. Trong đó, các dịch vụ cần phải giao tiếp với nhau trong hoạt động của toàn hệ thống, các giao tiếp đó được thực hiện bằng cách gọi API (application programming interfaces). Do API đảm nhiệm việc giao tiếp giữa các bộ phận của hệ thống, việc đảm bảo an ninh/bảo mật trong quá trình hoạt động của các API là rất quan trọng trong an ninh hệ thống và vận hành của hệ thống đó.

II. Các nguyên lý an ninh cơ bản của API:

- Phải có cơ chế quản lý và phân quyền truy cập dành cho nhiều đối tượng truy cập khác nhau. Ví dụ như xử lý của máy khách có quyền truy cập khác so với xử lý của máy quản trị viên. Nếu không, các thành phần truy cập không tin cậy vượt quyền hạn cho phép có thể thay đổi các thông số quan trọng của các thành phần trong hệ thống, gây nên lỗi nghiêm trọng, uy hiếp hoạt động của toàn hệ thống.
- Toàn bộ các hoạt động của API đều phải được đảm bảo an toàn, kể cả khi một chuỗi các hoạt động khác nhau của API được thực hiện vẫn phải đảm bảo độ an ninh giống như khi từng API được thực hiện riêng rẽ.
- API cần phải được cài đặt an ninh về mặt kĩ thuật. Một số vấn đề kĩ thuật như kiểm tra và phân quyền truy cập, mã hoá thông tin, hay kiểm tra kích cỡ đầu vào để tránh tấn công DoS,... .
- Bộ ba thuộc tính bảo mật thông tin cơ bản: bao gồm tính bảo mật (Confidentiality), tính toàn vẹn (Integrity) và tính sẵn sàng (availability), gọi tắt là CIA. Chúng được biết đến rộng rãi là bộ ba bảo mật thông tin, là ba yếu tố chính được sử dụng trong việc đo điểm chuẩn bảo mật hệ thống thông tin. Bộ ba CIA giúp thiết kế mô hình bảo mật và đánh giá sức mạnh của mô hình bảo mật hiện có.
+ Tính bảo mật: quan tâm về cách bảo vệ dữ liệu khỏi những người nhận ngoài ý muốn. Thuộc tính bảo mật nói tới việc bảo vệ các kênh vận chuyển và lưu trữ bằng mã hóa.
+ Tính toàn vẹn: là sự đảm bảo về tính đúng đắn và đáng tin cậy của dữ liệu cũng như khả năng phát hiện bất kỳ những sự thay đổi trái phép đã xảy ra. Nó đảm bảo rằng dữ liệu được bảo vệ khỏi sự thay đổi, sửa đổi hoặc xóa một cách trái phép hoặc đơn giản là do sơ ý.
+ Tính sẵn sàng: làm cho một hệ thống luôn sẵn sàng để người dùng hợp pháp có thể truy cập mọi lúc là mục tiêu cuối cùng của bất kỳ thiết kế hệ thống nào. Mục tiêu của thiết kế bảo mật phải là làm cho hệ thống có tính khả dụng cao bằng cách bảo vệ nó khỏi các nỗ lực truy cập bất hợp pháp, mặc dù làm như vậy là vô cùng khó khăn. Các cuộc tấn công, đặc biệt là trên một API công khai, có thể khác nhau, từ kẻ tấn công đưa phần mềm độc hại trong hệ thống đến tấn công từ chối dịch vụ phân tán (DDoS) có tổ chức cao.
- Cần phải xét các yêu cầu và phạm vi của hệ thống để xác định các cơ chế an ninh thích hợp.

III. Thiết kế cơ chế an ninh:

a. Để thiết kế chiến lược sử dụng cơ chế an ninh phù hợp với hệ thống, cần phải dựa vào các tiêu chí sau:

+ Xác định đúng bộ phận cần bảo vệ. Cần xác định đúng những bộ phận chức năng và dữ liệu cần được bảo vệ sao cho phù hợp với đặc thù hệ thống. Ví dụ đối với hệ thống mạng xã hội, thông tin về người dùng có thể được nhìn thấy bởi những người dùng khác để kết bạn và giao lưu, nhưng đối với hệ thống bán hàng điện tử thì điều này là hoàn toàn cấm kỵ. Ngoài ra cũng cần phải bảo vệ cả về phần cứng lẫn phần mềm logic, bởi nếu kẻ phá hoại xâm nhập vật lý phần cứng máy chủ chúng có thể từ đó cài hay phá hoại dữ liệu một cách trực tiếp. Nhìn chung, bất kỳ bộ phận nào của hệ thống mà khi nó bị thâm nhập sai cách sẽ gây hại (ảnh hưởng tiêu cực) tới người dùng hệ thống, thì các hệ thống đó là những bộ phận cần được bảo vệ.
+ Cần xác định rõ mục tiêu an ninh của hệ thống. Các mục tiêu an ninh của hệ thống chính là các thuộc tính chất lượng liên quan tới an ninh của hệ thống. Nó xác định cách hệ thống nên được bảo vệ theo những tiêu chí gì và như thế nào. Ví dụ như thông tin chỉ có thể được đọc bởi đối tượng nhất định (tính bảo mật), ... Với các thuộc tính trừu tượng này, rất khó để có thể tạo test cases và kiểm soát chất lượng cũng như thẩm định sản phẩm. Vì vậy, các yêu cầu này cần phải được làm mịn hơn nữa. Ví dụ thay vì tính bảo mật chung chung, ta sử dụng yêu cầu là người dùng cần phải đăng nhập vào hệ thống để sử dụng, và người dùng cần phải kết bạn với ngừi dùng khác để xem các thông tin cá nhân của người đó. Vì vậy, việc xác định các yêu cầu bảo mật và làm mịn chúng cùng với các yêu cầu phi chức năng khác là rất quan trọng.
+ Lựa chọn các cơ chế bảo mật để đạt được những yêu cầu bảo mật trên. Các cơ chế tiêu biểu sẽ được mô tả trong ý b.
+ Môi trường hoạt động của API và các nguy cơ trong môi trường đó, nhằm mô hình hoá các mối nguy tiềm ẩn. Trong mỗi môi trường hoạt động đều tiềm ẩn rất nhiều mối nguy cơ tiềm tàng, nhưng không phải bất kỳ mối nguy cơ nào cũng lớn hay thực sự quan trọng. Ví dụ đối với hệ thống bán bánh mì online, việc các crawler của các trang phân tích web đi dò tìm trên hệ thống không phải là vấn đề lớn, nhất là khi hệ thống khá đơn giản, tốn ít tài nguyên và không lưu dữ liệu người dùng. Các mối nguy tiềm tàng được đánh giá là nguy hiểm sẽ được cho vào tập mối đe doạ để lập kế hoạch giải quyết và phòng tránh. Đây gọi là mô hình hoá mô nguy hiểm. Nguy cơ từ các mối hoạ thuộc mỗi nhóm trên thường có thể được ngăn chặn theo nhóm nguy cơ, ví dụ như yêu cầu tất cả người dùng phải đăng nhập có thể loại bỏ nguy cơ mạo danh và sửa đổi dữ liệu trái phép.

b. Các cơ chế bảo mật:

- Các cơ chế bảo mật được sử dụng để thực hiện các yêu cầu về bảo mật. Một số các cơ chế phổ biến:
+ Cơ chế mã hoá: đảm bảo dữ liệu không thể bị thay đổi bởi bất kì bên nào không có quyền hạn, trong cả quá trình truyền tin lẫn trong hệ thống cơ sở dữ liệu. Mã hoá cũng đảm bảo thông tin không thể bị thay đổi bởi những kẻ tấn công. Đối với các API, chúng ta có thể sử dụng bảo mật lớp truyền tải (TLS), hay còn được gọi là HTTPS. Để lưu trữ, chúng ta có thể sử dụng mã hóa cấp đĩa (disk-level encryption) hoặc mã hóa cấp ứng dụng (application-level encryption). Để truyền tải tin nhắn, chúng ta nên dùng mã hóa cấp độ tin nhắn, để có thể sử dụng ngay cả một kênh không an toàn (như HTTP).

+ Cơ chế xác thực: đảm bảo người truy cập không bị mạo danh. Đơn giản nhất của xác thực là sử dụng username và password, ngoài ra hiện tại còn có thể có nhiều hơn 1 bước xác thực bằng cách bổ sung các phương thức khác như xác thực bằng chìa khoá vật lý, bằng thiết bị ngoại vi, sinh trắc học,... .
+ Cơ chế kiểm soát và uỷ quyền truy cập: để đảm bảo tính toàn vẹn và tính bảo mật của các bộ phận hệ thống đối với từng đối tượng truy cập, cụ thể là kiểm soát ai có thể truy cập vào phần nào và có thể có những thao tác như thế nào. Có hai loại kiểm soát truy cập: dựa trên danh tính người dùng và dùng token để truy cập.
+ Cơ chế lưu vết: để lưu lại các hành động đã được thực hiện với API. Lưu vết cho phép nhà phát triển gỡ lỗi dễ dàng hơn khi có lỗi xảy ra, hay sử dụng các công cụ theo dõi thời gian thực để tiên đoán trước lỗi sẽ xảy đến.
+ Cơ chế giới hạn tài nguyên: để tránh việc tấn công DoS, khiến cho toàn bộ dịch vụ không thể truy cập được bởi các dịch vụ khác đang cần nó. Để làm điều này, thông thường từng nhóm truy cập sẽ bị giới hạn theo lượng truy cập theo thời gian, thời gian truy cập, lưu lượng, ...
+ Cơ chế chống thoái thác (nonrepudiation): Bất cứ khi nào người dùng thực hiện một giao dịch thông qua API bằng cách chứng minh danh tính của mình thì từ đó trở đi, thông tin của người dùng sẽ không thể bị thay đổi để phục vụ cho việc xác thực sau này. Việc chống thoái thác phải cung cấp bằng chứng về nguồn gốc và tính toàn vẹn của dữ liệu.

IV. Một số cách để đảm bảo an ninh cho API:

a. Nhận dạng được lỗ hổng:

- Cách duy nhất để bảo mật API hiệu quả là biết phần nào trong vòng đời của API không an toàn. Điều này không phải dễ để làm vì phải xem xét toàn bộ vòng đời của API.

b. Tận dụng OAuth:

- Một trong những khía cạnh quan trọng nhất của bảo mật API là kiểm soát truy cập để xác thực và ủy quyền. Một công cụ mạnh mẽ để kiểm soát quyền truy cập API là OAuth, một khung ủy quyền dựa trên mã thông báo (token-based authorization) cho phép các dịch vụ của bên thứ ba truy cập thông tin mà không làm lộ thông tin đăng nhập của người dùng.

c. Mã hóa dữ liệu:

- Tất cả dữ liệu, đặc biệt là dữ liệu nhận dạng cá nhân, phải được mã hóa bằng phương pháp như Bảo mật tầng truyền tải (TLS). Các nhà phát triển cũng nên yêu cầu chữ ký để đảm bảo rằng chỉ những người dùng được ủy quyền mới được giải mã và sửa đổi dữ liệu.

d. Phát triển một mô hình mối đe dọa:

- Mô hình hóa mối đe dọa là một cách tiếp cận có cấu trúc để xác định và đánh giá rủi ro. Các mô hình mối đe dọa được sử dụng như một biện pháp phòng ngừa, nhưng chúng cũng nên được coi là một chu trình liên tục để đánh giá, giảm thiểu và ngăn chặn các lỗ hổng ứng dụng theo cách tự động nhưng vẫn được kiểm soát.