Integralność komunikacji API

Laboratorium 6

Wymagania

1. Środowisko programistyczne umożliwiające przygotowanie aplikacji w frameworku Spring Boot. Alternatywnie dowolne środowisko programistyczne umożliwiające przygotowanie aplikacji implementującej interfejs REST API.

Przydatne źródła

1. [link] – RFC 7515 opisujące JSON Web Signature
2. [link] – Krótki opis struktury JWS
3. [link] – HMAC – SHA256 Online Generator Tool

Wstęp

Projektowanie komunikacji pomiędzy usługami wchodzącymi w skład systemu zakłada także zaplanowanie sposób weryfikacji integralności komunikatów. Zapewnienie usługi bezpieczeństwa polegającej na możliwości weryfikacji, czy otrzymane żądanie nie zostało zmodyfikowane od momentu wysyłki przez aplikację kliencką jest możliwe zarówna w przypadku interfejsów SOAP i REST. Mechanizmy te sprowadzają się do wykonania podpisu parametrów zapytania, przy użyciu kryptografii symetrycznej lub asymetrycznej i przekazania wartości podpisu razem z komunikatem.

Zadanie 1

1. Do przygotowanych interfejsów dodać nagłówki umożliwiające przekazanie wartości podpisu komunikatu. Dla zapytań POST przyjmiemy, że podpis przekazywany będzie w nagłówku X-HMAC-SIGNATURE, dla zapytań PUT należy dodać obsługę parametry X-JWS-SIGNATURE

Zadanie 2

1. Dla zapytania POST przygotować implementację weryfikującą podpis przekazanego zapytania.
2. Obsługiwany format podpisu zakłada wykorzystanie metody HMAC SHA 256 przekazywany jako plain text.
3. Sekret wykorzystywany do generowania wartości HMAC SHA 256 to „123456”.
4. Przykładowa wartość HMAC SHA 256 może zostać wygenerowana przy wykorzystaniu narzędzia wskazanego jako link nr 3 na początku instrukcji.
5. Wygenerowana wartość powinna zostać przekazana jako nagłówek X-HMAC-SIGNATURE

6. Przygotowana implementacja powinna odczytać wartość nagłówka X-HMAC-SIGNATURE oraz JSON przekazany w zapytaniu, a następnie dokonać weryfikacji czy przekazane żądanie nie zostało zmodyfikowane.
7. Błędna wartość podpisu powinna zwracać odpowiedni błąd, wskazujący że zapytanie nie może być przetworzone ze względu na problem z weryfikacją integralności zapytania.

Zadanie 3

1. Dla zapytania PUT przygotować implementację weryfikującą podpis przekazanego zapytania.
2. Obsługiwany format podpisu zakłada wykorzystanie metody JWS w dowolnie wybranym trybie np. detached.
3. JWS może korzystać z dowolnie wybranego zestawu algorytmów:
    a. HMAC + SHA256
    b. RSASSA-PKCS1-v1_5 + SHA256
    c. ECDSA + P-256 + SHA256
4. W zależności od wybranego formatu program powinien dokonywać weryfikacji w oparciu o sekret lub klucz publiczny, komplementarny do klucza prywatnego, wykorzystanego przy generowaniu JWS.
5. W celu przygotowania wartości podpisu JWS powinien zostać przygotowany mini-program umożliwiający wygenerowania tokenu JWS. Alternatywnie można skorzystać z generatora online. **UWAGA!** W sprawozdaniu należy szczegółowo opisać sposób przeprowadzenia testu API - jak uruchomić przygotowaną metodę/program lub na podstawie jakiego generatora online, sekretu, klucza prywatnego można utworzyć odpowiedni token JWS, który po przekazaniu jako nagłówek umożliwi poprawną weryfikację.
6. Przygotowana implementacja powinna odczytać wartość nagłówka X-JWS-SIGNATURE oraz JSON przekazany w zapytaniu, a następnie dokonać weryfikacji czy przekazane żądanie nie zostało zmodyfikowane.
7. Błędna wartość podpisu powinna zwracać odpowiedni błąd, wskazujący że zapytanie nie może być przetworzone ze względu na problem z weryfikacją integralności zapytania.

Sprawozdania wraz z kodem źródłowym należy dostarczyć do pierwszej środy po zajęciach, do godziny rozpoczęcia zajęć.