Bezpieczeństwo komputerowe

Laboratorium - lista nr 7

Zadanie 1 (100 pkt) Zmodyfikuj (jeżeli to konieczne) działanie serwisu bankowego. Dodaj funkcję logowania dla administratora, który ma możliwość zatwierdzania przelewów – taki przelew klientowi będzie się pokazywał jako zrealizowany.

Przeprowadź ataki SQL Injection, XSS i XSRF na swój serwis. W szczególności dla SQL injection: przygotuj następujące zapytania:

1. umożliwiające obejrzenie danych, do których nie powinniśmy mieć dostępu (np. dane dotyczące przelewów zleconych przez innego klienta);
2. zatwierdzające zlecony przelew (pomimo tego, że nie jesteśmy administratorami serwisu);
3. pokazujące pliki w systemie operacyjnym (jeżeli wykorzystana baza danych ma takie funkcje/uprawnienia).

Dla XSS, XSRF przygotuj kod, który spowoduje wykonanie operacji zatwierdzenia przelewów przez administratora.

Celem zadania jest uświadomienia sobie zagrożeń związanych z tymi typami ataków. Jeżeli Twoja strona została przygotowana we frameworku, który automatycznie zabezpiecza przed atakami typu XSS, XSRF, ... to na potrzeby zadania wyłącz te zabezpieczenia (w większości przypadków można to bezproblemowo zrobić w plikach konfiguracyjnych).

Zadanie 2 (20 pkt) Zmodyfikuj działanie serwisu bankowego. Wygeneruj dla serwera certyfikat TLS dla domeny: www.mojWspanialyBank.com (ale może to być inny adres). (TLS Client Authentication) Wygeneruj certyfikat użytkownika, który będzie można zainstalować w przeglądarce. Skonfiguruj serwer www w ten sposób, aby zezwalał na połączenie jedynie użytkownikom, którzy przedstawią odpowiedni certyfikat.

Zadanie 3 (30 pkt) Zmodyfikuj działanie serwisu bankowego, aby udostępniał REST API, które będzie oferowało dokładnie tę samą funkcjonalność, co strona. Klient korzystający z API musi korzystać z certyfikatu TLS.

Poprawność działania przedstaw za pomocą programu curl. Aby klient przedstawił swój certyfikat, wykorzystaj opcje np: curl -X POST –header Ćontent-Type: application/json--cacert ./keys/ca.pem –key ./keys/mykey.pem –cert ./keys/mycert.pem –data '{"user":"123123", "password":ąsdfgh"}' https://www.mojwspanialybank.com/api/v1/login

Zadanie 4 (50 pkt) Klient api, po poprawnym zalogowaniu ma otrzymać klucz (do HS256), który ma wykorzystywać do uwierzytelniania kolejnych żądań. Zmodyfikuj działanie serwisu/api – wykorzystaj tokeny JWT (RFC 7519, `https://jwt.io/`).