Poniżej znajduje się raport przygotowany na podstawie dostarczonych materiałów, omawiający kluczowe zagadnienia z zakresu bezpieczeństwa aplikacji webowych i protokołów komunikacyjnych.

1. TLS (Transport Layer Security)

Protokół TLS jest fundamentem bezpiecznej komunikacji w Internecie, zapewniającym realizację celów **triady CIA** oraz uwierzytelniania 1, 2.

- **Poufność:** Szyfrowanie danych przesyłanych przez sieć chroni je przed odczytem przez osoby nieuprawnione 3, 4.
- **Integralność:** Mechanizmy kryptograficzne (funkcje skrótu i podpisy) gwarantują, że informacje nie zostały zmodyfikowane podczas transmisji 4-6.
- **Uwierzytelnianie serwera (Certyfikaty):** Proces weryfikacji tożsamości serwera opiera się na certyfikatach cyfrowych, które pełnią rolę „cyfrowego paszportu” 7, 8.
- **Łańcuch zaufania:** Zaufanie w systemie PKI jest hierarchiczne 9. Proces weryfikacji przebiega od **certyfikatu serwera**, przez **certyfikaty pośrednie (Intermediate CA)**, aż do zaufanego **certyfikatu głównego (Root CA/Host CA)**, który jest bezpośrednio zaufany przez system lub przeglądarkę 9, 10.
- **Handshake:** Jest to proces ustanawiania bezpiecznej sesji, podczas którego strony negocjują algorytmy, wymieniają klucze i uwierzytelniają się nawzajem 1, 11, 12.
- **Zawartość certyfikatu X.509:** Zgodnie ze standardem, certyfikat zawiera m.in. 10, 13:
- **Domeny:** Nazwa podmiotu (Subject/CN) oraz alternatywne nazwy (SAN).
- **Klucz publiczny:** Wraz z informacją o algorytmie.
- **Datę ważności:** Okresy „not before” i „not after”.
- **Wystawcę:** Nazwę urzędu certyfikacji (CA), który podpisał dokument.

2. HTTP (Hypertext Transfer Protocol)

W oparciu o źródła i wiedzę uzupełniającą (struktura zapytań nie jest szczegółowo opisana w źródłach), protokół HTTP stanowi warstwę komunikacyjną aplikacji 1, 14.

- **Model bezstanowy i porty:** HTTP z założenia jest bezstanowy, co oznacza, że każde żądanie jest traktowane niezależnie (informacja zewnętrzna). Standardowo korzysta z **portu 80**, natomiast jego bezpieczna wersja (HTTPS) wykorzystuje **port 443** 1, 8.
- **Struktura żądania (Informacja zewnętrzna):** Składa się z linii żądania (metoda, adres, wersja), nagłówków oddzielonych nową linią (CRLF) oraz opcjonalnego ciała (body).
- **Metody HTTP:**
- **GET:** Pobieranie danych 15.
- **POST:** Wysyłanie danych do serwera (np. w formularzach) 16, 17.
- **PUT / PATCH:** Aktualizacja zasobów (informacja zewnętrzna).
- **DELETE:** Usuwanie zasobów (informacja zewnętrzna).
- **HEAD / OPTIONS:** Pobieranie samych nagłówków lub sprawdzanie dostępnych metod (informacja zewnętrzna).
- **Kody odpowiedzi (Informacja zewnętrzna):**
- **2xx (Success):** Żądanie zakończone sukcesem.
- **5xx (Server Error):** Błąd po stronie serwera.

3. XSS (Cross-Site Scripting)

Atak polegający na wstrzyknięciu złośliwego kodu JavaScript do przeglądarki użytkownika 18.

- **Rodzaje ataków XSS:**
- **Stored (Trwały):** Kod jest trwale zapisany na serwerze (np. w bazie danych w komentarzu) i wykonuje się u każdego, kto wyświetli daną stronę 19, 20.
- **Reflected (Odbity):** Kod jest „odbijany" od serwera (np. przez parametr w URL) i wymaga nakłonienia ofiary do kliknięcia w link 21, 22.
- **DOM-based:** Atak odbywa się wyłącznie po stronie klienta; skrypt modyfikuje środowisko DOM bez udziału serwera 23, 24.
- **Skutki:** Kradzież ciasteczek sesyjnych, przejęcie sesji, modyfikacja treści strony, phishing oraz rozprzestrzenianie złośliwego oprogramowania 25, 26.
- **Ochrona:**
- **Walidacja wejść i sanityzacja:** Sprawdzanie, czy dane są zgodne z oczekiwanym wzorcem 27, 28.
- **Kodowanie wyjścia (Escapowanie):** Zamiana znaków specjalnych na encje HTML (np. użycie htmlspecialchars) 29, 30.
- **Unikanie innerHTML:** Zastąpienie go bezpieczniejszymi metodami, takimi jak textContent lub createTextNode, które nie interpretują tekstu jako kodu HTML 31, 32.

4. SQL Injection

Technika polegająca na wstrzykiwaniu kodu SQL do zapytań bazy danych poprzez nieprawidłowo zabezpieczone pola wejściowe 33.

- **Mechanizm:** Atakujący wykorzystuje brak rozróżnienia między płaszczyzną danych a płaszczyzną sterowania w SQL, co pozwala na manipulację logiką zapytania (np. poprzez dodanie ' OR 1=1 --) 34, 35.
- **Ochrona:** Stosowanie **zapytań sparametryzowanych** (prepared statements), korzystanie z systemów **ORM** oraz walidacja typów danych 36, 37.

5. Typy ataków w narzędziach testowych (Informacja zewnętrzna)

Terminy takie jak *Sniper* czy *Pitchfork* odnoszą się do trybów działania narzędzi klasy proxy (np. Burp Suite Intruder), które służą do automatyzacji ataków XSS i SQLi:

- **Sniper:** Jeden zestaw payloadów wstrzykiwany po kolei w każde zdefiniowane miejsce.
- **Battering Ram:** Ten sam payload wstrzykiwany jednocześnie we wszystkie miejsca.
- **Pitchfork:** Wiele zestawów payloadów wstrzykiwanych równolegle w różne miejsca (połączone indeksem).
- **Cluster Bomb:** Testowanie wszystkich możliwych kombinacji wielu zestawów payloadów.

Czy potrzebujesz dodatkowych informacji na temat konkretnych metod ochrony, takich jak polityka **Content Security Policy (CSP)**, która jest istotnym elementem obrony przed XSS 38, 39?