Wymagania bezpieczeństwa dla aplikacji www

Departament Bezpieczeństwa

Spis treści

1. Cel dokumentu

Niniejszy dokument zawiera wytyczne bezpieczeństwa na poziomie architektury i konfiguracji aplikacji www w środowisku teleinformatycznym CeZ (Centrum e-Zdrowia) lub MZ (Ministerstwo Zdrowia). Celem dokumentu jest usystematyzowanie wiedzy dotyczącej bezpieczeństwa teleinformatycznego systemu z uwzględnieniem architektury systemu, obsługiwanych protokołów kryptograficznych oraz powiązań z procesami biznesowych i funkcjonalnymi.

1.1 Metryka dokumentu

Osoba odpowiedzialna	Łukasz Kucharzewski
Wersja dokumentu	0.9
Data ostatniej modyfikacji	26.04.2021
Status dokumentu	*Zatwierdzony*

1.2 Historia zmian

Status: Robocza/Do weryfikacji/Zatwierdzony

Wersja	Data	Osoba modyfikująca	Opis modyfikacji	Status
0.9	23.04.2021	Łukasz Kucharzewski	Utworzenie projektu dokumentu	Zatwierdzony

Centrum e-Zdrowia
ul. Stanisława Dubois 5A
00-184 Warszawa
tel.: +48 22 597-09-27
fax: +48 22 597-09-37
biuro@cez.gov.pl | www.cez.gov.pl
NIP: 5251575309
REGON: 001377706

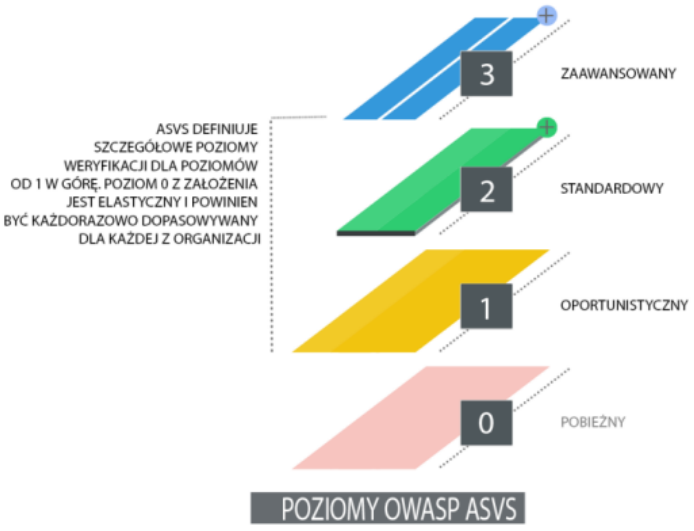
2. Wprowadzenie

2.1 Metodologia

Bezpieczeństwo aplikacji www musi być zgodne z wymaganiami zdefiniowanymi w standardzie OWASP (Open Web Application Security Project) Application Security Verification Standard w wersji 4.0.2 rozumianych jako szereg wytycznych podczas tworzenia listy kontrolnej bezpiecznego kodowania specyficznej dla aplikacji, platformy lub organizacji. Dostosowanie ASVS do konkretnych przypadków zwiększa presję na stosowanie wymogów bezpieczeństwa, które są najważniejsze dla projektów realizowanych w CeZ/MZ.

Application Security Verification Standard definiuje trzy poziomy weryfikacji bezpieczeństwa, w ramach każdego z poziomów zwiększa się jej głębokość.

1. ASVS Poziom 1 jest przeznaczony dla wszystkich aplikacji,
2. ASVS Poziom 2 jest przeznaczony dla aplikacji, które zawierają dane wymagające ochrony,
3. ASVS Poziom 3 jest przeznaczony dla najbardziej krytycznych aplikacji - takich, które wykonują transakcje znaczniej wartości, zawierają wrażliwe dane medyczne, a także innych aplikacji, które wymagają najwyższego poziomu zaufania.



Rysunek 1 Poziomy OWASP ASVS

Każdy poziom ASVS zawiera listę wymagań bezpieczeństwa. Każde z tych wymagań może być mapowane na specyficzne cechy i właściwości dotyczące zabezpieczeń, które powinny zostać wbudowane w oprogramowanie przez deweloperów aplikacji.

Centrum e-Zdrowia
ul. Stanisława Dubois 5A
00-184 Warszawa
tel.: +48 22 597-09-27
fax: +48 22 597-09-37
biuro@cez.gov.pl | www.cez.gov.pl
NIP: 5251575309
REGON: 001377706

W celu poprawnego wybrania poziomu zabezpieczeń wymagana jest klasyfikacja informacji dla danego systemu.

Ze względu na specyfikę aplikacji oraz sektora działalności, dla aplikacji www wykorzystywanych w środowisku teleinformatycznym CeZ lub MZ przyjęto następujące wytyczne:

1. **Poziom 2 ASVS jest domyślnie wymagany dla wszystkich aplikacji webowych w CeZ/MZ jako poziom minimalny.**

Aplikacja osiąga Poziom 2 ASVS (Standardowy) w przypadku, gdy właściwie chroni przed większością obecnych ryzyk dotyczących oprogramowania. Poziom 2 zapewnia, że mechanizmy bezpieczeństwa są wdrożone, są skuteczne oraz są wykorzystywane wewnątrz aplikacji. Poziom 2 jest zazwyczaj właściwy dla aplikacji, które obsługują istotne transakcje biznesowe - włączając w to przetwarzanie informacji medycznych, informacji, które realizują funkcje wrażliwe lub krytyczne dla biznesu, a także, które przetwarzają inne zasoby wymagające ochrony.

2. **Poziom 3 ASVS jest wymagany dla wszystkich aplikacji w CeZ/MZ, które przetwarzają dane wrażliwe (prawnie chronione), w szczególności dane medyczne.**

Poziom 3 jest najwyższym poziomem weryfikacji w ramach ASVS. Jest zazwyczaj zarezerwowany dla aplikacji, które wymagają szczególnego poziomu weryfikacji OWASP Application Security Verification Standard 4.0.2 mechanizmów bezpieczeństwa, takich jak te, które można znaleźć w zastosowaniach ochronie zdrowia i bezpieczeństwa publicznego, ochronie infrastruktury krytycznej itp. Organizacje mogą wymagać zastosowania Poziomu 3 ASVS dla aplikacji, które realizują funkcje krytyczne, w przypadkach, gdy awaria może mieć znaczący wpływ na działalność organizacji, a nawet na jej przetrwanie. Aplikacja osiąga Poziom 3 ASVS (Zaawansowany) jeśli odpowiednio chroni przed zaawansowanymi podatnościami w zabezpieczeniach aplikacji, a także demonstruje zasady dobrego projektowania zabezpieczeń. Aplikacja znajdująca się na Poziomie 3 ASVS wymaga bardziej dogłębnej analizy, architektury, kodowania i testowania niż na pozostałych poziomach. Bezpieczna aplikacja w znacznej mierze składa się z modułów (w celu ułatwienia np. jej elastyczności, skalowalności a przede wszystkim stosowania kolejnych warstw zabezpieczeń) a każdy moduł (wydzielony w sposób fizyczny i/lub wykorzystując połączenia sieciowe) dba o poprawność własnych wymagań bezpieczeństwa (tzw. wielowarstwowa ochrona - ang. *defence in depth*), które muszą być następnie właściwie udokumentowane. Wymagania obejmują zabezpieczenia zapewniające poufność (np. szyfrowanie), integralność (np. transakcje, walidacja wejście), dostępność (np. sprawna obsługa obciążeń - ang. *handling load gracefully*), uwierzytelnianie (w tym między systemami), niezaprzeczalność, autoryzację i audyt (logowanie).

Centrum e-Zdrowia
ul. Stanisława Dubois 5A
00-184 Warszawa
tel.: +48 22 597-09-27
fax: +48 22 597-09-37
biuro@cez.gov.pl | www.cez.gov.pl
NIP: 5251575309
REGON: 001377706

UWAGA: Dla aplikacji nie przetwarzających danych wrażliwych (dane osobowe, dane medyczne itp.) możliwe jest zastosowanie Poziomu 1 ASVS. Wymagana jest jednak dodatkowa analiza ryzyka dla aplikacji oraz klasyfikacja informacji przetwarzanych przez aplikację.

3. Bezpieczeństwo aplikacji

3.1 Wytyczne bezpieczeństwa dla ASVS

Poniżej zdefiniowano najważniejsze wytyczne bezpieczeństwa wyznaczone przez standard ASVS, które muszą być spełnione przy budowaniu bezpiecznej aplikacji i towarzyszących jej komponentów. Na potrzeby interpretacji wymogów posłużono się specyfikacją RFC 2119.

L.P	Wymóg i Opis	Poziom 1	Poziom 2	Poziom 3
1	**Wymagania weryfikacji dla architektury, projektowania i modelowania zagrożeń** Upewnij się, że weryfikowana aplikacja w odpowiedni sposób spełnia poniższe ogólne wymagania: • dla poziomu pierwszego (1), wszystkie komponenty aplikacji są zidentyfikowane i istnieje powód dla których zostały tam umieszczone, • dla poziomu drugiego (2), architektura została zdefiniowana a kod jest zgodny z architekturą, dla poziomu trzeciego (3), istnieje zdefiniowana architektura oraz dokumentacja projektowa i jest ona wykorzystywana w sposób skuteczny.			
1.1	Zweryfikuj, czy wszystkie komponenty aplikacji są zidentyfikowane, są znane i są rzeczywiście potrzebne.	MUST	MUST	MUST
1.2	Zweryfikuj, czy wszystkie komponenty są zidentyfikowane takie jak biblioteki, moduły i zewnętrzne systemy, które nie są częścią aplikacji ale są wymagane do jej działania. Zweryfikuj, czy wszystkie komponenty są zidentyfikowane takie jak biblioteki, moduły i zewnętrzne systemy, które nie są częścią aplikacji ale są wymagane do jej działania.	OPTIONAL	MUST	MUST
1.3	Zweryfikuj, czy została zdefiniowana wysoko-poziomowa architektura aplikacji.	OPTIONAL	MUST	MUST
1.4	Zweryfikuj, czy komponenty aplikacji są zdefiniowane w kontekście funkcji biznesowych oraz funkcji bezpieczeństwa, które dostarczają.	OPTIONAL	OPTIONAL	MUST
1.5	Zweryfikuj, czy wszystkie komponenty niebędące częścią aplikacji, choć będące przez nią wykorzystywane, są	OPTIONAL	OPTIONAL	MUST

Centrum e-Zdrowia
ul. Stanisława Dubois 5A
00-184 Warszawa
tel.: +48 22 597-09-27
fax: +48 22 597-09-37
biuro@cez.gov.pl | www.cez.gov.pl
NIP: 5251575309
REGON: 001377706

	zdefiniowane pod kątem funkcjonalności - w tym funkcjonalności bezpieczeństwa, które dostarczają			
1.6	Zweryfikuj, czy model oceny ryzyka dla aplikacji został utworzony i czy pokrywa wszystkie ryzyka w ramach STRIDE (Podszywanie się - Spoofing, Modyfikacja danych - Tampering, Zaprzeczalność - Repudiation, Wyciek informacji - Information Disclosure i Eskalacja uprawnień - Privileges Escalation).	OPTIONAL	OPTIONAL	MUST
1.7	Zweryfikuj, czy wszystkie zabezpieczenia (włącznie z bibliotekami komunikującymi się z zewnętrznymi usługami bezpieczeństwa) są wdrażane w sposób scentralizowany.	OPTIONAL	OPTIONAL	MUST
1.8	Zweryfikuj, czy komponenty są odseparowane od siebie poprzez zdefiniowane zabezpieczenia, takie jak segmentacja sieci, reguły zapory sieciowej czy grupy dostępu w usługach chmurowych.	OPTIONAL	MUST	MUST
1.9	Zweryfikuj, czy aplikacja ma jasno zdefiniowany rozdział pomiędzy warstwą danych, warstwą kontrolera i warstwą prezentacyjną, tak aby decyzje dotyczące bezpieczeństwa były wymuszane na zaufanych systemach.	OPTIONAL	MUST	MUST
1.10	Zweryfikuj, czy w kodzie aplikacji klienckiej nie ma wrażliwych informacji z zakresu logiki biznesowej, sekretnych kluczy i innych zastrzeżonych informacji.	OPTIONAL	MUST	MUST
1.11	Zweryfikuj czy wszystkie komponenty aplikacji, biblioteki, moduły, struktura ramowa, platformy i systemy operacyjne są wolne od znanych podatności.	OPTIONAL	MUST	MUST
2	**Wymagania weryfikacji dla uwierzytelniania** Uwierzytelnianie to akt ustanowienia lub potwierdzenia, że coś (ktoś) jest autentyczne i że potwierdzenia dokonane przez lub dla tej rzeczy są prawdziwe. Upewnij się, że weryfikowana aplikacja w odpowiedni sposób spełnia poniższe ogólne wymagania: • weryfikuje cyfrową tożsamość nadawcy w trakcie komunikacji. zapewnia, że tylko upoważnione podmioty mogą się uwierzytelnić, a dane uwierzytelniające są transportowane w sposób bezpieczny.			
2.1	Zweryfikuj, czy wszystkie strony oraz zasoby standardowo wymagają uwierzytelnienia, za wyjątkiem tych, które mają być dostępne publicznie (zasada pełnej mediacji).	MUST	MUST	MUST
2.2	Zweryfikuj, czy pola zawierające dane uwierzytelniające nie są uzupełniane przez aplikację. Propozycje uzupełniania realizowane przez aplikację oznacza, że są one przechowywane czystym tekstem lub w odwracalnym formacie, co jest zabronione.	MUST	MUST	MUST

Centrum e-Zdrowia
ul. Stanisława Dubois 5A
00-184 Warszawa
tel.: +48 22 597-09-27
fax: +48 22 597-09-37
biuro@cez.gov.pl | www.cez.gov.pl
NIP: 5251575309
REGON: 001377706

2.3	Zweryfikuj, czy wszystkie mechanizmy uwierzytelniania są wymuszane po stronie serwera.	MUST	MUST	MUST
2.4	Zweryfikuj, czy wszystkie mechanizmy uwierzytelniania, które kończą pracę niepowodzeniem, robią to w sposób bezpieczny, aby mieć pewność, że atakujący nie może się zalogować.	MUST	MUST	MUST
2.5	Zweryfikuj, czy pole dla hasła zezwala lub sprzyja używaniu haseł (passphrase) i nie ma wprowadzonych ograniczeń chroniących przed korzystaniem z menedżerów haseł oraz wpisywaniem długich lub bardzo złożonych haseł.	MUST	MUST	MUST
2.6	Zweryfikuj, czy wszystkie operacje dotyczące uwierzytelniania (takie jak rejestracja, aktualizacja profilu, przypomnienie loginu, przypomnienie hasła, unieważnienie zgubionego kodu do telefonicznej obsługi klienta lub automatycznej obsługi klienta), które powodują odzyskanie dostępu do konta, posiadają przynajmniej takie same zabezpieczenia jak podstawowe mechanizmy uwierzytelnienia	MUST	MUST	MUST
2.7	Zweryfikuj, czy funkcja zmiany hasła zawiera konieczność podania hasła obecnie używanego, nowego hasła oraz konieczność ponownego wpisania nowego hasła.	MUST	MUST	MUST
2.8	Zweryfikuj, czy wszystkie decyzje dotyczące uwierzytelnienia mogą być logowane bez przechowywania poufnych identyfikatorów sesji lub haseł. Powinno to zawierać zapytania do odpowiednich metadanych niezbędnych do ewentualnego postępowania wyjaśniającego w zakresie bezpieczeństwa.	OPTIONAL	MUST	MUST
2.9	Zweryfikuj, czy hasła do kont są przechowywane wykorzystując jednokierunkową funkcję skrótu z solą i czy są wystarczająco skomplikowane by się obronić przed atakami brute force oraz odtwarzaniem hasła z jego skrótu.	OPTIONAL	MUST	MUST
2.10	Zweryfikuj, czy dane uwierzytelniające są przesyłane wykorzystując właściwy zaszyfrowany link oraz, czy wszystkie strony/funkcje które wymagają by użytkownik podał dane uwierzytelniające wymagają takiego szyfrowanego linku.	MUST	MUST	MUST
2.11	Zweryfikuj, czy mechanizmy odzyskiwania hasła nie ujawniają dotychczasowych haseł i czy nowe hasło nie jest przesyłane w formie jawnej do użytkownika.	MUST	MUST	MUST
2.12	Zweryfikuj, czy nie jest możliwa enumeracja wykorzystująca loginy użytkowników, funkcję resetowania hasła lub funkcjonalność przypomnienia nazwy użytkownika.	MUST	MUST	MUST
2.13	Zweryfikuj, czy platforma lub inne komponenty z których korzysta aplikacja nie używają domyślnych haseł (np. admin/password).	MUST	MUST	MUST

Centrum e-Zdrowia
ul. Stanisława Dubois 5A
00-184 Warszawa
tel.: +48 22 597-09-27
fax: +48 22 597-09-37
biuro@cez.gov.pl | www.cez.gov.pl
NIP: 5251575309
REGON: 001377706

2.14	Zweryfikuj, czy wdrożono mechanizmy przeciwdziałające automatyzacji, aby zapobiegać testowaniu ujawnionych danych uwierzytelniających, atakom brute force i atakom blokującym konta.	MUST	MUST	MUST
2.15	Zweryfikuj, czy wszystkie dane uwierzytelniające służące do uzyskiwania dostępu do usług zewnętrznych względem aplikacji, są zaszyfrowane i przechowywane w zabezpieczonej lokalizacji (a nie w kodzie źródłowym).	OPTIONAL	MUST	MUST
2.16	Zweryfikuj, czy funkcje odzyskiwania hasła i inne ścieżki odzyskiwania poświadczeń uwzględniają użycie tokenu udostępniającego jednorazowe, zmienne w czasie hasła (Time-based One-Time Password algorithm - TOTP) lub innego tokena programowego, mechanizmu push lub mechanizmu odzyskiwania offline. Wykorzystywanie losowych wartości przesyłanych pocztą elektroniczną lub SMS powinno być jedynie ostatecznością, gdyż dowiedziono ich słabości.	MUST	MUST	MUST
2.17	Zweryfikuj że status blokowania konta uwzględnia mechanizm "miękkiej" i "twardej" blokady, przy czym te statusy nie wykluczają się nawzajem. Jeżeli konto jest zablokowane w sposób "miękki" ze względu na atak brute force, nie powinno to resetować statusu "twardej" blokady	OPTIONAL	MUST	MUST
2.18	Zweryfikuj, czy w przypadku wykorzystywania współdzielonych sekretnych pytań, nie naruszają one przepisów dotyczących prywatności i są na tyle trudne by w rzeczywistości chronić aplikację.	MUST	MUST	MUST
2.19	Zweryfikuj, czy system umożliwia zablokowanie ponownego wykorzystania określonej liczby haseł, poprzednio użytych przez użytkownika.	OPTIONAL	MUST	MUST
2.20	Zweryfikuj, czy powtórne uwierzytelnianie, dwuskładnikowe uwierzytelnianie lub podpisywanie danych transakcyjnych jest stosowane w przypadku transakcji wysokiej wartości.	OPTIONAL	MUST	MUST
2.21	Zweryfikuj, czy wdrożono mechanizmy blokujące użycie znanych lub słabych haseł.	MUST	MUST	MUST
2.22	Zweryfikuj, czy wszystkie próby uwierzytelnienia, niezależnie czy zakończone powodzeniem czy niepowodzeniem, odpowiadają w tym samym średnim czasie.	OPTIONAL	OPTIONAL	MUST
2.23	Zweryfikuj, czy hasła, klucze bądź inne informacje uwierzytelniające nie są zapisane w kodzie źródłowym lub repozytoriach kodu.	OPTIONAL	OPTIONAL	MUST

Centrum e-Zdrowia
ul. Stanisława Dubois 5A
00-184 Warszawa
tel.: +48 22 597-09-27
fax: +48 22 597-09-37
biuro@cez.gov.pl | www.cez.gov.pl
NIP: 5251575309
REGON: 001377706

2.24	Zweryfikuj, czy aplikacja pozwala użytkownikom na uwierzytelnienie przy wykorzystaniu uprawnionego bezpiecznego mechanizmu uwierzytelniania.	MUST	MUST	MUST
2.25	Zweryfikuj, czy jeżeli aplikacja pozwala użytkownikom na uwierzytelnienie, mogą oni to zrobić wykorzystując dwa czynniki lub inne silne metody uwierzytelniania albo też wykorzystując inny podobny sposób postępowania, który chroni przed ujawnieniem nazwy wraz z hasłem.	OPTIONAL	MUST	MUST
2.26	Zweryfikuj, czy interfejs administracyjny nie jest dostępny dla stron niezaufanych.	MUST	MUST	MUST
2.27	Funkcja autouzupełniania w przeglądarce lub integracja z menedżerami haseł powinny być dopuszczone, chyba że są zabronione na podstawie przeprowadzonej analizy ryzyka.	MUST	MUST	MUST
3	**Wymagania weryfikacji dla zarządzania sesją** Jednym z głównych komponentów każdej webaplikacji jest mechanizm, przy pomocy którego nadzorowany i utrzymywany jest jej stan, w trakcie interakcji z użytkownikiem. Jest to określane jako zarządzanie sesją i jest definiowane jako zestaw narzędzi nadzorujących stanowo (state-full) interakcje pomiędzy użytkownikiem i webaplikacją. Upewnij się, że weryfikowana aplikacja w odpowiedni sposób spełnia poniższe ogólne wymagania: • sesje są unikalne dla każdego użytkownika i nie mogą zostać odgadnięte lub współdzielone. sesje są unieważniane, gdy tylko przestają być niezbędne oraz przerywane, gdy nie są wykorzystywane.			
3.1	Zweryfikuj, czy aplikacja nie używa zmodyfikowanego managera sesji lub, jeśli jest on używany, to czy jest odporny na wszystkie powszechne ataki na zarządzanie sesją.	MUST	MUST	MUST
3.2	Zweryfikuj, czy sesje są unieważniane po wylogowaniu się użytkownika.	MUST	MUST	MUST
3.3	Zweryfikuj, czy sesje wygasają po określonym czasie bezczynności.	MUST	MUST	MUST
3.4	Zweryfikuj, czy sesje wygasają po administracyjnie konfigurowalnym maksymalnym okresie, niezależnie od aktywności (bezwzględny okres ważności).			MUST
3.5	Zweryfikuj, czy wszystkie strony, do których dostęp wymaga uwierzytelnienia, zawierają łatwy i widoczny dostęp do funkcji wylogowania.	MUST	MUST	MUST
3.6	Zweryfikuj, czy identyfikatory sesji nigdy nie są ujawniane w URL, w komunikatach błędów lub logach. Należy również zweryfikować, czy aplikacja nie wspiera przepisywania w URL ciasteczek sesyjnych (ang. URL-rewriting).	MUST	MUST	MUST

Centrum e-Zdrowia
ul. Stanisława Dubois 5A
00-184 Warszawa
tel.: +48 22 597-09-27
fax: +48 22 597-09-37
biuro@cez.gov.pl | www.cez.gov.pl
NIP: 5251575309
REGON: 001377706

3.7	Zweryfikuj, czy każde pomyślne uwierzytelnienie a także ponowne uwierzytelnienie, tworzy nową sesję z nowym identyfikatorem.	MUST	MUST	MUST
3.8	Zweryfikuj, czy tylko identyfikatory sesji wygenerowane przez platformę aplikacji są uznawane przez nią za aktywne.	OPTIONAL	MUST	MUST
3.9	Zweryfikuj, czy identyfikatory sesji są wystarczająco długie, losowe i unikalne w obrębie odpowiedniej aktywnej bazy sesji.	OPTIONAL	MUST	MUST
3.10	Zweryfikuj, czy identyfikatory sesji przechowywane w ciasteczkach mają ustawioną ścieżkę z wartością odpowiednio restrykcyjną dla danej aplikacji i czy tokeny sesji uwierzytelniania mają dodatkowo ustawione atrybuty "HttpOnly" i "secure".	MUST	MUST	MUST
3.11	Zweryfikuj, czy aplikacja ogranicza liczbę jednoczesnych aktywnych sesji.	MUST	MUST	MUST
3.12	Zweryfikuj, czy lista aktywnych sesji jest wyświetlana dla każdego użytkownika w profilu konta lub podobnym. Użytkownik powinien posiadać możliwość zakończenia dowolnej aktywnej sesji.	MUST	MUST	MUST
3.13	Zweryfikuj, czy użytkownik ma zaproponowaną opcję zakończenia wszystkich innych aktywnych sesji, po pomyślnym zakończeniu procesu zmiany hasła.	MUST	MUST	MUST
4	**Wymagania weryfikacji dla kontroli dostępu** Uwierzytelnianie jest koncepcją zapewniającą dostęp jedynie do tych zasobów, na które wyrażono zgodę. Upewnij się, że weryfikowana aplikacja w odpowiedni sposób spełnia poniższe ogólne wymagania: • osoby uzyskujące dostęp posiadają ważne dane uwierzytelniające, • użytkownicy są powiązani z dobrze zdefiniowanymi zestawami ról i uprawnień. role i uprawnienia są chronione przed ponownym wykorzystaniem lub modyfikacją.			
4.1	Zweryfikuj, czy została wdrożona zasada minimalnych uprawnień - użytkownicy powinni mieć dostęp tylko do funkcji, plików, linków URL, usług oraz innych zasobów do których posiadają zezwolenie.	MUST	MUST	MUST
4.2	Zweryfikuj, czy dostęp do wrażliwych zapisów jest chroniony w taki sposób, aby tylko autoryzowane obiekty lub dane były dostępne dla użytkownika (dla przykładu chroń przed ingerowaniem użytkowników w parametry umożliwiające zobaczenie lub dokonywanie zmian na koncie innego użytkownika).	MUST	MUST	MUST
4.3	Zweryfikuj, czy listowanie katalogów jest wyłączone (chyba, że jest celowo dozwolone). Dodatkowo aplikacje nie powinny	MUST	MUST	MUST

Centrum e-Zdrowia
ul. Stanisława Dubois 5A
00-184 Warszawa
tel.: +48 22 597-09-27
fax: +48 22 597-09-37
biuro@cez.gov.pl | www.cez.gov.pl
NIP: 5251575309
REGON: 001377706

	pozwalać na odkrycie lub ujawnienie plików lub metadanych katalogów takich jak Thumbs.db, .DS_Store, .git lub foldery .svn.			
4.4	Zweryfikuj, czy mechanizmy kontroli dostępu, które kończą pracę niepowodzeniem, robią to w sposób bezpieczny	MUST	MUST	MUST
4.5	Zweryfikuj, czy te same reguły kontroli dostępu występujące w warstwie prezentacji są również egzekwowane po stronie serwera.	MUST	MUST	MUST
4.6	Zweryfikuj, czy wszystkie atrybuty użytkowników i danych oraz informacje o zasadach dostępu wykorzystywane przez mechanizmy kontroli dostępu, nie mogą być modyfikowane przez użytkowników, chyba, że są oni do tego uprawnieni.	OPTIONAL	MUST	MUST
4.7	Zweryfikuj, czy istnieje scentralizowany mechanizm zabezpieczający dostęp do każdego typu chronionych zasobów (również dla bibliotek wywołujących zewnętrzne usługi autoryzacji).	OPTIONAL	OPTIONAL	MUST
4.8	Zweryfikuj, czy wszystkie decyzje dotyczące kontroli dostępu mogą być logowane, a wszystkie decyzje zakończone niepowodzeniem są logowane.	OPTIONAL	MUST	MUST
4.9	Zweryfikuj, czy aplikacja bądź platforma generuje silne, losowe tokeny zabezpieczające przed atakami typu CSRF lub posiadają inne mechanizmy ochrony transakcji.	MUST	MUST	MUST
4.10	Zweryfikuj, czy system może się ochronić się przed masowym lub ciągłym dostępem do zabezpieczonych funkcji, zasobów bądź danych. Na przykład należy rozważyć wykorzystanie zarządcy zasobów w celu ograniczenia liczby edycji na godzinę lub zapobiec odczytaniu wszystkich danych przez jednego użytkownika w sposób zautomatyzowany (tzw. „scrapping”).	OPTIONAL	MUST	MUST
4.11	Zweryfikuj, czy aplikacja wykorzystuje dodatkowe uwierzytelnianie (takie jak uwierzytelnianie wieloetapowe bądź biorące pod uwagę profil ryzyka danego użytkownika) dla mniej ważnych systemów i/lub segregację obowiązków dla ważniejszych systemów, aby wymuszać mechanizmy chroniące przed defraudacjami, które są wdrażane bazując na zidentyfikowanym ryzyku aplikacji i dawnych przypadkach defraudacji.	OPTIONAL	MUST	MUST
4.12	Zweryfikuj, czy aplikacja w poprawny sposób wymusza autoryzację zależną od kontekstu, tak aby nie umożliwiać nieautoryzowanych manipulacji realizowanych poprzez zmiany wartości parametrów.	MUST	MUST	MUST
5	**Wymagania weryfikacji dla obsługi złośliwych danych wejściowych**			

Centrum e-Zdrowia
ul. Stanisława Dubois 5A
00-184 Warszawa
tel.: +48 22 597-09-27
fax: +48 22 597-09-37
biuro@cez.gov.pl | www.cez.gov.pl
NIP: 5251575309
REGON: 001377706

	Najczęstszą słabością spotykaną w web-aplikacjach jest niepowodzenie w trakcie poprawnej walidacji danych wejściowych pochodzących od klientów lub z samych środowisk, zanim zostały one użyte. To prowadzi do praktycznie wszystkich najpoważniejszych podatności w aplikacjach webowych , takich jak cross site scripting, SQL injection, interpreter injection, ataki na locale/Unicode, ataki na system plików, przepełnienie bufora. Upewnij się, że weryfikowana aplikacja w odpowiedni sposób spełnia poniższe ogólne wymagania: • wszystkie dane wejściowe są walidowane aby zapewnić, że są poprawne i dostosowane do zamierzonych celów, • dane z zewnętrznych źródeł lub od klientów nigdy nie powinny być traktowane jako zaufane i powinny być odpowiednio traktowane.			
5.1	Zweryfikuj, czy środowisko uruchomieniowe nie jest podatne na przepełnienia bufora lub czy mechanizmy bezpieczeństwa zapobiegają wystąpieniu przepełnienia bufora.	MUST	MUST	MUST
5.2	Zweryfikuj, czy wszystkie prowadzone po stronie serwera walidacje wejścia, które zakończone niepowodzeniem, powodują odrzucenie żądania oraz są logowane.	MUST	MUST	MUST
5.3	Zweryfikuj, czy wszystkie mechanizmy walidacji są egzekwowane po stronie serwera.	MUST	MUST	MUST
5.4	Zweryfikuj, czy aplikacja wykorzystuje jeden mechanizm walidacji danych wejściowych dla każdego typu przyjmowanych danych.	OPTIONAL	OPTIONAL	MUST
5.5	Zweryfikuj, czy wszystkie kwerendy SQL, HQL, OSQL, NOSQL, składowane procedury, wywołania do składowanych procedur są chronione poprzez wykorzystywanie przygotowanych zapytań lub sparametryzowanych kwerend, i tym samym nie są podatne na atak SQL Injection.	MUST	MUST	MUST
5.6	Zweryfikuj, czy aplikacja nie jest podatna na atak LDAP Injection lub mechanizmy bezpieczeństwa zapobiegają wystąpieniu takiego ataku.	MUST	MUST	MUST
5.7	Zweryfikuj, czy środowisko uruchomieniowe nie jest podatne na atak wstrzyknięcia komend systemu operacyjnego lub mechanizmy bezpieczeństwa zapobiegają wystąpieniu takiego ataku.	MUST	MUST	MUST
5.8	Zweryfikuj, czy aplikacja nie jest podatna na ataki zdalnego lub lokalnego dołączenia plików (Remote File Inclusion - RFI lub Local File Inclusion - LFI) gdy wykorzystywana jest treść będąca ścieżką do plików.	MUST	MUST	MUST
5.9	Zweryfikuj, czy aplikacja nie jest podatna na ataki XML Injection, XML External Entity, XPath query lub mechanizmy bezpieczeństwa zapobiegają wystąpieniu takich ataków.	MUST	MUST	MUST

Centrum e-Zdrowia
ul. Stanisława Dubois 5A
00-184 Warszawa
tel.: +48 22 597-09-27
fax: +48 22 597-09-37
biuro@cez.gov.pl | www.cez.gov.pl
NIP: 5251575309
REGON: 001377706

5.10	Upewnij się, że wszystkie łańcuchy zmiennych włączane w HTML lub inny kod uruchamiany po stronie klienta webowego są albo w poprawny sposób ręcznie wprowadzone w sposób zależny od kontekstu, albo wykorzystują szablony, wprowadzając je automatycznie w sposób zależny od kontekstu aby zapewnić, że aplikacja nie jest podatna na ataki XSS typu „reflected", „stored" i DOM.	MUST	MUST	MUST
5.11	Jeżeli platforma aplikacji pozwala na masowe przypisywanie parametrów (mass assignment), zwane także automatycznym wiązaniem zmiennych (automatic variable binding) z przychodzącego żądania do modelu, zweryfikuj, czy istotne z punktu widzenia bezpieczeństwa pola (np. takie jak "stan_konta", "rola", "password") są chronione przed złośliwym automatycznym wiązaniem.	OPTIONAL	MUST	MUST
5.12	Zweryfikuj, czy system jest zabezpieczony przed atakami typu HTTP Parametr Pollution, szczególnie jeżeli platforma aplikacji nie rozróżnia źródła pochodzenia parametrów żądania (GET, POST, nagłówki, ciasteczka, środowisko, itd.)	OPTIONAL	MUST	MUST
5.13	Zweryfikuj, czy walidacja po stronie klienta jest wykorzystywana jako druga linia obrony - będąca uzupełnieniem walidacji wykonywanej po stronie serwera.	OPTIONAL	MUST	MUST
5.14	Zweryfikuj, czy wszystkie dane wejściowe są walidowane. Dotyczy to nie tylko pól formularzy HTML, ale wszystkich źródeł wejścia takich jak wywołania REST, parametry zapytań, nagłówki HTTP, ciasteczka, pliki wsadowe, kanały RSS itd. Należy używać walidacji pozytywnej (whitelisting), następnie eliminować znane "złe" ciągi znaków (greylisting) lub odrzucać znane "złe" wejścia (blacklisting).	OPTIONAL	MUST	MUST
5.15	Zweryfikuj, czy ustrukturyzowane dane są silnie zdefiniowane i walidowane względem schematu uwzględniając: dopuszczone znaki, długość i zgodność ze wzorcem (np. numer karty kredytowej, numer telefonu lub walidację, czy relacja dwóch pól wejściowych jest odpowiednia - np. czy podana lokalizacja jest zgodna z podanym kodem pocztowym).	OPTIONAL	MUST	MUST
5.16	Zweryfikuj, czy niestrukturalne dane są wyczyszczone i zawierają dozwolone znaki oraz długość, a także czy wszystkie znaki potencjalnie szkodliwe dla danego kontekstu powinny zostać usunięte (np. nazwy lokalne pisane w Unicode oraz apostrofy)	OPTIONAL	MUST	MUST
5.17	Zweryfikuj, czy niezaufany kod HTML z edytorów WYSIWYG lub podobnych jest wyczyszczony oraz obsługiwany odpowiednio w zakresie walidacji wejścia i enkodowania wyjścia.	MUST	MUST	MUST

Centrum e-Zdrowia
ul. Stanisława Dubois 5A
00-184 Warszawa
tel.: +48 22 597-09-27
fax: +48 22 597-09-37
biuro@cez.gov.pl | www.cez.gov.pl
NIP: 5251575309
REGON: 001377706

5.18	W przypadku korzystania z technologii szablonów z funkcją automatycznego cytowania, gdy wyłączone jest cytowanie przez UI, zapewnij należyte czyszczenie kodu HTML.	OPTIONAL	MUST	MUST
5.19	Zweryfikuj, czy dane przenoszone pomiędzy kontekstami DOM używają bezpiecznych metod JavaScript np. .innerText i .val.	OPTIONAL	MUST	MUST
5.20	Zweryfikuj, czy przy parsowaniu JSON w przeglądarce, metoda JSON.parse jest wykorzystywana do parsowania JSON na kliencie. Nie wykorzystuj eval() parsowania JSON po stronie klienta.	OPTIONAL	MUST	MUST
5.21	Zweryfikuj, czy dane zapisywane po uwierzytelnieniu w pamięci przeglądarki np. w DOM są czyszczone po zakończeniu sesji.	OPTIONAL	MUST	MUST
6	**Wymagania weryfikacji dla nieaktywnych mechanizmów kryptograficznych** Upewnij się, że weryfikowana aplikacja w odpowiedni sposób spełnia poniższe ogólne wymagania: • wszystkie moduły kryptograficzne kończące pracę niepowiedzeniem robią to w sposób bezpieczny, • w przypadku gdy wymagana jest losowość, jest wykorzystywany odpowiedni generator liczb losowych, • dostęp do kluczy jest zarządzany w bezpieczny sposób.			
6.1	Zweryfikuj, czy wszystkie moduły kryptograficzne, które kończą pracę niepowodzeniem robią to w sposób bezpieczny a błędy są obsługiwane w sposób zapobiegający atakom oracle padding.	MUST	MUST	MUST
6.2	Zweryfikuj, czy wszystkie liczby losowe, losowe nazwy plików, losowe GUID'y oraz losowe ciągi znaków, których atakujący nie powinien odgadnąć, generowane są z wykorzystaniem zatwierdzonego generatora liczb losowych z modułu kryptograficznego.	OPTIONAL	MUST	MUST
6.3	Zweryfikuj, czy wszystkie moduły kryptograficzne wykorzystywane przez aplikację zostały sprawdzone na zgodność ze standardem FISP 140-2 lub równorzędnym.	MUST	MUST	MUST
6.4	Zweryfikuj, czy moduły kryptograficzne pracują w zaakceptowanym trybie, zgodnym dla opublikowanych dla nich polityk bezpieczeństwa.	OPTIONAL	OPTIONAL	MUST
6.5	Zweryfikuj, czy istnieje jasna polityka określająca jak są zarządzane klucze kryptograficzne (np. jak są generowane, rozprowadzane, unieważniane, w jaki sposób wygasają). Zweryfikuj, czy ta polityka jest właściwie egzekwowana.	OPTIONAL	MUST	MUST

Centrum e-Zdrowia
ul. Stanisława Dubois 5A
00-184 Warszawa
tel.: +48 22 597-09-27
fax: +48 22 597-09-37
biuro@cez.gov.pl | www.cez.gov.pl
NIP: 5251575309
REGON: 001377706

6.6	Zweryfikuj, czy użytkownicy usług kryptograficznych nie mają bezpośredniego dostępu do kluczowych materiałów. Izoluj procesy kryptograficzne, włączając w to klucz główny, i rozważ wykorzystanie zwirtualizowanego lub sprzętowego modułu zabezpieczeń (HSM).	OPTIONAL	OPTIONAL	MUST
6.7	Dane osobowe powinny być przechowywane w postaci zaszyfrowanej i należy zapewnić aby komunikacja odbywała się za pomocą zabezpieczonych kanałów.	OPTIONAL	MUST	MUST
6.8	Zweryfikuj, czy hasła wymagające ochrony oraz klucze znajdujące się w pamięci, są nadpisywane zerami gdy tylko przestają być wymagane, aby zabezpieczyć się przed atakami zrzucania pamięci (memory dumping).	OPTIONAL	MUST	MUST
6.9	Zweryfikuj, czy wszystkie klucze i hasła mogą być wymieniane oraz, że są generowane lub zmieniane podczas instalacji.	OPTIONAL	MUST	MUST
6.10	Zweryfikuj, czy entropia generowanych liczb losowych jest wystarczająca, nawet gdy aplikacja jest mocno obciążona, ew. aplikacja powinna w adekwatny sposób reagować na zmniejszenie się puli entropii.	OPTIONAL	OPTIONAL	MUST
7	**Wymagania weryfikacji dla obsługi i logowania błędów** Głównym celem obsługi i logowania błędów jest zapewnienie reakcji przydatnej użytkownikom, administratorom i zespołom reagowania na incydenty. Celem nie jest masowe tworzenie logów, lecz logów wysokiej jakości, niosących więcej treści niż odrzucany szum. Logi wysokiej jakości często będą zawierać dane wymagające ochrony i dlatego muszą być chronione zgodnie z lokalnymi przepisami oraz dyrektywami dotyczącymi prywatności. Powinno o obejmować następujące zasady: • nie gromadzenie lub logowanie informacji wymagających ochrony, jeżeli nie jest to niezbędnie wymagane, • zapewnienie, że wszystkie logowane informacje są obsługiwane w sposób bezpieczny i chronione zgodnie z klasyfikacją tych danych, • zapewnienie, że logi nie są przetrzymywane nieustannie, lecz mają zdefiniowaną ważność na okres tak krótki jak to możliwe. Jeżeli logi zawierają dane osobowe lub informacje podlegające ochronie, to stają się jednymi z najbardziej wrażliwych informacji przetwarzanymi w aplikacji i tym samym stają się atrakcyjnym celem dla atakujących.			
7.1	Zweryfikuj, czy aplikacja nie zwraca komunikatów o błędach lub śladów stosu (stack traces), które zawierają dane wrażliwe i które mogą pomóc atakującym. Zawierają się w tym identyfikatory sesji, wersje oprogramowania/ platformy i dane osobowe.	MUST	MUST	MUST

Centrum e-Zdrowia
ul. Stanisława Dubois 5A
00-184 Warszawa
tel.: +48 22 597-09-27
fax: +48 22 597-09-37
biuro@cez.gov.pl | www.cez.gov.pl
NIP: 5251575309
REGON: 001377706

7.2	Zweryfikuj, czy logika zarządzania błędami w mechanizmach bezpieczeństwa domyślnie zabrania do nich dostępu.	OPTIONAL	MUST	MUST
7.3	Zweryfikuj, czy mechanizmy logowania zdarzeń bezpieczeństwa zapewniają możliwość rejestracji zdarzeń istotnych z punktu widzenia bezpieczeństwa zarówno zakończonych sukcesem jak i porażką.	OPTIONAL	MUST	MUST
7.4	Zweryfikuj, czy log każdego zdarzenia zawiera niezbędną informację, która pozwoli na precyzyjną analizę czasową w przypadku wystąpienia zdarzenia.	OPTIONAL	MUST	MUST
7.5	Zweryfikuj, czy wszystkie zdarzenia zawierające niezaufane dane nie zostaną uruchomione jako kod w oprogramowaniu służącym do przeglądania logów.	OPTIONAL	OPTIONAL	MUST
7.6	Zweryfikuj, czy logi bezpieczeństwa są chronione przed nieautoryzowanym dostępem i modyfikacją.	OPTIONAL	MUST	MUST
7.7	Zweryfikuj, czy aplikacja nie loguje wrażliwych danych zdefiniowanych zgodnie z lokalnymi regulacjami lub polityką prywatności, danych wrażliwych z punktu widzenia organizacji zdefiniowanych w ramach szacowania ryzyka, lub wrażliwych danych uwierzytelniających, które mogłyby pomóc atakującemu, włączając w to identyfikatory sesji, hasła, ciągi hash lub tokeny API.	OPTIONAL	MUST	MUST
7.8	Zweryfikuj, że znaki niedrukowalne oraz separatory pól są prawidłowo zakodowane w rejestrach logów, w sposób zapobiegający wstrzykiwaniu logów.	OPTIONAL	OPTIONAL	MUST
7.9	Zweryfikuj, że pola logów pochodzące ze źródeł zaufanych i niezaufanych są rozróżnialne we wpisach logów.	OPTIONAL	OPTIONAL	MUST
7.10	Zweryfikuj, że logi audytowe lub podobne rejestry umożliwiają niezaprzeczalność kluczowych transakcji.	OPTIONAL	MUST	MUST
7.11	Zweryfikuj, czy logi bezpieczeństwa posiadają jakąś formę kontroli integralności lub mechanizmy zapobiegające nieautoryzowanej modyfikacji.	OPTIONAL	OPTIONAL	MUST
7.12	Zweryfikuj, czy logi są przechowywane w innej partycji niż ta, w której uruchomiona jest aplikacja, z zapewnieniem odpowiedniej rotacji logów.	OPTIONAL	OPTIONAL	MUST
7.13	Źródła czasu powinny być synchronizowane aby zapewnić, że logi mają poprawny czas.	MUST	MUST	MUST
8	**Wymagania weryfikacji dla mechanizmów ochrony danych** Istnieją trzy elementy aby należycie chronić dane: poufność, integralność i dostępność. Zakłada to, że ochrona danych jest wymuszana na zaufanych systemach, takich jak serwery, które zostały utwardzone i posiadają wystarczające zabezpieczenia. Aplikacje powinny jednak zakładać, że			

Centrum e-Zdrowia
ul. Stanisława Dubois 5A
00-184 Warszawa
tel.: +48 22 597-09-27
fax: +48 22 597-09-37
biuro@cez.gov.pl | www.cez.gov.pl
NIP: 5251575309
REGON: 001377706

	zabezpieczenia wszystkich urządzeń są w jakimś stopniu skompromitowane. Gdy aplikacja przesyła lub przechowuje dane wymagające ochrony na niebezpiecznych urządzeniach, jakich jak współdzielone komputery, telefony i tablety, to aplikacja jest odpowiedzialna za to, że dane te będą zaszyfrowane i nie będzie można ich w łatwy sposób ich zdobyć, zmienić lub ujawnić. Zależy więc zapewnić, że weryfikowana aplikacja w sposób satysfakcjonujący przestrzega następujących ogólnych wymagań bezpieczeństwa: • poufność: dane powinny być chronione przed nieautoryzowanym podglądem lub ujawnieniem, zarówno podczas transmisji jak i podczas przechowywania, • integralność: dane powinny być chronione przed złośliwym tworzeniem, zmianą lub kasowaniem przez nieupoważnionych atakujących, • dostępność: dane powinny być dostępne dla autoryzowanych użytkowników gdy tylko są potrzebne.			
8.1	Zweryfikuj, czy dla wszystkich formularzy zawierających dane wrażliwe, wyłączone jest cachowanie po stronie klienta, włączając w to funkcjonalność automatycznego wypełnienia pól.	MUST	MUST	MUST
8.2	Zweryfikuj, czy zidentyfikowano listę danych wrażliwych przetwarzanych przez aplikację i czy istnieje jasna polityka określająca jak dostęp do tych danych powinien być kontrolowany, szyfrowany i przyznawany zgodnie z odpowiednim prawodawstwem w zakresie ochrony danych.	OPTIONAL	OPTIONAL	MUST
8.3	Zweryfikuj, czy wszystkie dane wrażliwe są przesyłane do serwera wewnątrz wiadomości HTTP lub jej nagłówka (np. parametry URL nie powinny być używane do przesyłania danych wrażliwych).	MUST	MUST	MUST
8.4	Zweryfikuj, czy aplikacja posiada ustawienia w nagłówku, uniemożliwiające cachowanie danych odpowiednie do założonego ryzyka aplikacji.	MUST	MUST	MUST
8.5	Zweryfikuj, czy na serwerze wszystkie wrażliwe dane znajdujące się w cache lub kopiach tymczasowych, są chronione przed nieautoryzowanym dostępem lub czyszczone/unieważniane po uzyskaniu dostępu do danych wrażliwych przez upoważnionego użytkownika.	OPTIONAL	MUST	MUST
8.6	Zweryfikuj, czy istnieje metoda usunięcia z aplikacji każdego typu danych wrażliwych, na koniec ich wymaganego okresu retencji.	OPTIONAL	OPTIONAL	MUST
8.7	Zweryfikuj, czy aplikacja minimalizuje ilość parametrów wysyłanych w zapytaniach takimi kanałami jak: ukryte pola, zmienne systemu AJAX, ciasteczka i nagłówki wiadomości.	OPTIONAL	MUST	MUST
8.8	Zweryfikuj, czy aplikacja potrafi wykrywać i alarmować o nienormalnej liczbie zapytań o informacje mogących świadczyć	OPTIONAL	OPTIONAL	MUST

Centrum e-Zdrowia
ul. Stanisława Dubois 5A
00-184 Warszawa
tel.: +48 22 597-09-27
fax: +48 22 597-09-37
biuro@cez.gov.pl | www.cez.gov.pl
NIP: 5251575309
REGON: 001377706

	o próbie gromadzenia danych, na przykład poprzez „screen scraping”.			
8.9	Zweryfikuj, czy dane przechowywane po stronie klienta (takie jak lokalnie przechowywane informacje o sesjach HTML5, przechowywane sesje, IndexedDB, ciasteczka: zwykłe i Flash), nie zawierają danych podlegających ochronie oraz danych osobowych (PII).	MUST	MUST	MUST
8.10	Zweryfikuj, czy dostęp do danych wrażliwych jest logowany, w przypadku gdy dane są gromadzone zgodnie z odpowiednimi przepisami w zakresie ochrony danych lub gdy logowanie dostępów jest wymagane.	OPTIONAL	MUST	MUST
8.11	Zweryfikuj, czy dane wymagające ochrony znajdujące się w pamięci, są nadpisywane zerami gdy tylko przestaną być niezbędne, aby zabezpieczyć się przed atakami wykorzystującymi zrzuty pamięci (memory dumping).	OPTIONAL	MUST	MUST
9	**Wymagania weryfikacji dla zabezpieczania komunikacji** Upewnij się, że weryfikowana aplikacja w odpowiedni sposób spełnia poniższe ogólne wymagania: • TLS w bezpiecznej wersji (np. TLS 1.2+) jest zawsze wykorzystywany gdy dane wymagające ochrony są transmitowane, • Silne algorytmy i szyfry są zawsze wykorzystywane.			
9.1	Zweryfikuj, czy możliwe jest zbudowanie ścieżki od zaufanego CA do każdego serwerowego certyfikatu Transport Layer Security (TLS) i czy każdy certyfikat serwera jest ważny.	MUST	MUST	MUST
9.2	Zweryfikuj, czy dla wszystkich połączeń (zewnętrznych i wewnętrznych), które są uwierzytelniane lub związane z wrażliwymi danymi lub funkcjami, jest wykorzystywany TLS a także nie jest możliwe pogorszenie parametrów połączenia do połączenia niezabezpieczonego. Upewnij się, że preferowany jest najsilniejszy dostępny algorytm szyfrowania.	MUST	MUST	MUST
9.3	Zweryfikuj, czy błędy związane z wewnętrznymi połączeniami TLS pomiędzy komponentami są logowane.	OPTIONAL	OPTIONAL	MUST
9.4	Zweryfikuj, czy ścieżki certyfikatów są budowane i weryfikowane dla wszystkich certyfikatów klienckich przy użyciu skonfigurowanych powiązań zaufania i informacji dotyczących unieważnionych certyfikatów.	OPTIONAL	OPTIONAL	MUST
9.5	Zweryfikuj, czy wszystkie połączenia do zewnętrznych systemów, które dotyczą wrażliwych informacji lub funkcji są uwierzytelniane.	OPTIONAL	MUST	MUST

Centrum e-Zdrowia
ul. Stanisława Dubois 5A
00-184 Warszawa
tel.: +48 22 597-09-27
fax: +48 22 597-09-37
biuro@cez.gov.pl | www.cez.gov.pl
NIP: 5251575309
REGON: 001377706

9.6	Zweryfikuj, czy aplikacja używa implementacji TLS opartej na pojedynczym standardzie, skonfigurowanej do działania w zatwierdzonym trybie pracy.	OPTIONAL	OPTIONAL	MUST
9.7	Zweryfikuj, czy mechanizm przypinania certyfikatów publicznych TLS (HPKP) jest używany dla kluczy produkcyjnych i zapasowych. Więcej informacji możesz znaleźć w odnośnikach w niniejszym standardzie.	OPTIONAL	OPTIONAL	MUST
9.8	Zweryfikuj, czy nagłówki HTTP Strict Transport Security są włączone do wszystkich zapytań i dla wszystkich poddomen np. w postaci: Strict-Transport-Security: max-age=15724800; includeSubdomains.	MUST	MUST	MUST
9.9	Zweryfikuj, czy produkcyjny URL został zgłoszony do listy predefiniowanych domen używających mechanizmu Strict Transport Security, utrzymywanych przez producentów przeglądarek. Więcej informacji znajdziesz w odnośnikach poniżej.	OPTIONAL	OPTIONAL	MUST
9.10	Upewnij się, że w celu zmniejszenia prawdopodobieństwa pasywnych ataków polegających na zapisywaniu ruchu, używane są szyfry wspierające doskonałe utajnienie przekazywania (forward secrecy).	MUST	MUST	MUST
9.11	Zweryfikuj, czy są włączone i właściwie skonfigurowane mechanizmy unieważniania certyfikatów, takie jak „zszywanie” odpowiedzi Online Certificate Status Protocol (OSCP).	MUST	MUST	MUST
9.12	Zweryfikuj, czy używane są tylko silne algorytmy, szyfry i protokoły w ramach całej hierarchii certyfikatów, włącznie z certyfikatem root i certyfikatami pośrednimi w ramach wybranego urzędu certyfikacji.	MUST	MUST	MUST
9.13	Zweryfikuj, czy konfiguracja TLS jest zgodna z bieżącymi najlepszymi praktykami szczególnie dlatego, że popularne ustawienia, szyfry i algorytmy z czasem mogą okazać się niebezpieczne.	MUST	MUST	MUST
10	**Wymagania weryfikacji dla konfigurowania bezpieczeństwa http** Upewnij się, że weryfikowana aplikacja w odpowiedni sposób spełnia poniższe ogólne wymagania: • konfiguracja serwera aplikacyjnego jest odpowiednio utwardzona, • odpowiedzi http zawierają bezpieczny zestaw znaków w nagłówku "„content type”.			
10.1	Zweryfikuj, czy aplikacja akceptuje tylko zdefiniowany zestaw metod zapytań HTTP, takich jak GET i POST oraz czy nieużywane metody (takie jak TRACE, PUT, DELETE) są w sposób wyraźny zablokowane.	MUST	MUST	MUST

Centrum e-Zdrowia
ul. Stanisława Dubois 5A
00-184 Warszawa
tel.: +48 22 597-09-27
fax: +48 22 597-09-37
biuro@cez.gov.pl | www.cez.gov.pl
NIP: 5251575309
REGON: 001377706

10.2	Zweryfikuj, czy każda odpowiedź HTTP zawiera nagłówek „content type", określający bezpieczny zestaw znaków (np. UTF-8, ISO 8859-1).	MUST	MUST	MUST
10.3	Zweryfikuj, czy w nagłówki HTTP dodawane przez zaufane proxy lub urządzenia SSO, takie jak token okaziciela, są uwierzytelniane przez aplikację.	OPTIONAL	MUST	MUST
10.4	Zweryfikuj, czy właściwy nagłówek X-FRAME-OPTIONS jest używany dla witryn, których zawartość nie powinna być wyświetlana w ramkach innych serwisów.	OPTIONAL	MUST	MUST
10.5	Zweryfikuj, czy nagłówki HTTP lub jakakolwiek część odpowiedzi HTTP nie ujawniają szczegółowej informacji o wersjach komponentów systemu.	MUST	MUST	MUST
10.6	Zweryfikuj, czy wszystkie odpowiedzi z API zawierają nagłówki X-Content-Type-Options: nosniff i ContentDisposition: attachment; filename="api.json" (lub inne nazwy plików odpowiednie dla jego typu).	MUST	MUST	MUST
10.7	Zweryfikuj, czy mechanizm Content Security Policy V2 (CSP) jest używany aby zapobiegać podatnościom wstrzyknięcia DOM, XSS, JSON i JavaScript	MUST	MUST	MUST
10.8	Zweryfikuj, czy nagłówek X-XSS-Protection: 1; mode=block jest użyty by uruchomić w przeglądarce filtry chroniące przed reflected XSS	MUST	MUST	MUST
11	**Wymagania weryfikacji zabezpieczeń przed złośliwym kodem** Upewnij się, że weryfikowana aplikacja w odpowiedni sposób spełnia poniższe ogólne wymagania: • złośliwe czynności są obsługiwane w sposób bezpieczny i poprawny, tak aby nie wywrzeć wpływu na resztę aplikacji, • aplikacja nie ma wbudowanych "bomb czasowych" ani innych ataków bazujących na czasie, • aplikacja nie próbuje się kontaktować ze złośliwymi lub nieautoryzowanymi lokalizacjami, • aplikacja nie ma tylnych furtek, jajek z niespodzianką (easter egg), ataków typu salami czy błędów logicznych, które mogłyby być kontrolowane przez atakującego. Złośliwy kod jest niezmiernie rzadki i trudno do wykryć. Ręczna analiza kodu linia po linii może wspomóc wykrycie bomb logicznych, ale nawet najbardziej doświadczony analityk z trudem go wykryje - nawet gdy będzie wiedział o jego istnieniu. Nie uda się spełnić wymagań tej sekcji bez dostępu do kodu źródłowego, włączając w to tak dużo zewnętrznych bibliotek jak tylko możliwe.			
11.1	Zweryfikuj, czy wszystkie złośliwe działania są odpowiednio sandboxowane, konteneryzowane lub izolowane aby opóźnić i powstrzymać atakujących przed atakami na inne aplikacje.	OPTIONAL	OPTIONAL	MUST

Centrum e-Zdrowia
ul. Stanisława Dubois 5A
00-184 Warszawa
tel.: +48 22 597-09-27
fax: +48 22 597-09-37
biuro@cez.gov.pl | www.cez.gov.pl
NIP: 5251575309
REGON: 001377706

11.2	Zweryfikuj kod źródłowy aplikacji oraz tyle bibliotek dostarczonych przez strony trzecie, ile to tylko możliwe i potwierdź, że nie zawierają złośliwego kodu, tylnych furtek, ukrytych niespodzianek (Easter eggs) oraz błędów logicznych w ramach uwierzytelniania, kontroli dostępu, walidacji danych wejściowych i logice biznesowej dla transakcji wysokiej wartości.	OPTIONAL	OPTIONAL	MUST
12	**Wymagania weryfikacji logiki biznesowej** Upewnij się, że weryfikowana aplikacja w odpowiedni sposób spełnia poniższe ogólne wymagania: • przepływ logiki biznesowej jest sekwencyjny i uporządkowany, • logika biznesowa zawiera ograniczania pozwalające na wykrywanie i zapobieganie zautomatyzowanym atakom, • przepływy logiki biznesowej o wysokiej wartości biorą pod uwagę nadużycia czy nieuczciwe osoby i tym samym mają wbudowane zabezpieczenia przed podszywaniem się, modyfikowaniem danych, zaprzeczaniem, wyciekiem informacji i eskalacją uprawnień.			
12.1	Zweryfikuj, czy aplikacja zezwala na przetwarzanie procesów logiki biznesowej jedynie krok po kroku, wszystkie kroki w procesie przetwarzane są w czasie jaki zajmuje to człowiekowi, kroki ustawione z złej kolejności nie są przetwarzane także nie pozwala się na pomijanie kroków, użycie kroków z procesów innych użytkowników i zbyt szybkie wysłanie operacji.	OPTIONAL	MUST	MUST
12.2	Zweryfikuj, czy aplikacja posiada limity biznesowe i prawidłowo wymusza je dla każdego użytkownika, a także czy można zdefiniować powiadomienia oraz automatyczną odpowiedź na ataki automatyczne bądź nietypowe operacje.	OPTIONAL	MUST	MUST
13	**Wymagania weryfikacji dla plików i innych zasobów** Upewnij się, że weryfikowana aplikacja w odpowiedni sposób spełnia poniższe ogólne wymagania: • niezaufane dane z plików powinny być obsługiwane w sposób poprawny i bezpieczny, • pliki źródłowe otrzymane z niezaufanych źródeł są przechowywane poza katalogiem główny aplikacji (webroot) i z ograniczonymi uprawnieniami.			
13.1	Zweryfikuj, czy przeadresowania i przekierowania URL są dozwolone jedynie dla predefiniowanych (white label) stron internetowych i pokazują użytkownikowi ostrzeżenie w sytuacji przekierowania do potencjalnie niezaufanych treści.	MUST	MUST	MUST
13.2	Zweryfikuj, czy niezaufane dane z plików wysłanych do aplikacji nie są bezpośrednio przekazywane do komend wykonujących operacje na drzewie plików, w szczególności by zapewnić	MUST	MUST	MUST

Centrum e-Zdrowia
ul. Stanisława Dubois 5A
00-184 Warszawa
tel.: +48 22 597-09-27
fax: +48 22 597-09-37
biuro@cez.gov.pl | www.cez.gov.pl
NIP: 5251575309
REGON: 001377706

	ochronę przed podatnościami typu „Path Traversal”, „Local File Inclusion” i „OS command injection”.			
13.3	Zweryfikuj, czy pliki pozyskane z niezaufanych źródeł są walidowane pod kątem oczekiwanego typu pliku i czy są skanowane programem antywirusowym w celu ochrony przed szkodliwą zawartością.	MUST	MUST	MUST
13.4	Zweryfikuj, czy niezaufane dane nie są użyte bezpośrednio w mechanizmach ładowania, ładowania klas i zwracania, by zapewnić ochronę przed atakami typu „Remote/Local File Inclusion”.	MUST	MUST	MUST
13.5	Zweryfikuj, czy niezaufane dane nie są wykorzystywane w ramach mechanizmu „Cross-Origin Resource Sharing” (CORS), w celu zapewnienia ochrony przed potencjalnie złośliwymi zewnętrznymi treściami.	MUST	MUST	MUST
13.6	Zweryfikuj, czy pliki otrzymane z nieznanych źródeł są umieszczone poza katalogiem głównym aplikacji (webroot), z minimalnymi uprawnieniami i z zalecaną silną walidacją.	OPTIONAL	MUST	MUST
13.7	Zweryfikuj, czy serwer webowy lub aplikacyjny są domyślnie skonfigurowane tak, aby odrzucać połączenia do zewnętrznych zasobów lub systemów poza serwerem www lub aplikacyjnym.	OPTIONAL	MUST	MUST
13.8	Zweryfikuj, czy kod aplikacji nie wykonuje danych otrzymanych z niezaufanych źródeł.	MUST	MUST	MUST
13.9	Nie używaj Flasha, Active-X, Silverlighta, NACL, appletów Java ani innej technologii klienckiej, która nie jest natywnie wspierana przez standardy W3C w przeglądarkach.	MUST	MUST	MUST
14	**Wymagania weryfikacji dla aplikacji mobilnych** Ta sekcja zawiera zabezpieczenia specyficzne dla aplikacji mobilnych. Aplikacje mobilne powinny: • mieć taki sam poziom zabezpieczeń w mobilnym kliencie jak na serwerze, poprzez wymuszanie mechanizmów kontrolnych w bezpiecznym środowisku, • przechowywanie informacji wymagających ochrony na urządzeniach powinno być realizowane w sposób bezpieczny, • transmitowanie wszystkich danych wymagających ochrony z urządzeń powinno być realizowane z myślą o bezpieczeństwie warstwy transmisji.			
14.1	Zweryfikuj, czy wartości identyfikatorów, takie jak UDID czy IMEI, przechowywanych na urządzeniu i odzyskiwanych przez inne aplikacje, nie są wykorzystywane jako tokeny uwierzytelnienia.	MUST	MUST	MUST
14.2	Zweryfikuj, czy aplikacja mobilna nie przechowuje wrażliwych danych na potencjalnie niezaszyfrowanych zasobach	MUST	MUST	MUST

Centrum e-Zdrowia
ul. Stanisława Dubois 5A
00-184 Warszawa
tel.: +48 22 597-09-27
fax: +48 22 597-09-37
biuro@cez.gov.pl | www.cez.gov.pl
NIP: 5251575309
REGON: 001377706

	współdzielonych na urządzeniu (karta SD, współdzielone foldery).			
14.3	Zweryfikuj, czy wrażliwe dane nie są przechowywane na urządzeniu w sposób niezabezpieczony, nawet w obszarach chronionych systemowo, takich jak pęki kluczy (key chains).	MUST	MUST	MUST
14.4	Zweryfikuj, czy klucze, tokeny API lub hasła są generowane dynamicznie przez aplikację.	OPTIONAL	MUST	MUST
14.5	Zweryfikuj, czy aplikacja zapobiega wyciekom poufnych informacji (np. zrzuty ekranów zapisują się w aktualnej aplikacji podczas gdy aplikacja działa w tle lub zapisuje wrażliwe informacje w konsoli).	OPTIONAL	MUST	MUST
14.6	Zweryfikuj, czy aplikacja wymaga minimalnych uprawnień dla wymaganych funkcjonalności i zasobów.	OPTIONAL	MUST	MUST
14.7	Zweryfikuj, czy wrażliwy kod aplikacji umieszczany jest w pamięci w sposób nieprzewidywalny (np. wykorzystując ASLR).	MUST	MUST	MUST
14.8	Zweryfikuj, czy obecne techniki zapobiegające debugowaniu są wystarczające do zatrzymania lub opóźnienia potencjalnych atakujących przed wstrzyknięciem debuggerów do aplikacji (np. GDB).	OPTIONAL	OPTIONAL	MUST
14.9	Zweryfikuj, czy aplikacja nie eksportuje wrażliwych danych w obrębie androidowych mechanizmów „activities”, „intents”, „content providers” itp. innym aplikacjom na tym samym urządzeniu.	MUST	MUST	MUST
14.10	Zweryfikuj, czy dane wymagające ochrony znajdujące się w pamięci, są nadpisywane zerami gdy tylko przestaną być niezbędne, aby zabezpieczyć się przed atakami zrzucania pamięci (memory dumping).	OPTIONAL	OPTIONAL	MUST
14.11	Zweryfikuj, czy aplikacja waliduje poprawność danych wejściowych dla eksportowanych czynności, intencji i dostarczycieli treści.	MUST	MUST	MUST
15	**Wymagania weryfikacyjne dla web serwisów** Upewnij się, że weryfikowana aplikacja, która wykorzystuje usługi sieciowe RESTful lub SOAP ma: • adekwatne uwierzytelnianie, zarządzanie sesją i autoryzację wszystkich serwisów sieciowych, • walidację wszystkich parametrów wejściowych, które są transmitowane z mniej do bardziej zaufanych warstw,			

Centrum e-Zdrowia
ul. Stanisława Dubois 5A
00-184 Warszawa
tel.: +48 22 597-09-27
fax: +48 22 597-09-37
biuro@cez.gov.pl | www.cez.gov.pl
NIP: 5251575309
REGON: 001377706

	• podstawową interoperacyjność usług sieciowych SOAP, w celu promowania wykorzystywania API.			
15.1	Zweryfikuj, czy ten sam styl kodowania znaków jest wykorzystywany przez klienta i serwer	MUST	MUST	MUST
15.2	Zweryfikuj, czy dostęp do funkcji administracyjnych i zarządczych w ramach aplikacji jest ograniczony do wąskiej grupy administratorów.	MUST	MUST	MUST
15.3	Zweryfikuj, czy schemat XML lub JSON jest stosowany i czy jest weryfikowany przed zaakceptowaniem danych wejściowych.	MUST	MUST	MUST
15.4	Zweryfikuj, czy wszystkie dane wejściowe mają odpowiednie ograniczenia długości.	MUST	MUST	MUST
15.5	Zweryfikuj, czy usługi sieciowe oparte o SOAP są zgodne co najmniej z profilem podstawowym Web ServicesInteroperability (WS-I Basic Profile). W szczególności dotyczy to szyfrowania TLS	MUST	MUST	MUST
15.6	Zweryfikuj, czy uwierzytelnienie i autoryzacja dotycząca sesji są wykorzystywane. Proszę skorzystać z sekcji 2, 3 i 4 w celu uzyskania dalszych instrukcji. Należy unikać stosowania statycznych kluczy API i podobnych rozwiązań.	MUST	MUST	MUST
15.7	Zweryfikuj, czy usługi REST są chronione przed atakami typu Cross-Site Request Forgery wykorzystując co najmniej jeden mechanizm spośród weryfikacji ORIGIN, podwójnego wprowadzania szablonów ciasteczek (cookie pattern), CSRF nonces oraz sprawdzeń punktu odniesienia (referrer checks).	MUST	MUST	MUST
15.8	Zweryfikuj, czy usługi REST wyraźnie sprawdzają czy przychodzące wartości Content-Type są spodziewanymi, np. czy jest to application/xml lub application/json.	OPTIONAL	MUST	MUST
15.9	Zweryfikuj, czy treść wiadomości jest podpisana, aby zapewnić bezpieczną transmisję pomiędzy klientem a usługą, wykorzystując JSON Web Signing lub WS-Security dla żądań SOAP.	OPTIONAL	MUST	MUST
15.10	Zweryfikuj, czy nie istnieją alternatywne i mniej bezpieczne ścieżki dostępu.	OPTIONAL	MUST	MUST
16	**Wymagania weryfikacyjne dla procesu konfiguracji** Upewnij się, że weryfikowana aplikacja: • wykorzystuje aktualne biblioteki i platformy, • wykorzystuje bezpieczną konfigurację bazową,			

Centrum e-Zdrowia
ul. Stanisława Dubois 5A
00-184 Warszawa
tel.: +48 22 597-09-27
fax: +48 22 597-09-37
biuro@cez.gov.pl | www.cez.gov.pl
NIP: 5251575309
REGON: 001377706

	• jest wystarczająco utwardzona, tak aby zmiany zainicjowane przez użytkowników w konfiguracji bazowej niekoniecznie eksponowały lub tworzyły słabości bezpieczeństwa lub uszkadzały zasadnicze systemy.			
16.1	Wszystkie komponenty powinny być aktualne, z właściwymi ustawieniami dotyczącymi bezpieczeństwa i wersjami. To powinno uwzględniać: usunięcie niepotrzebnych wpisów konfiguracyjnych i folderów takich jak przykładowe aplikacje, dokumentację platformy oraz domyślnych bądź przykładowych użytkowników.	MUST	MUST	MUST
16.2	Komunikacja pomiędzy komponentami, na przykład pomiędzy serwerem aplikacyjnym a serwerem bazodanowym, powinna być szyfrowana, szczególnie w przypadku gdy komponenty znajdują się w różnych kontenerach lub na różnych systemach.	OPTIONAL	MUST	MUST
16.3	Komunikacja pomiędzy komponentami, na przykład pomiędzy serwerem aplikacyjnym a serwerem bazodanowym, powinna być uwierzytelniona z użyciem konta z najmniejszymi koniecznymi przywilejami.	OPTIONAL	MUST	MUST
16.4	Zweryfikuj, czy wdrożone aplikacja są należycie zabezpieczone poprzez mechanizmy typu sandbox, umieszczenie w kontenerach bądź izolowane w celu opóźnienia lub powstrzymania atakujących przed uzyskaniem dostępu do innych aplikacji.	OPTIONAL	MUST	MUST
16.5	Zweryfikuj, czy proces budowania oraz wdrażania oprogramowania jest wykonany w sposób bezpieczny.	OPTIONAL	MUST	MUST
16.6	Zweryfikuj, czy uprawnieni administratorzy mają możliwość sprawdzenia spójności wszystkich ustawień istotnych z punktu widzenia bezpieczeństwa, aby zapewnić, że nie były one modyfikowane w sposób nieupoważniony.	OPTIONAL	OPTIONAL	MUST
16.7	Zweryfikuj, czy wszystkie komponenty aplikacji są podpisane.	OPTIONAL	OPTIONAL	MUST
16.8	Zweryfikuj, czy zewnętrze biblioteki pochodzą z zaufanych repozytoriów.	OPTIONAL	OPTIONAL	MUST
16.9	Upewnij się, że w procesie budowania (built process) oprogramowania, włączone są wszystkie mechanizmy takie jak ASLR, DEP i inne związane z bezpieczeństwem.	OPTIONAL	OPTIONAL	MUST
16.10	Zweryfikuj czy wszystkie zasoby aplikacyjne, takie jak biblioteki JavaScript, arkusze stylów CSS i fonty webowe są hostowane przez aplikację, a nie przez CDN (Content Delivery Network) lub zewnętrznych dostawców.	OPTIONAL	OPTIONAL	MUST

Centrum e-Zdrowia
ul. Stanisława Dubois 5A
00-184 Warszawa
tel.: +48 22 597-09-27
fax: +48 22 597-09-37
biuro@cez.gov.pl | www.cez.gov.pl
NIP: 5251575309
REGON: 001377706