eRecruiter

Data aktualizacji
30.07.2024

Bezpieczeństwo aplikacji system.eRecruiter.pl

erecruiter.pl

Spis treści

Wymagane zabezpieczenia zostały opracowane zgodnie ze standardem OWASP ASVS i najlepszymi praktykami bezpieczeństwa aplikacji.

1. Wymagania dotyczące architektury

Lp.	Opis zabezpieczenia	Spełnione / niespełnione	Komentarz
1.1	Architektura aplikacji składa się z co najmniej trzech warstw: • warstwy danych (baza danych) • warstwy logiki biznesowej (aplikacyjna) • warstwy prezentacyjnej (interfejs użytkownika)	Spełnione	
1.2	Poszczególne komponenty architektury aplikacji odseparowane są od siebie poprzez zdefiniowane zabezpieczenia tj. segmentacja sieci, reguły zapory sieciowej czy grupy dostępu w usługach chmurowych.	Spełnione	
1.3	W komunikacji używane są tylko uzasadnione porty i protokoły służące do komunikacji pomiędzy komponentami architektury.	Spełnione	
1.4	Wszystkie komponenty aplikacji, biblioteki, moduły, platformy i systemy operacyjne są regularnie monitorowane pod kątem podatności i wolne od znanych podatności.	Spełnione	
1.5	Kod aplikacji nie zawiera wrażliwych informacji z zakresu logiki biznesowej, sekretnych kluczy i innych zastrzeżonych informacji.	Spełnione	
1.6	Dostęp do danych ograniczony jest na poziomie bazy danych, a nie aplikacji. Aplikacja nie komunikuje się z bazą na uprawnieniach DBA.	Spełnione	
1.7	Aplikacja nie wykorzystuje protokołów uważanych powszechnie za niebezpieczne (np. ftp, NFS).	Spełnione	
1.8	Na styku z Internetem Aplikacja jest zabezpieczona brzegowym firewallem	Spełnione	
1.9	Logi systemowe i aplikacyjne są monitorowane i przechowywane przez system SIEM.	Spełnione	
1.10	Aplikacja poddawana jest regularnym testom penetracyjnym, a znalezione podatności natychmiast poprawiane. Ostatni test penetracyjny Aplikacji wraz z dokumentacją wykonania testu miał miejsce nie dawniej niż rok przed podpisaniem umowy oraz wszystkie znalezione podatności zostały usunięte.	Spełnione	.

2. Wymagania dotyczące uwierzytelniania

Lp.	Opis zabezpieczenia	Spełnione / niespełnione	Komentarz
2.1	Wszystkie strony oraz zasoby standardowo wymagają uwierzytelnienia, za wyjątkiem tych, które mają być dostępne publicznie (zasada pełnej mediacji).	Spełnione	
2.2	Wszystkie uwierzytelniania użytkowników realizowane są po stronie serwera.	Spełnione	
2.3	Wszystkie mechanizmy uwierzytelniania, które kończą pracę niepowodzeniem, robią to w sposób bezpieczny, aby mieć pewność, że atakujący nie może się zalogować.	Spełnione	Po 5 próbach nieprawidłowego logowania hasło ulega blokadzie na 30 minut. Nieudane próby są logowane w systemie.
2.4	Pole do wprowadzania hasła jest przystosowane do obsługi haseł długich i złożonych oraz nie posiada zabezpieczenia chroniącego przed użyciem menedżera haseł.	Spełnione	
2.5	Wszystkie operacje dotyczące uwierzytelniania (takie jak rejestracja, aktualizacja profilu, przypomnienie loginu, przypomnienie hasła), które powodują odzyskanie dostępu do konta, posiadają przynajmniej takie same zabezpieczenia jak podstawowe mechanizmy uwierzytelniania.	Spełnione	
2.6	Funkcja zmiany hasła zawiera konieczność podania hasła obecnie używanego, nowego hasła oraz konieczność ponownego wpisania nowego hasła.	Spełnione	
2.7	Wszystkie próby logowania są przechowywane bez poufnych identyfikatorów sesji lub haseł.	Spełnione	
2.8	Hasła do kont są przechowywane z wykorzystaniem jednokierunkowej funkcji skrótu z solą i są wystarczająco skomplikowane, by się obronić przed atakami brute force oraz odtwarzaniem hasła z jego skrótu.	Spełnione	Wymagania dotyczące złożoności hasła są konfigurowalne przez administratora systemu.
2.9	Dane uwierzytelniające są przesyłane z wykorzystaniem zaszyfrowanego linku oraz wszystkie strony/funkcje, które wymagają, by użytkownik podał dane uwierzytelniające, wymagają takiego szyfrowanego linku.	Spełnione	
2.10	Mechanizm odzyskiwania hasła nie ujawnia dotychczasowych haseł i nowe hasło nie jest przesyłane w formie jawnej do użytkownika.	Spełnione	
2.11	Nie jest możliwa wykonanie enumeracji wykorzystując loginy użytkowników, funkcję resetowania hasła lub funkcjonalność przypomnienia nazwy użytkownika.	Spełnione	
2.12	Aplikacja i inne komponenty, z których korzysta aplikacja, nie używają domyślnych haseł (np. admin/password).	Spełnione	
2.13	Aplikacja posiada wdrożone mechanizmy przeciwdziałające automatyzacji, aby zapobiegać testowaniu ujawnionych danych uwierzytelniających, atakom brute force i atakom blokującym konta.	Spełnione	Po 5 próbach nieprawidłowego logowania hasło ulega blokadzie na 30 minut. Nieudane próby są logowane w systemie.

Lp.	Opis zabezpieczenia	Spełnione / niespełnione	Komentarz
2.14	Wszystkie dane uwierzytelniające służące do uzyskiwania dostępu do usług zewnętrznych względem aplikacji, są zaszyfrowane i przechowywane w zabezpieczonej lokalizacji (a nie w kodzie źródłowym).	Spełnione	
2.15	Aplikacja blokuje czasowo dostęp po min. 5 niewłaściwie wprowadzonych hasłach. Administrator posiada możliwość zablokowania użytkownika w szczególnych przypadkach.	Spełnione	
2.16	W przypadku, gdy Aplikacja wykorzystuje bazę sekretnych pytań, nie narusza przepisów dotyczących prywatności. Dodatkowo pytania są na tyle trudne, by w rzeczywistości chronić aplikację.	Nie dotyczy	Aplikacja nie wykorzystuje bazy sekretnych pytań.
2.17	Aplikacja blokuje ponowne wykorzystanie 5 poprzednio użytych haseł przez użytkownika.	Spełnione	
2.18	Aplikacja umożliwia wykorzystywanie dwuskładnikowego uwierzytelniania.	Niespełnione	Na życzenie klienta może być włączony filtr adresów IP zawężający dostęp do aplikacji do konkretnych adresów IP. Wymagane jest posiadanie jednego lub kilku stałych adresów IP po stronie klienta. Istnieje także możliwość integracji mechanizmu ADFS (przy wykorzystaniu protokołu SAML 2.0) klienta z aplikacją eRecruiter
2.19	Aplikacja nie pozwala na używanie prostych i popularnych haseł.	Spełnione	Wymagania dla haseł: Ma co najmniej 12 znaków Zawiera małe litery Zawiera duże litery Zawiera cyfry Zawiera znaki specjalne (np.: ?, !, @, #) Musi się zmieniać co 60 dni Nie może być takie samo jak 5 ostatnich haseł
2.20	Klucze bądź inne informacje uwierzytelniające nie są zapisane w kodzie źródłowym lub repozytoriach kodu Aplikacji.	Spełnione	
2.21	Interfejs administracyjny Aplikacji nie jest dostępny z poziomu sieci niezaufanych (np. z Internetu) lub jest zabezpieczony z wykorzystaniem metod podwójnej autentykacji.	Spełnione	Panel administratora dostępny jest na zasadach takich samych jak pozostałe części systemu i z wykorzystaniem tych samych metod autentykacji, co pozostali użytkownicy. Panel BackOffice/Extranet to jest dostępny tylko dla pracowników eRecruiter i dostępny tylko z poziomu naszej sieci firmowej (przez VPN lub z biura).

3. Wymagania dotyczące zarządzania sesją

Lp.	Opis zabezpieczenia	Spełnione / niespełnione	Komentarz
3.1	Aplikacja nie używa zmodyfikowanego managera sesji.	Spełnione	
3.2	Sesje są unieważniane po wylogowaniu się użytkownika.	Spełnione	
3.3	Sesje wygasają po określonym czasie bezczynności.	Spełnione	Po 60 minutach.
3.4	Wszystkie strony, do których dostęp wymaga uwierzytelnienia, zawierają łatwy i widoczny dostęp do funkcji wylogowania.	Spełnione	
3.5	Identyfikatory sesji nigdy nie są ujawniane w URL, w komunikatach błędów lub logach oraz aplikacja nie wspiera przepisywania w URL ciasteczek sesyjnych (ang. URL-rewriting).	Spełnione	
3.6	Każde pomyślne uwierzytelnienie, a także ponowne uwierzytelnienie, tworzy nową sesję	Spełnione	
3.7	Identyfikatory sesji wygenerowane przez platformę aplikacji są uznawane przez nią za aktywne.	Spełnione	
3.8	Identyfikatory sesji są wystarczająco długie (min 128 bitów), losowe i unikalne w obrębie odpowiedniej aktywnej bazy sesji.	Spełnione	Tokeny sesji mają długość 32 bajtów
3.9	Identyfikatory sesji przechowywane w ciasteczkach mają ustawioną ścieżkę z wartością odpowiednio restrykcyjną dla danej aplikacji i tokeny sesji uwierzytelniania mają dodatkowo ustawione atrybuty "HttpOnly" i "secure".	Spełnione	
3.10	Aplikacja ogranicza liczbę jednoczesnych aktywnych sesji.	Spełnione	
3.11	Lista aktywnych sesji jest wyświetlana dla każdego użytkownika w profilu konta lub podobnym. Użytkownik powinien posiadać możliwość zakończenia dowolnej aktywnej sesji.	Nie dotyczy	Aplikacja nie pozwala na logowanie się na różnych urządzeniach na to samo konto. Ponowne zalogowanie zamyka poprzednią aktywną sesję, jeżeli taka istnieje.
3.12	Użytkownik ma zaproponowaną opcję zakończenia wszystkich innych aktywnych sesji, po pomyślnym zakończeniu procesu zmiany hasła.	Spełnione	

4. Wymagania dotyczące kontroli dostępu

Lp.	Opis zabezpieczenia	Spełnione / niespełnione	Komentarz
4.1	Wdrożona zasada minimalnych uprawnień - użytkownicy powinni mieć dostęp tylko do funkcji, plików, linków URL, usług oraz innych zasobów, do których posiadają zezwolenie.	Spełnione	
4.2	Dostęp do wrażliwych zapisów jest chroniony w taki sposób, aby tylko autoryzowane obiekty lub dane były dostępne dla użytkownika (np. ochrona przed ingerowaniem użytkowników w parametry umożliwiające zobaczenie lub dokonywanie zmian na koncie innego użytkownika).	Spełnione	
4.3	Listowanie katalogów jest wyłączone (chyba, że jest celowo dozwolone). Dodatkowo aplikacje nie powinny pozwalać na odkrycie lub ujawnienie plików lub metadanych katalogów takich jak Thumbs.db, .DS_Store, .git lub foldery .svn.	Spełnione	
4.4	Mechanizmy kontroli dostępu, które kończą pracę niepowodzeniem, robią to w sposób bezpieczny (np. strona z błędem nie wyświetla żadnych szczegółów).	Spełnione	
4.5	Reguły kontroli dostępu występujące w warstwie prezentacji są również egzekwowane po stronie serwera.	Spełnione	
4.6	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.	Spełnione	
4.7	Wszystkie decyzje dotyczące kontroli dostępu mogą być logowane, a wszystkie decyzje zakończone niepowodzeniem są logowane.	Spełnione	
4.8	Aplikacja bądź platforma generuje silne, losowe tokeny zabezpieczające przed atakami typu CSRF lub posiada inne mechanizmy ochrony transakcji.	Spełnione	
4.9	System ma wdrożone mechanizmy ochrony przed masowym lub ciągłym dostępem do zabezpieczonych funkcji, zasobów, bądź danych, na przykład mechanizmy zapobiegające odczytaniu wszystkich danych przez jednego użytkownika w sposób zautomatyzowany (tzw. „scrapping").	Niespełnione	System nie posiada takich mechanizmów, jednak akcje wykonywane na kandydacie, jak np. podgląd/pobranie cv są zapisywane w logach aplikacji.
4.10	Aplikacja w poprawny sposób wymusza autoryzację zależną od kontekstu, tak aby nie umożliwiać nieautoryzowanych manipulacji realizowanych poprzez zmiany wartości parametrów.	Spełnione	

5. Wymagania dotyczące obsługi złośliwych danych wejściowych

Lp.	Opis zabezpieczenia	Spełnione / niespełnione	Komentarz
5.1	Aplikacja wykorzystuje jeden mechanizm walidacji danych wejściowych dla każdego typu przyjmowanych danych.	Spełnione	
5.2	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.	Spełnione	
5.3	Aplikacja nie jest podatna na atak LDAP Injection lub mechanizmy bezpieczeństwa zapobiegają wystąpieniu takiego ataku.	Nie dotyczy	Aplikacja nie korzysta z protokołu Lightweight Directory Access.
5.4	Środowisko uruchomieniowe nie jest podatne na atak wstrzyknięcia komend systemu operacyjnego lub mechanizmy bezpieczeństwa zapobiegają wystąpieniu takiego ataku.	Spełnione	
5.5	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.	Spełnione	
5.6	Aplikacja nie jest podatna na ataki XML Injection, XML External Entity, XPath query lub mechanizmy bezpieczeństwa zapobiegają wystąpieniu takich ataków.	Spełnione	
5.7	Wszystkie łańcuchy zmiennych, włączane w HTML lub inny kod uruchamiany po stronie klienta webowego, są w poprawny sposób ręcznie wprowadzone zależnie 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.	Spełnione	
5.8	Jeżeli platforma aplikacji pozwala na masowe przypisywanie parametrów (mass assignment), zwane także automatycznym wiązaniem zmiennych (automatic variable binding) z przychodzącego żądania do modelu, wówczas pola (np. takie jak "stan_konta", "rola", "password") są chronione przed złośliwym automatycznym wiązaniem.	Spełnione	
5.9	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.)	Spełnione	
5.10	Walidacja po stronie klienta jest wykorzystywana jako druga linia obrony, będąca uzupełnieniem walidacji wykonywanej po stronie serwera.	Spełnione	
5.11	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.	Spełnione	

Lp.	Opis zabezpieczenia	Spełnione / niespełnione	Komentarz
5.12	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).	Spełnione	
5.13	Niestrukturalne dane są wyczyszczone i zawierają dozwolone znaki oraz długość, a także wszystkie znaki potencjalnie szkodliwe dla danego kontekstu powinny zostać usunięte (np. nazwy lokalne pisane w Unicode oraz apostrofy, jak w słowach ねこ czy O'Hara).	Spełnione	
5.14	Niezaufany kod HTML z edytorów WYSIWYG lub podobnych jest wyczyszczony oraz obsługiwany odpowiednio w zakresie walidacji wejścia i enkodowania wyjścia.	Spełnione	
5.15	W przypadku korzystania z technologii szablonów z funkcją automatycznego cytowania, gdy wyłączone jest cytowanie przez UI, zapewnione jest należyte czyszczenie kodu HTML.	Nie dotyczy	Aplikacja nie posiada funkcji automatycznego cytowania. Wszystkie szablony wiadomości są czyszczone z potencjalnie niebezpiecznej zawartości
5.16	Przy parsowaniu JSON w przeglądarce, metoda JSON.parse jest wykorzystywana do parsowania JSON na kliencie.	Spełnione	
5.17	Dane zapisywane po uwierzytelnieniu w pamięci przeglądarki np. w DOM są czyszczone po zakończeniu sesji.	Spełnione	
5.18	System wspiera przeglądarki internetowe Chrome, Microsoft Edge i Mozilla Firefox w najnowszych wersjach.	Spełnione	

6. Wymagania dotyczące nieaktywnych mechanizmów kryptograficznych

Lp.	Opis zabezpieczenia	Spełnione / niespełnione	Komentarz
6.1	Wszystkie moduły kryptograficzne, które kończą pracę niepowodzeniem, robią to w sposób bezpieczny, a błędy są obsługiwane w sposób zapobiegający atakom oracle padding.	Spełnione	
6.2	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.	Spełnione	
6.3	Wszystkie moduły kryptograficzne, wykorzystywane przez aplikację, zostały sprawdzone na zgodność ze standardem FISP 140-2 lub równorzędnym.	Spełnione	
6.4	Istnieje jasna polityka, określająca jak są zarządzane klucze kryptograficzne (np. jak są generowane, rozprowadzane, unieważniane, w jaki sposób wygasają).	Spełnione	
6.5	Dane osobowe powinny być przechowywane w postaci zaszyfrowanej i należy zapewnić, aby komunikacja odbywała się za pomocą zabezpieczonych kanałów.	Spełnione	
6.6	Hasła wymagające ochrony oraz klucze znajdujące się w pamięci są nadpisywane zerami, gdy tylko przestają być wymagane, aby zabezpieczyć się przed atakami zrzutu pamięci (memory dumping).	Spełnione	Wykonanie ataku zrzutu pamięci wymaga dostępu do serwera aplikacyjnego. Taki dostęp mają jedynie wyznaczone osoby. Dodatkowo taka pamięć jest natychmiastowo nadpisywana poprzez kolejne obiekty w pamięci.
6.7	Wszystkie klucze i hasła mogą być wymieniane i są generowane lub zmieniane podczas instalacji.	Nie dotyczy	Aplikacja dystrybuowana jest w modelu aplikacja jako usługa, w związku z tym u klienta nie występuje potrzeba instalacji.

7. Wymagania dotyczące obsługi i logowania błędów

Lp.	Opis zabezpieczenia	Spełnione / niespełnione	Komentarz
7.1	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.	Spełnione	
7.2	Logika zarządzania błędami w mechanizmach bezpieczeństwa domyślnie zabrania do nich dostępu.	Spełnione	
7.3	Mechanizmy logowania zdarzeń bezpieczeństwa zapewniają możliwość rejestracji zdarzeń istotnych z punktu widzenia bezpieczeństwa, zarówno tych zakończonych sukcesem jak i porażką.	Spełnione	Nie są logowane próby wejścia z niepoprawnego adresu IP (w przypadku konfiguracji zabezpieczenia wskazującego dozwoloną pulę adresów IP).
7.4	Log każdego zdarzenia zawiera niezbędną informację, która pozwoli na precyzyjną analizę czasową w przypadku wystąpienia zdarzenia.	Spełnione	
7.5	Logi bezpieczeństwa są chronione przed nieautoryzowanym dostępem i modyfikacją.	Spełnione	
7.6	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.	Niespełnione	Logowany jest adres IP dla IKK oraz Oceny Aplikacji, pozostałe dane wrażliwe nie są logowane.
7.7	Znaki niedrukowalne oraz separatory pól są prawidłowo zakodowane w rejestrach logów, w sposób zapobiegający wstrzykiwaniu logów.	Spełnione	
7.8	Logi audytowe lub podobne rejestry umożliwiają niezaprzeczalność kluczowych transakcji.	Spełnione	
7.9	Źródła czasu powinny być synchronizowane aby zapewnić, że logi mają poprawny czas.	Spełnione	

8. Wymagania dotyczące mechanizmów ochrony danych

Lp.	Opis zabezpieczenia	Spełnione / niespełnione	Komentarz
8.1	Wszystkie dane wrażliwe są przesyłane do serwera wewnątrz wiadomości HTTP lub jej nagłówka (np. parametry URL nie powinny być używane do przesyłania danych wrażliwych).	Spełnione	
8.2	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.	Spełnione	
8.3	Aplikacja minimalizuje ilość parametrów wysyłanych w zapytaniach takimi kanałami jak: ukryte pola, zmienne systemu AJAX, ciasteczka i nagłówki wiadomości.	Spełnione	
8.4	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).	Spełnione	
8.5	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.	Spełnione	

9. Wymagania dotyczące zabezpieczeń komunikacji

Lp.	Opis zabezpieczenia	Spełnione / niespełnione	Komentarz
9.1	Ważne certyfikaty SSL z zaufanego CA.	Spełnione	.
9.2	We wszystkich połączeniach (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. Wymagany jest najsilniejszy dostępny algorytm szyfrowania.	Spełnione	
9.3	Wszystkie połączenia do zewnętrznych systemów, które dotyczą wrażliwych informacji lub funkcji, są uwierzytelniane.	Spełnione	
9.4	Nagłówki HTTP Strict Transport Security są włączone do wszystkich zapytań i dla wszystkich poddomen np. w postaci: Strict-Transport-Security: max-age=15724800; includeSubdomains.	Spełnione	
9.5	W celu zmniejszenia prawdopodobieństwa pasywnych ataków, polegających na zapisywaniu ruchu, używane są szyfry wspierające doskonałe utajnienie przekazywania (forward secrecy).	Spełnione	
9.6	Włączone są i właściwie skonfigurowane mechanizmy unieważniania certyfikatów, takie jak „zszywanie" odpowiedzi Online Certificate Status Protocol (OSCP).	Spełnione	
9.7	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.	Spełnione	
9.8	Konfiguracja TLS jest zgodna z bieżącymi najlepszymi praktykami szczególnie dlatego, że popularne ustawienia, szyfry i algorytmy z czasem mogą okazać się niebezpieczne.	Spełnione	

10. Wymagania dotyczące konfigurowania http

Lp.	Opis zabezpieczenia	Spełnione / niespełnione	Komentarz
10.1	Aplikacja akceptuje tylko zdefiniowany zestaw metod zapytań http (takich jak GET, POST, PUT, DELETE) oraz nieużywane metody (takie jak TRACE) są w sposób wyraźny zablokowane.	Spełnione	
10.2	Każda odpowiedź HTTP zawiera nagłówek „content type", określający bezpieczny zestaw znaków (np. UTF-8, ISO 8859-1).	Spełnione	
10.3	Nagłówki HTTP dodawane przez zaufane proxy lub urządzenia SSO, takie jak token okaziciela, są uwierzytelniane przez aplikację.	Nie dotyczy	Aplikacja wymusza szyfrowanie https, w związku z tym niemożliwe jest dodanie do przesyłanej wiadomości nagłówków http.
10.4	Właściwy nagłówek X-FRAME-OPTIONS jest używany dla witryn, których zawartość nie powinna być wyświetlana w ramkach innych serwisów.	Spełnione	
10.5	Nagłówki HTTP lub jakakolwiek część odpowiedzi HTTP nie ujawniają szczegółowej informacji o wersjach komponentów systemu.	Spełnione	
10.6	Wszystkie odpowiedzi z API zawierają nagłówki określający typ pobieranego pliku.	Spełnione	
10.7	Mechanizm Content Security Policy V2 (CSP) jest używany, aby zapobiegać podatnościom wstrzyknięcia DOM, XSS, JSON i JavaScript	Spełnione	
10.8	Nagłówek X-XSS-Protection: 1; mode=block jest użyty, by uruchomić w przeglądarce filtry chroniące przed reflected XSS.	Niespełnione	Nie jest to podejście zgodne z aktualnymi wytycznymi: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-XSS-Protection

11. Wymagania dotyczące bezpieczeństwa plików i zasobów

Lp.	Opis zabezpieczenia	Spełnione / niespełnione	Komentarz
11.1	Przeadresowania i przekierowania URL są dozwolone jedynie dla predefiniowanych (white label) stron internetowych i pokazują użytkownikowi ostrzeżenie w sytuacji przekierowania do potencjalnie niezaufanych treści.	Spełnione	
11.2	Niezaufane dane z plików wysłanych do aplikacji nie są bezpośrednio przekazywane do komend wykonujących operacje na drzewie plików, w szczególności, by zapewnić ochronę przed podatnościami typu „Path Traversal", „Local File Inclusion" i „OS command injection".	Spełnione	
11.3	Pliki pozyskane z niezaufanych źródeł są walidowane pod kątem oczekiwanego typu pliku i są skanowane programem antywirusowym w celu ochrony przed szkodliwą zawartością.	Częściowo spełnione	Aplikacja nie ingeruje w dane przesyłane przez kandydatów za pomocą formularzy aplikacyjnych. Takie pliki są walidowane pod kątem typu plików (pliki wykonywalne jak np. .exe nie są dopuszczane), jednak nie są skanowane oprogramowanie antywirusowym. Rozpoznawanie pliku odbywa się zarówno po rozszerzeniu jak i zawartości pierwszych bajtów pliku.
11.4	Niezaufane dane nie są użyte bezpośrednio w mechanizmach ładowania, ładowania klas i zwracania, by zapewnić ochronę przed atakami typu „Remote/Local File Inclusion".	Spełnione	
11.5	Niezaufane dane nie są wykorzystywane w ramach mechanizmu „Cross-Origin Resource Sharing" (CORS), w celu zapewnienia ochrony przed potencjalnie złośliwymi zewnętrznymi treściami.	Spełnione	
11.6	Pliki otrzymane z nieznanych źródeł są umieszczone poza katalogiem głównym aplikacji (webroot) z minimalnymi uprawnieniami i z zalecaną silną walidacją.	Spełnione	
11.7	Serwer webowy lub aplikacyjny są domyślnie skonfigurowane tak, aby odrzucać połączenia do zewnętrznych zasobów lub systemów poza serwerem www lub aplikacyjnym.	Spełnione	
11.8	Kod aplikacji nie wykonuje danych otrzymanych z niezaufanych źródeł.	Spełnione	
11.9	Nie używane są Flash, Active-X, Silverlight, NACL oraz applety Java, które nie są natywnie wspierane przez standardy W3C w przeglądarkach.	Spełnione	
11.10	Dostawca jest w stanie precyzyjnie określić kraje, w których są przetwarzane dane.	Spełnione	Wykorzystywane centra danych znajdują się w EOG (Holandii, Irlandii oraz Polsce).

12. Wymagania dotyczące bezpieczeństwa webservice'ów

Lp.	Opis zabezpieczenia	Spełnione / niespełnione	Komentarz
12.1	Styl kodowania znaków jest wykorzystywany przez klienta i serwer.	Spełnione	
12.2	Dostęp do funkcji administracyjnych i zarządczych w ramach aplikacji jest ograniczony do wąskiej grupy administratorów.	Spełnione	
12.3	Schemat XML lub JSON jest stosowany i jest weryfikowany przed zaakceptowaniem danych wejściowych.	Spełnione	
12.4	Wszystkie dane wejściowe mają odpowiednie ograniczenia długości.	Spełnione	
12.5	Usługi sieciowe oparte o SOAP są zgodne co najmniej z profilem podstawowym Web Services-Interoperability (WS-I Basic Profile). W szczególności dotyczy to szyfrowania TLS.	Spełnione	
12.6	Uwierzytelnienie i autoryzacja dotycząca sesji są wykorzystywane. Należy unikać stosowania statycznych kluczy API i podobnych rozwiązań.	Spełnione	
12.7	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).	Spełnione	
12.8	Usługi REST wyraźnie sprawdzają czy przychodzące wartości Content-Type są spodziewanymi, np. czy jest to application/xml lub application/json.	Spełnione	
12.9	Treść wiadomości jest podpisana, aby zapewnić bezpieczną transmisję pomiędzy klientem a usługą, wykorzystując JSON Web Signing lub WS-Security dla żądań SOAP.	Spełnione	
12.10	Nie istnieją alternatywne i mniej bezpieczne ścieżki dostępu.	Spełnione	

13. Wymagania dotyczące bezpieczeństwa procesu konfiguracji

Lp.	Opis zabezpieczenia	Spełnione / niespełnione	Komentarz
13.1	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.	Spełnione	
13.2	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.	Spełnione	
13.3	Komunikacja pomiędzy komponentami, na przykład pomiędzy serwerem aplikacyjnym a serwerem bazodanowym, powinna być uwierzytelniona z użyciem konta z najmniejszymi koniecznymi przywilejami.	Spełnione	
13.4	Wdrożone aplikacje są należycie zabezpieczone poprzez mechanizmy typu sandbox, umieszczenie w kontenerach bądź izolowane w celu opóźnienia lub powstrzymania atakujących przed uzyskaniem dostępu do innych aplikacji.	Spełnione	
13.5	Proces budowania oraz wdrażania oprogramowania jest wykonany w sposób bezpieczny.	Spełnione	