Raport podatności

znalezionych w systemie skycms.pl

Testowany system: **SkyCMS**
Data wykonania testów: **01-31.10.2022**
Miejsce wykonania audytu: **Poznań**
Audytor: **Dawid Radziński**

Podsumowanie wykonanych prac

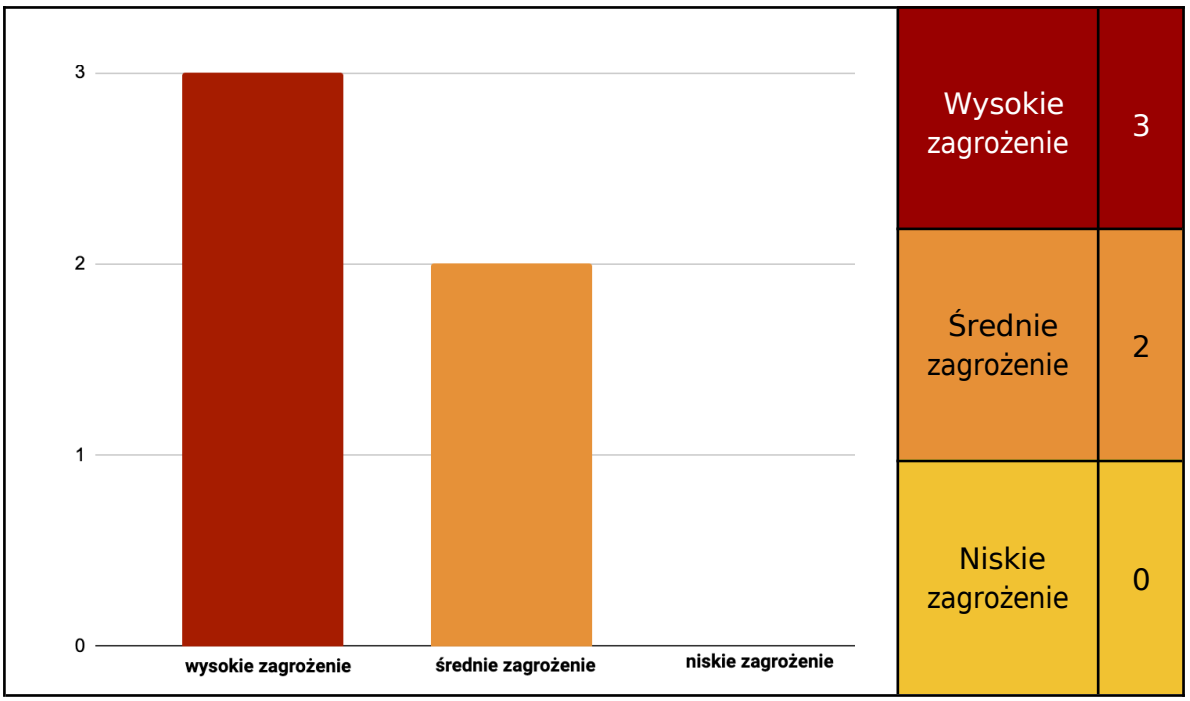
Niniejszy raport jest podsumowaniem testów bezpieczeństwa systemu SkyCMS, które zostały wykonane na przygotowanym przez klienta środowisku testowym, oddzielonym od wersji produkcyjnej aplikacji.

Audyt skupiał się na diagnozowaniu fragmentów systemu podatnych na przejęcie kontroli nad systemem SkyCMS (tj. wykonywanie dowolnego kodu po stronie serwera) przez napastnika nieposiadającego dodatkowych uprawnień do systemu.

W audycie opisane zostały dwa typy podatności dotyczące zarówno front-endu, jak i back-endu testowanego systemu. Są to:

- **Podatności typu XSS**,
- **Podatności typu SQL Injection**.

Wszystkie błędy i podatności zostały szczegółowo opisane w dalszej części raportu.



Lista znalezionych podatności

Podatności SQL Injection

Podatności typu SQL Injection ingerują w część aplikacji, która jest niewidoczna dla struktury HTML, ale odpowiada za całą dodatkowa funkcjonalność i działanie wszystkich akcji, które użytkownik wykonuje. Jest to tak zwany *back-end* danej witryny.
W nowoczesnych aplikacjach historia wszystkich działań i zmian wprowadzanych przez użytkowników jest przechowywana w tzw. *bazie danych.* Są tam zapisywane nie tylko informacje o każdej ich aktywności, ale również wszelkie dane dostępowe (loginy i hasła) zarówno zwykłych użytkowników, jak i administratorów danej aplikacji.

Ataki typu SQL Injection polegają na **wysłaniu do bazy danych konkretnego zapytania, które ma zmusić ją do wykonania akcji zdefiniowanej przez atakującego**. Najczęściej w ten sposób hakerzy starają się odczytać zawartość bazy danych, czyli wszystkich informacji, jakie są w niej przechowywane. W niektórych przypadkach **napastnik ma również możliwość dowolnej modyfikacji danych w niej zawartych**.

Następnie napastnik ma możliwość **przejęcia kont administratorów** danej aplikacji, co często wiąże się z dostaniem się na jej serwer, gdzie przechowywane są wszystkie pliki oraz kopie zapasowe strony. Gdy haker dostaje się do zasobów serwera, może zmusić serwer do wykonywania dowolnych niezamierzonych akcji. Jedną z nich może być wykorzystywanie zasobów serwera do *wydobywania kryptowalut,* użycie serwera jako bramki do maskowania IP. Napastnik może również wstrzyknąć w strukturę strony kod własnych reklam, umożliwiający zarabianie pieniędzy niechcianym firmom trzecim.

Numer podatności	**1**
Poziom niebezpieczeństwa	**wysoki**
Lokalizacja	/292/katalog-firm.html
Podatny parametr	sortKey=RAND(23613);SELECT%20SLEEP(5)
Wymagane uprawnienia	brak

Numer podatności	**2**
Poziom niebezpieczeństwa	**wysoki**
Lokalizacja	/1130/wyszukiwarka_zaawansowana.html
Podatny parametr	imie_nazwisko=Jan+Testowski+dfgdfgfd'%20AND %20(SELECT% 209270%20FROM %20(SELECT(SLEEP(7)))eLch)%20AND%20'x sZf'='xsZf
Wymagane uprawnienia	brak

Numer podatności	**3**
Poziom niebezpieczeństwa	**wysoki**
Lokalizacja	/1130/wyszukiwarka_zaawansowana.html
Podatny parametr	menu=274);SELECT%20SLEEP(9)--%20-
Wymagane uprawnienia	brak

Proof of concept	Aplikacja po wykonaniu tego requestu czeka określoną przez nas ilość sekund zgodnie z komendą audytora (komenda “SLEEP”). Udowadnia to, że baza danych wykonuje polecenia osoby testującej, której udało się z nią skomunikować pomimo braku uprawnień i stosownego dostępu do jej zawartości. Jest ona w ten sposób podatna na komendy dowolnego napastnika.
Opis	W aplikacji zidentyfikowano błędy SQL Injection, pozwalające napastnikom na pełny dostęp do bazy danych aplikacji SkyCMS. W konsekwencji wykorzystania tej podatności możliwy jest: • Dostęp do hashy haseł użytkowników i administratorów systemu, • Dostęp do danych wszystkich użytkowników systemu, • Dostęp do danych konfiguracyjnych bazy danych.
Rekomendacje naprawy	Zaleca się, by używać zapytań parametryzowanych we wszystkich miejscach, w których budowane są zapytania SQL. Jeżeli jednym z parametrów zapytania jest nazwa kolumny, zaleca się, by ją weryfikować względem listy dopuszczalnych kolumn lub weryfikować względem dopuszczalnych znaków (np. tylko znaki alfanumeryczne i znak podkreślenia), zaś w samym zapytaniu nazwy kolumn umieszczać wewnątrz backticków (znaków `).

Podatności typu XSS

Podatności typu XSS (ang. *Cross-site scripting*) charakteryzują się ingerencją w strukturę HTML oraz CSS danego systemu. Dają one zatem napastnikowi możliwość zmiany i przekonfigurowania tak zwanego *front-endu* danej aplikacji, czyli części widocznej dla użytkownika.

Jak to działa? Gdy użytkownik loguje się do aplikacji, specjalnie dla niego wygenerowana zostaje konkretna *sesja* (ang. *user session*), która odróżnia go od pozostałych użytkowników. W ten sposób serwer aplikacji rozpoznaje każdego z zalogowanych w danym momencie użytkowników. Za pomocą ataków typu XSS **haker może ukraść daną sesję użytkownika**, a dokładniej, jego *pliki cookie* lub *tokeny JWT.*
Dzięki temu zabiegowi **napastnik oszukuje serwer aplikacji, podszywając się pod dowolnego użytkownika** bez znajomości jego loginu i hasła.

Tego typu błędy umożliwiają też zmianę funkcjonalności aplikacji, zmuszając użytkownika do wykonania akcji, której wcale nie chciał wykonać, poprzez na przykład podmianę funkcjonalności przycisków, które ma on do dyspozycji (np. przycisk wystawienia komentarza lub przeglądania listy wysłanych wiadomości). Dodatkowo w niektórych przypadkach błędy XSS umożliwiają dodanie na stałe do struktury strony fragmentów kodu JavaScript, który może wykonywać akcje nieprzewidziane przez twórców danej aplikacji. Kod taki może na przykład wysyłać na zewnętrzny serwer dane, którymi użytkownik się dzieli w aplikacji lub przechwytywać wciśnięcia konkretnych klawiszy na klawiaturze.

Numer podatności	**4**
Poziom niebezpieczeństwa	**średni**
Lokalizacja	/1129/wyniki-wyszukiwania.html
Podatny parametr	Key=sad%22%3E%3Cscript%3Ealert(1)%3C/script%3E&Type=1&Page=1
Wymagane uprawnienia	brak

Numer podatności	**5**
Poziom niebezpieczeństwa	**średni**
Lokalizacja	/1130/wyszukiwarka_zaawansowana.html
Podatny parametr	zrodlo=podstrony%27){};alert(%27xss%27);if(%27a
Wymagane uprawnienia	brak

Proof of concept	http://demo.skycms.netk.pl/1129/wyniki-wyszukiwania.html?Key=sad%22%3E%3Cscript%3Ealert(1)%3C/script%3E&Type=1&Page=1 Po wejściu w powyższy link, na stronie pojawiał się alert wygenerowany przez napastnika.
Opis	Wszystkie wykryte podczas audytu ataki typu XSS były umieszczane przez audytora bezpośrednio w adres URL, po wejściu przez użytkownika w taki link, wstrzyknięty wcześniej przez napastnika kod JavaScript uruchamia się, dzięki czemu możliwa jest **kradzież sesji użytkownika**, lub wykonanie dowolnej komendy w języku JavaScript. Scenariusz ataku: 1. Haker wchodzi na daną stronę i modyfikuje link do danej lokalizacji w aplikacji, 2. Następnie wysyła zmodyfikowany link do użytkownika, 3. W efekcie napastnik może podszyć się pod użytkownika, który kliknął dany link. 4. W ten sposób haker uzyskuje dostęp do konta użytkownika, podszywając się pod niego.
Rekomendacje naprawy	Należy się zabezpieczyć przed wstrzyknięciami złośliwego kodu JavaScript poprzez użycie biblioteki wycinającej niebezpieczne znaczniki i atrybuty HTML. Przykładem takiej biblioteki działającej na poziomie JS jest DOMPurify (https://github.com/cure53/DOMPurify).