

Audyt bezpieczeństwa WackoPicko

W ramach projektu „Wykonanie audytu
bezpieczeństwa serwisu WWW z OWASP BWA.”

CBEK00013P

Zarządzanie infrastrukturą teleinformatyczną

K02-28a Wtorek TN, 13:15-15:00

Jakub Stuglik 259503

Bartłomiej Piotrowski 259492

Spis treści

Wstęp	4
1. Reflected XSS.....	5
Stopień niebezpieczeństwa	5
Definicja podatności	5
Opis działania podatności na stronie	5
Proof of concept.....	5
Jak się chronić?	6
2. Stored XSS.....	7
Stopień niebezpieczeństwa	7
Definicja podatności	7
Opis działania podatności na stronie	7
Proof of concept.....	7
Jak się chronić?	8
3. Session ID.....	9
Stopień niebezpieczeństwa	9
Definicja podatności	9
Opis działania podatności na stronie	9
Proof of concept.....	9
Jak się chronić?	11
4. Reflected SQL Injection	12
Stopień niebezpieczeństwa	12
Definicja podatności	12
Opis działania podatności na stronie	12
Proof of concept.....	12
Jak się chronić?	13
5. Parameter Manipulation	14
Stopień niebezpieczeństwa	14
Definicja podatności	14
Opis działania podatności na stronie	14
Proof of concept.....	14
Jak się chronić?	15
6. Reflected XSS Behind JavaScript.....	16
Stopień niebezpieczeństwa	16
Definicja podatności	16
Opis działania podatności na stronie	16

Proof of concept.....	16
Jak się chronić?.....	17
7. Słaby login/hasło	18
Stopień niebezpieczeństwa	18
Definicja podatności	18
Opis działania podatności na stronie	18
Proof of concept.....	18
Jak się chronić?.....	19
8. Stored SQL Injection	20
Stopień niebezpieczeństwa	20
Definicja podatności	20
Opis działania podatności na stronie	20
Proof of concept.....	20
Jak się chronić?.....	21
9. Multi-Step Stored XSS.....	22
Stopień niebezpieczeństwa	22
Definicja podatności	22
Opis działania podatności na stronie	22
Proof of concept.....	22
Jak się chronić?.....	24
10. Forceful Browsing.....	25
Stopień niebezpieczeństwa	25
Definicja podatności	25
Opis działania podatności na stronie	25
Proof of concept.....	25
Jak się chronić?.....	26
11. Logic Flaw	27
Stopień niebezpieczeństwa	27
Definicja podatności	27
Opis działania podatności na stronie	27
Proof of concept.....	27
Jak się chronić?.....	28
Podsumowanie.....	29

Wstęp

WackoPicko to strona internetowa zawierająca znane podatności i pozwoli nam przeprowadzić audyt bezpieczeństwa.

WackoPicko to strona do udostępniania i kupowania zdjęć. Typowy użytkownik WackoPicko może przysyłać zdjęcia, przeglądać zdjęcia innych użytkowników, komentować zdjęcia i kupować prawa do wersji wysokiej jakości zdjęcia.

- 1. Uwierzytelnianie** - WackoPicko dostarcza spersonalizowane treści zarejestrowanym użytkownikom. Pomimo ostatnich wysiłków na rzecz jednolitego logowania na wielu witrynach, większość aplikacji internetowych wymaga od użytkownika utworzenia konta, aby skorzystać z oferowanych usług. Dlatego WackoPicko ma system rejestracji użytkownika. Po utworzeniu konta użytkownik może się zalogować, aby uzyskać dostęp do ograniczonych funkcji WackoPicko.
- 2. Przesyłanie zdjęć** - Kiedy zarejestrowany użytkownik przesyła zdjęcie na WackoPicko, inni użytkownicy mogą je skomentować, a także kupić prawa do wersji wysokiej jakości.
- 3. Komentowanie zdjęć** - Po przesłaniu zdjęcia na WackoPicko, wszyscy zarejestrowani użytkownicy mogą skomentować zdjęcie, wypełniając formularz. Po utworzeniu komentarz jest wyświetlany razem ze zdjęciem, z pozostałymi komentarzami powiązanymi ze zdjęciem.
- 4. Kupowanie zdjęć** - Zarejestrowany użytkownik WackoPicko może kupić wersję zdjęcia w wysokiej jakości. Kupno przebiega w wielu etapach, podobnie jak w sklepach internetowych, przy użyciu koszyka zakupów. Po dodaniu zdjęć do koszyka, użytkownik może przejrzeć całkowitą cenę, zastosować kupony rabatowe i dokonać zamówienia. Po zakupie zdjęć użytkownik otrzymuje linki do wersji wysokiej jakości zdjęć.
- 5. Wyszukiwanie** - Aby ułatwić użytkownikom wyszukiwanie różnych zdjęć, WackoPicko zapewnia pasek wyszukiwania na górze każdej strony. Funkcja wyszukiwania korzysta z pola tagów, które zostało uzupełnione podczas przysyłania zdjęcia. Po wprowadzeniu zapytania użytkownik otrzymuje listę wszystkich zdjęć, które mają tagi pasujące do zapytania.
- 6. Księga gości** - Strona księgi gości umożliwia otrzymywanie opinii od wszystkich odwiedzających stronę WackoPicko. Formularz używany do przysyłania opinii zawiera pola "imię" i "komentarz".
- 7. Obszar administratora** - WackoPicko posiada specjalny obszar tylko dla administratorów, który ma inny mechanizm logowania niż dla zwykłych użytkowników. Administratorzy mogą wykonywać specjalne czynności, takie jak usuwanie kont użytkowników lub zmiana tagów zdjęcia.

Przed rozpoczęciem ręcznego sprawdzania strony internetowej przeprowadzono automatyczny skan za pomocą aplikacji OWASP ZAP, który został załączony do projektu.

1. Reflected XSS

Stopień niebezpieczeństwa
High

Definicja podatności

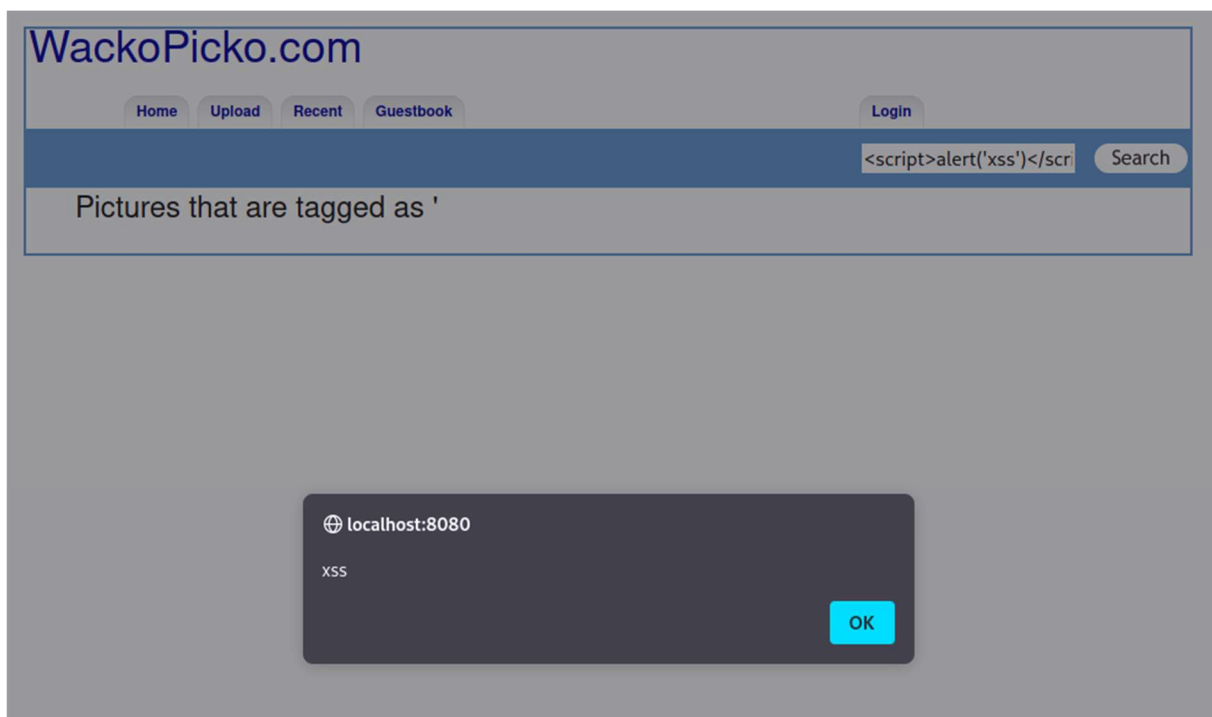
Reflected cross-site scripting (lub XSS) powstaje, gdy aplikacja odbiera dane w żądaniu HTTP i dołącza te dane do natychmiastowej odpowiedzi w niebezpieczny sposób.

Opis działania podatności na stronie

Na stronie wyszukiwania występuje luka XSS, która jest dostępna bez potrzeby logowania się do aplikacji. Parametr zapytania nie jest oczyszczany przed wysłaniem go do użytkownika. Obecność luki można zauważyć, ustawiając parametr zapytania na `<script>alert('xss')</script>`. Gdy ten skrypt zostanie odesłany do użytkownika, spowoduje to wyświetlenie przez przeglądarkę komunikatu. (Oczywiście osoba atakująca mogłaby wykorzystać tę lukę do wykonania złośliwej czynności zamiast wyświetlania komunikatu).

Proof of concept

W polu wyszukiwania wpisano skrypt (`<script>alert('oops');</script>`) i wciśnięto ENTER. Jak widać, treść nie została oczyszczona, a skrypt jest interpretowany przez przeglądarkę:



Zdjęcie 1

Jak się chronić?

Aby zabezpieczyć się przed atakami XSS typu reflected, należy przestrzegać kilku zasad:

- 1) Używać funkcji sanitize lub encode dla danych, które są wyświetlane na stronie.
- 2) Unikać przesyłania danych użytkownika bezpośrednio do kodu HTML lub JavaScript.
- 3) Zawsze sprawdzać dane wejściowe użytkownika pod kątem niedozwolonych znaków lub kodów.
- 4) Używać mechanizmów autoryzacji i uwierzytelniania, aby zabezpieczyć dostęp do danych wrażliwych.
- 5) Zawsze używać nowych wersji przeglądarek i aktualizować aplikacje, aby korzystać z najnowszych zabezpieczeń przeciwko atakom XSS.

2. Stored XSS

Stopień niebezpieczeństwa

High

Definicja podatności

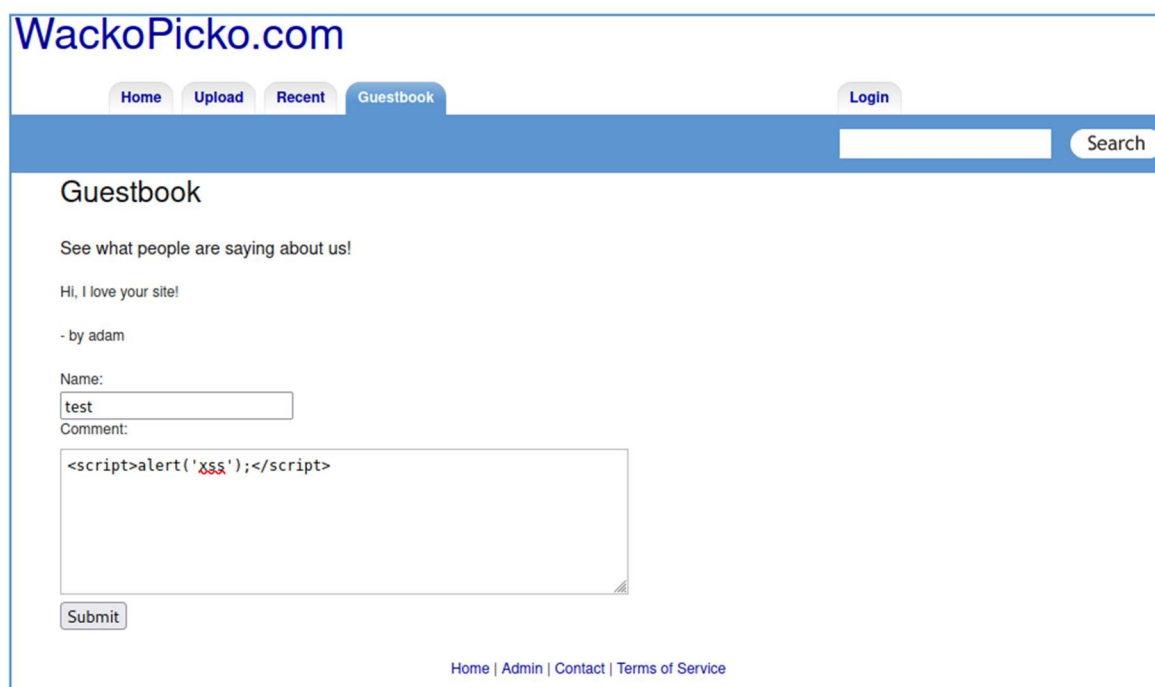
Stored cross-site scripting (znane również jako second-order XSS) powstaje, gdy aplikacja otrzymuje dane z niezaufanego źródła i dołącza te dane do swoich późniejszych odpowiedzi HTTP w niebezpieczny sposób.

Opis działania podatności na stronie

Istnieje luka stored XSS na stronie Guestbook. Pole komentarza nie jest odpowiednio oczyszczone, dlatego atakujący może wykorzystać tę lukę, tworząc komentarz zawierający kod JavaScript. Kiedy użytkownik odwiedza stronę księgi gości, atak zostanie uruchomiony i (być może złośliwy) kod JavaScript zostanie wykonany.

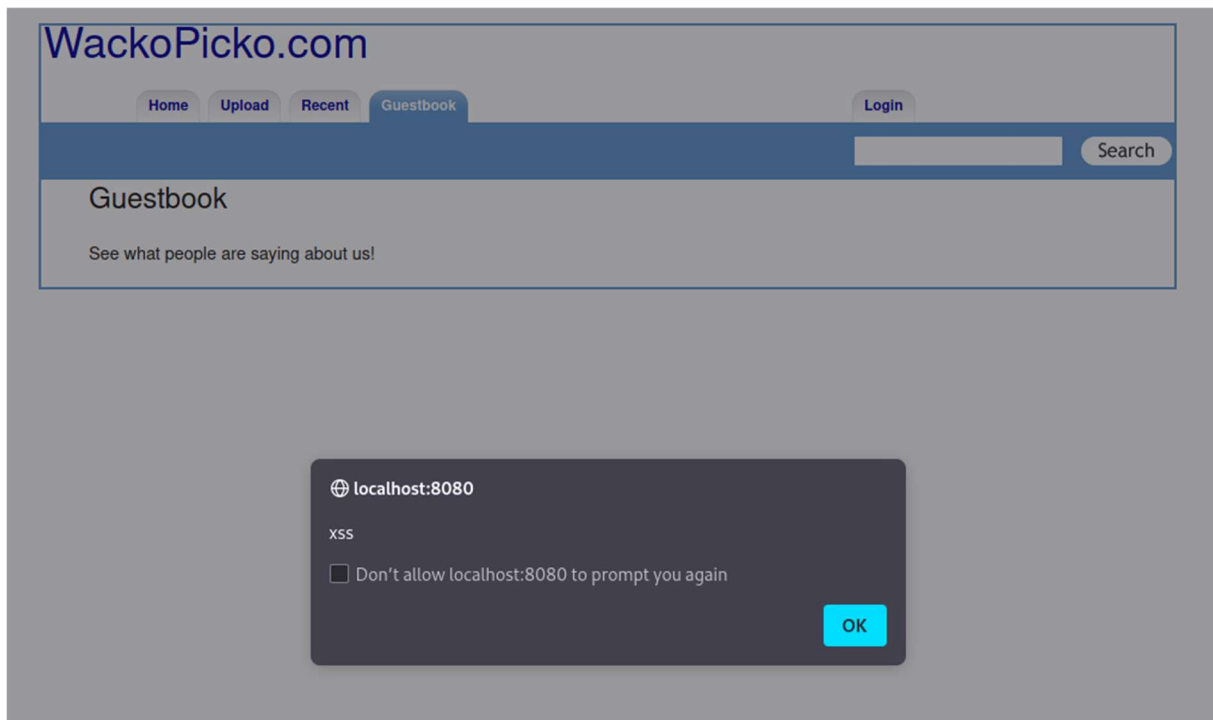
Proof of concept

W WackoPICKO kliknięto zakładkę "Guestbook" i wpisano `<script>alert('xss');</script>` w polu komentarza.



Zdjęcie 2

Kiedy formularz jest walidowany, zapisuje zawartość w bazie danych. Za każdym razem, gdy kliknie się na zakładkę „Guestbook”, pojawi się wyskakujące okienko, ponieważ przeglądarka wykona kod JavaScript zawarty w polu komentarzy.



Zdjęcie 3

Jak się chronić?

Aby zabezpieczyć się przed atakami XSS typu stored, należy przestrzegać kilku zasad:

- 1) Używać funkcji sanitize lub encode dla danych, które są zapisywane na serwerze, takie jak dane użytkownika, komentarze, itp.
- 2) Unikać przesyłania danych użytkownika bezpośrednio do kodu HTML lub JavaScript.
- 3) Zawsze sprawdzać dane wejściowe użytkownika pod kątem niedozwolonych znaków lub kodów.
- 4) Używać mechanizmów autoryzacji i uwierzytelniania, aby zabezpieczyć dostęp do danych wrażliwych.
- 5) Zawsze używać nowych wersji przeglądarek i aktualizować aplikacje, aby korzystać z najnowszych zabezpieczeń przeciwko atakom XSS.
- 6) Regularnie skanować swoje aplikacje pod kątem luk bezpieczeństwa.
- 7) Stosować zabezpieczenia przed stored XSS, takie jak whitelisting, czy izolacja skryptów odpowiedzialnych za przechowywanie danych.
- 8) Dbać o regularne aktualizowanie swoich systemów i aplikacji.

3. Session ID

Stopień niebezpieczeństwa
High

Definicja podatności

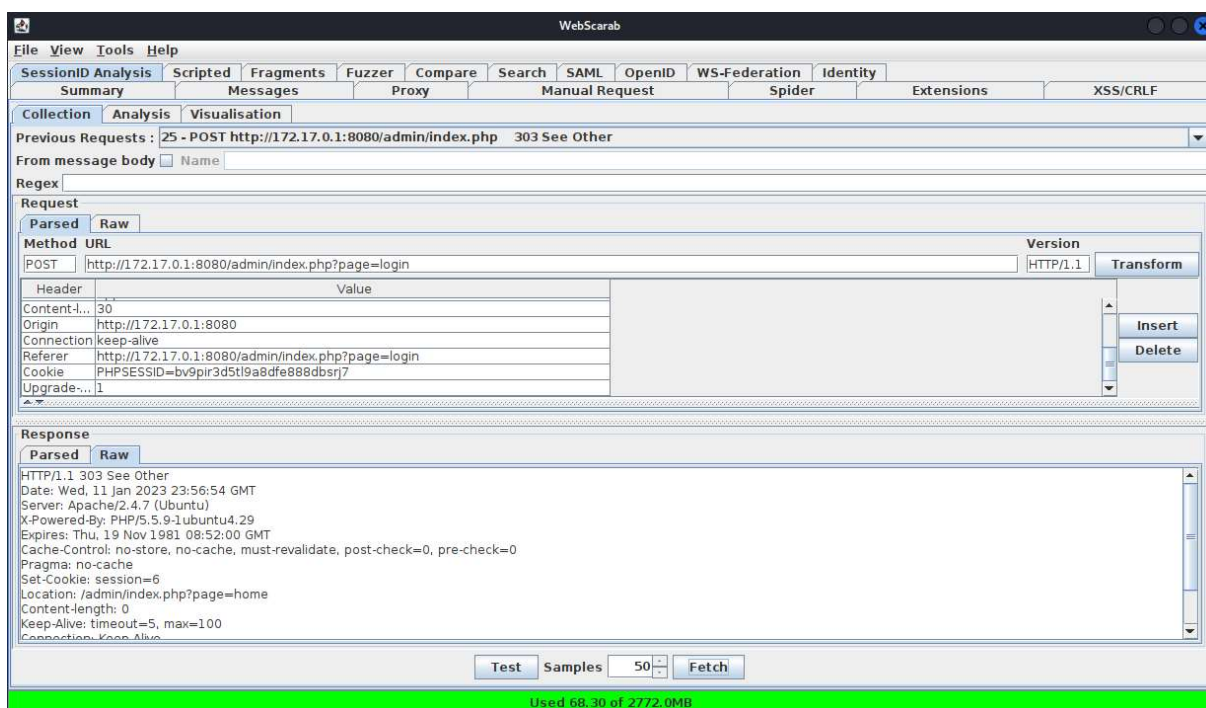
Session ID vulnerability to luka bezpieczeństwa, która pozwala atakującemu na przejęcie kontroli nad sesją użytkownika poprzez wykorzystanie unikalnego identyfikatora sesji przydzielonego użytkownikowi po zalogowaniu do systemu. Atakujący może go zdobyć wykorzystując różne metody, takie jak kradzież ciasteczek, phishing, lub sniffing sieci, a następnie wykorzystać go, aby przejąć sesję użytkownika.

Opis działania podatności na stronie

Aby przeanalizować rozkład generowanych identyfikatorów sesji dla logowania administratora, używamy WebScarab. Pokażemy, że rozkład identyfikatorów sesji jest stopniowy i z tego powodu przewidywalny.

Proof of concept

Analiza identyfikatorów sesji za pomocą WebScarab pozwala łatwo udowodnić, że identyfikator sesji (session=x) ma rozkład stopniowy w czasie:



Zdjęcie 4

WebScarab

File View Tools Help

Summary Messages Proxy Manual Request Spider Extensions XSS/CRLF SessionID Analysis Scripted Fragments Fuzzer Compare Search SAML OpenID WS-Federation Identity

Collection Analysis Visualisation

Session Identifier: 172.17.0.1/admin session

Date	Value	Numeric	Difference
2023/01/12 00:57:02.605	17	7	1
2023/01/12 00:57:02.661	18	8	1
2023/01/12 00:57:02.764	19	9	1
2023/01/12 00:57:02.859	20	10	1
2023/01/12 00:57:02.943	21	11	1
2023/01/12 00:57:03.44	22	12	1
2023/01/12 00:57:03.158	23	13	1
2023/01/12 00:57:03.249	24	14	1
2023/01/12 00:57:03.349	25	15	1
2023/01/12 00:57:03.451	26	16	1
2023/01/12 00:57:03.553	27	17	1
2023/01/12 00:57:03.657	28	18	1
2023/01/12 00:57:03.756	29	19	1
2023/01/12 00:57:03.856	30	20	1
2023/01/12 00:57:03.967	31	21	1
2023/01/12 00:57:04.59	32	22	1
2023/01/12 00:57:04.62	33	23	1
2023/01/12 00:57:04.190	34	24	1
2023/01/12 00:57:04.263	35	25	1
2023/01/12 00:57:04.377	36	26	1
2023/01/12 00:57:04.497	37	27	1
2023/01/12 00:57:04.668	38	28	1
2023/01/12 00:57:04.717	39	29	1
2023/01/12 00:57:04.767	40	30	1
2023/01/12 00:57:04.896	41	31	1
2023/01/12 00:57:04.978	42	32	1
2023/01/12 00:57:05.71	43	33	1
2023/01/12 00:57:05.172	44	34	1
2023/01/12 00:57:05.280	45	35	1
2023/01/12 00:57:05.380	46	36	1
2023/01/12 00:57:05.480	47	37	1
2023/01/12 00:57:05.582	48	38	1
2023/01/12 00:57:05.696	49	39	1
2023/01/12 00:57:05.805	50	40	1
2023/01/12 00:57:05.886	51	41	1
2023/01/12 00:57:06.3	52	42	1
2023/01/12 00:57:06.87	53	43	1
2023/01/12 00:57:06.191	54	44	1
2023/01/12 00:57:06.309	55	45	1
2023/01/12 00:57:06.432	56	46	1

Minimum: 0

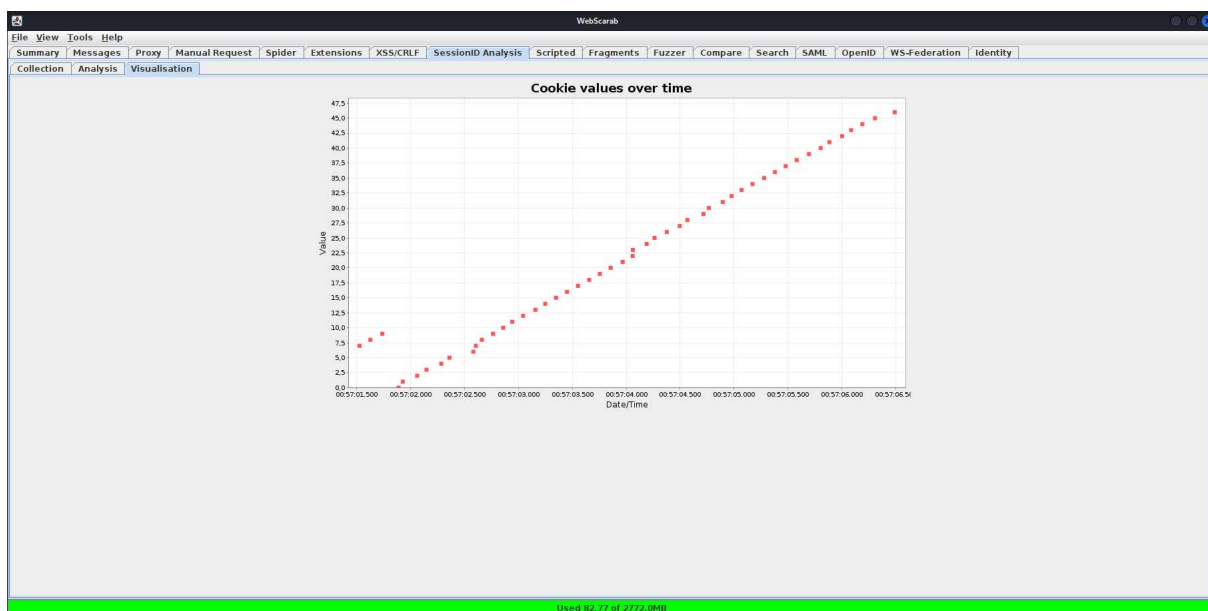
Maximum: 46

Range: 46.0

Clear Export

Used 78.93 of 2772.0MB

Zdjęcie 5



Zdjęcie 6

Jak się chronić?

Aby zabezpieczyć się przed atakami XSS typu reflected, należy przestrzegać kilku zasad:

- 1) Stosować bezpieczne metody transmisji danych, takie jak HTTPS, aby zabezpieczyć przesyłanie danych między przeglądarką a serwerem.
- 2) Stosować długie i losowe identyfikatory sesji, które są trudne do zgadnięcia przez atakującego.
- 3) Stosować mechanizmy uwierzytelniania wieloskładnikowego, takie jak hasła i tokeny, aby zwiększyć bezpieczeństwo sesji.
- 4) Regularnie monitorować swoje systemy i aplikacje w celu wykrycia anomalii.
- 5) W przypadku wykrycia ataku na identyfikator sesji, należy natychmiast unieważnić aktualne sesje i przydzielić nowe identyfikatory sesji.
- 6) Stosować mechanizmy, które pozwolą na blokowanie lub ograniczanie dostępu po kilku nieudanych próbach logowania.
- 7) Regularnie aktualizować swoje systemy i aplikacje, aby skorzystać z najnowszych zabezpieczeń.

4. Reflected SQL Injection

Stopień niebezpieczeństwa

High

Definicja podatności

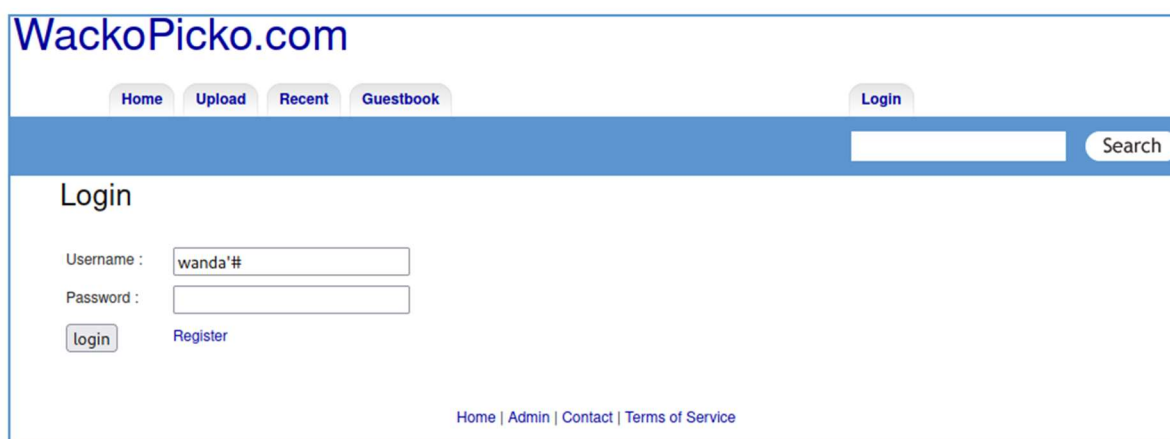
Reflected SQL injection to rodzaj ataku, w którym osoba atakująca jest w stanie wstrzyknąć złośliwy kod SQL do pola wejściowego aplikacji internetowej, a następnie wstrzyknięty kod jest natychmiast wykonywany przez bazę danych. Atakujący może następnie wykorzystać to do uzyskania nieautoryzowanego dostępu do poufnych informacji, a nawet zmodyfikowania lub usunięcia danych przechowywanych w bazie danych. Ten rodzaj ataku jest również znany jako „nietrwale” wstrzyknięcie, ponieważ złośliwy kod nie jest przechowywany w bazie danych, lecz odzwierciedlany z powrotem w odpowiedzi aplikacji skierowanej do użytkownika.

Opis działania podatności na stronie

WackoPicko zawiera podatność Reflected SQL Injection w polu nazwy użytkownika formularza logowania. Wprowadzając dodatkowe znaki w pole nazwy użytkownika można wykonać dowolne zapytanie w bazie danych i uzyskać np. loginy i hasła wszystkich użytkowników w systemie.

Proof of concept

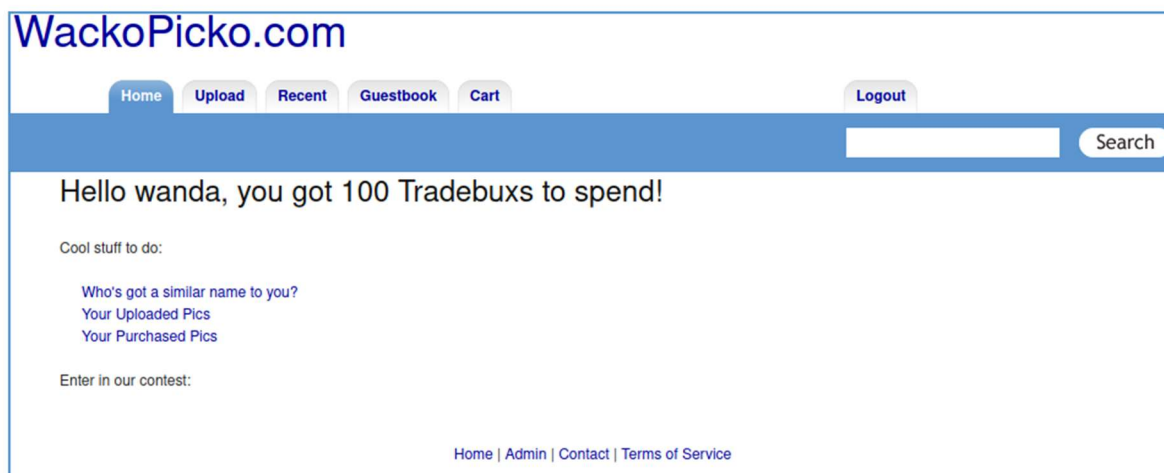
Wpisując login wraz ze znakami '#' komentowane jest to, co znajduje się po prawej stronie symbolu hash.



The screenshot shows the WackoPicko.com website. At the top, there is a navigation bar with links: Home, Upload, Recent, Guestbook, and a Login button. Below this is a search bar with a 'Search' button. The main content area is titled 'Login' and contains a form with two input fields: 'Username' and 'Password'. The 'Username' field contains the text 'wanda'#. Below the fields are two buttons: 'login' and 'Register'. At the bottom of the page, there is a footer with links: Home | Admin | Contact | Terms of Service.

Zdjęcie 7

Następnie możliwe jest logowanie bez hasła, jeśli znany jest login:



Zdjęcie 8

Jak się chronić?

Istnieje kilka sposobów ochrony przed odbitym wstrzyknięciem SQL:

- 1) Upewnić się, że wszelkie dane wejściowe użytkownika są sprawdzane i czyszczone przed ich przetworzeniem przez aplikację. Może to pomóc w zapobieganiu wstrzykiwaniu złośliwego kodu do aplikacji.
- 2) Użyć zapytań sparametryzowanych lub przygotowanych instrukcji. Może to pomóc w zapobieganiu SQL Injection przez oddzielenie danych wprowadzanych przez użytkownika od kodu SQL.
- 3) Używać bibliotek ORM, które mogą automatycznie wykrywać i zapobiegać atakom typu SQL Injection.
- 4) Podczas tworzenia kont użytkowników bazy danych i zarządzania nimi należy stosować zasadę najmniejszych uprawnień.
- 5) Regularnie testować i monitorować witryny pod kątem luk w zabezpieczeniach.
- 6) Należy zauważyć, że żadna pojedyncza metoda nie jest niezawodna, dlatego najlepiej jest stosować kombinację tych technik, aby zapewnić najlepszą ochronę przed atakami typu SQL Injection. Deweloperzy powinni również upewnić się, że ich kod jest regularnie sprawdzany pod kątem luk w zabezpieczeniach oraz że system jest aktualizowany o najnowsze poprawki bezpieczeństwa.

5. Parameter Manipulation

Stopień niebezpieczeństwa

High

Definicja podatności

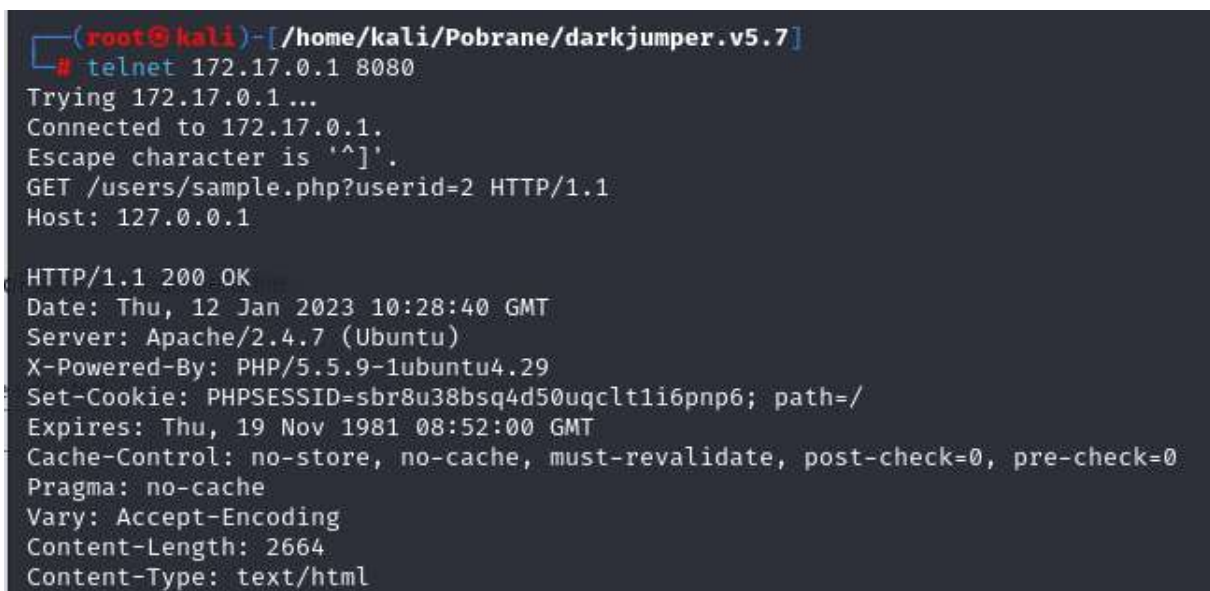
Parameter Manipulation (manipulacja parametrami) to rodzaj ataku, w którym atakujący manipuluje wartościami parametrów przesyłanych do aplikacji lub serwera. Celem tego ataku jest zmiana działania aplikacji lub przejęcie kontroli nad nią. Atakujący może manipulować parametrami przez różne metody, takie jak edycja żądania HTTP, modyfikacja adresu URL lub danych przesyłanych przez formularz. Może to prowadzić do różnych problemów bezpieczeństwa, takich jak wyciek danych, przejęcie konta użytkownika, czy nieautoryzowany dostęp do danych.

Opis działania podatności na stronie

Prosty test polega na połączeniu z aplikacją poprzez "fuzzing" losowych wartości, aby analizować kod zwrotny HTTP.

Proof of concept

Poniższy przykład pokazuje 200 OK, co oznacza, że identyfikator użytkownika #2 istnieje:



```
(root@kali)-[/home/kali/Pobrane/darkjumper.v5.7]
# telnet 172.17.0.1 8080
Trying 172.17.0.1...
Connected to 172.17.0.1.
Escape character is '^]'.
GET /users/sample.php?userid=2 HTTP/1.1
Host: 127.0.0.1

HTTP/1.1 200 OK
Date: Thu, 12 Jan 2023 10:28:40 GMT
Server: Apache/2.4.7 (Ubuntu)
X-Powered-By: PHP/5.5.9-1ubuntu4.29
Set-Cookie: PHPSESSID=sbr8u38bsq4d50uqclt1i6pnp6; path=/
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Pragma: no-cache
Vary: Accept-Encoding
Content-Length: 2664
Content-Type: text/html
```

Zdjęcie 9

Analogicznie użytkownik o ID 34 nie istnieje (kod 404)

```
(root@kali)-[/home/kali/Pobrane/darkjumper.v5.7]
# telnet 172.17.0.1 8080
Trying 172.17.0.1...
Connected to 172.17.0.1.
Escape character is '^]'.
GET /users/sample.php?userid=34 HTTP/1.1
Host: 127.0.0.1

HTTP/1.1 404 Not Found
Date: Thu, 12 Jan 2023 10:32:33 GMT
Server: Apache/2.4.7 (Ubuntu)
X-Powered-By: PHP/5.5.9-1ubuntu4.29
Set-Cookie: PHPSESSID=gfm2cr57bsdnh3pdudf09bf3es5; path=/
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Pragma: no-cache
Content-Length: 0
Content-Type: text/html
```

Zdjęcie 10

Jak się chronić?

Aby chronić się przed atakami manipulacji parametrów, należy:

- 1) Stosować skuteczne metody walidacji danych wejściowych, aby upewnić się, że przesłane wartości parametrów są prawidłowe i odpowiednie.
- 2) Kontrolować dostęp do danych, upewniając się, że tylko odpowiedni personel ma dostęp do ważnych informacji.
- 3) Regularnie monitorować systemy i aplikacje w celu wykrycia anomalii.
- 4) Stosować mechanizmy uwierzytelniania wieloskładnikowego, takie jak hasła i tokeny, aby zwiększyć bezpieczeństwo.
- 5) Regularnie aktualizuj swoje systemy i aplikacje, aby skorzystać z najnowszych zabezpieczeń.
- 6) Stosowanie mechanizmów logowania i rejestrowanie dostępu do systemu, w celu wykrycia nieautoryzowanej zmiany parametrów.
- 7) W przypadku wykrycia ataku manipulacji parametrami, należy natychmiast unieważnić aktualne sesje i przydzielić nowe identyfikatory sesji.

6. Reflected XSS Behind JavaScript

Stopień niebezpieczeństwa

High

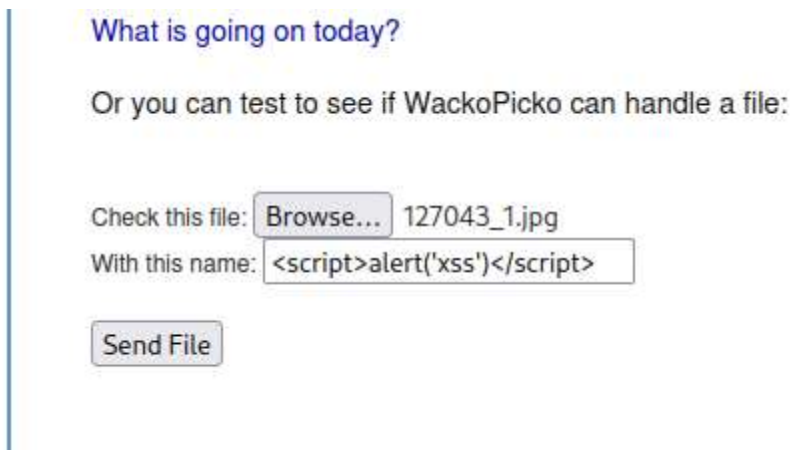
Definicja podatności

Reflected XSS Behind JavaScript oznacza specyficzny rodzaj ataku cross-site scripting (XSS), który pojawia się, gdy aplikacja internetowa zawiera lukę w formularzu opartym na JavaScript. W tym scenariuszu, atakujący wstrzykuje złośliwy kod do pola formularza, a następnie serwer odbija kod z powrotem do przeglądarki użytkownika, umieszczając go w kodzie JavaScript. Przeglądarka następnie wykonuje kod, co pozwala atakującemu na kradzież wrażliwych informacji lub uruchomienie dowolnego kodu na maszynie użytkownika.

Opis działania podatności na stronie

Istnieje luka reflected XSS na stronie głównej. Można sprawdzić czy WackoPicko jest w stanie pobrać obrazek, natomiast w nazwie pliku wpisujemy skrypt, który wykona się po kliknięciu Send File.

Proof of concept



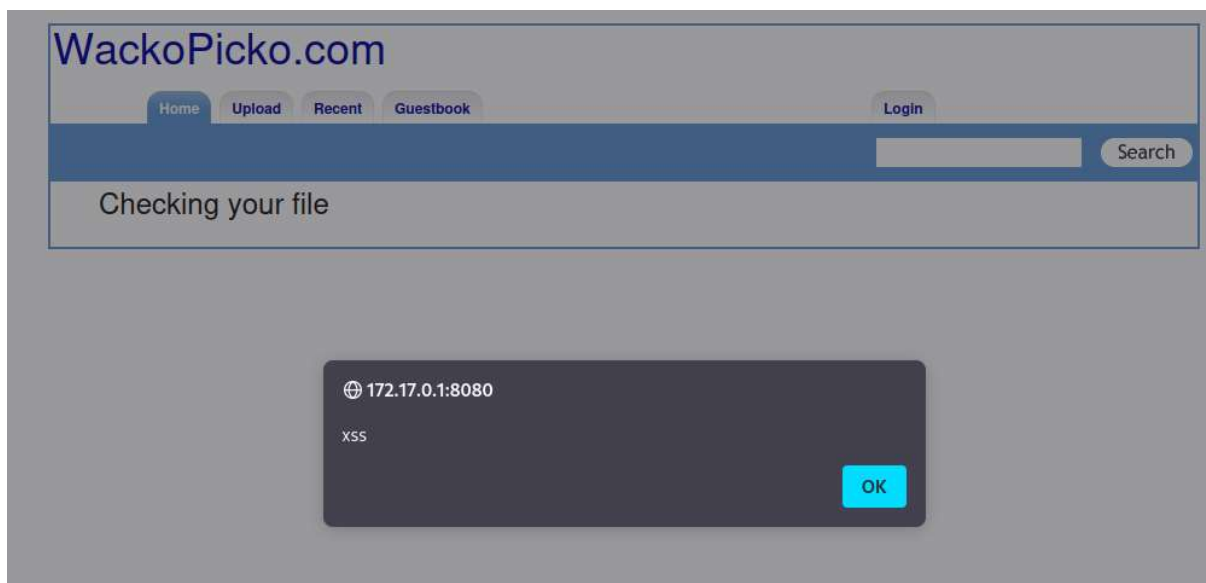
What is going on today?

Or you can test to see if WackoPicko can handle a file:

Check this file: 127043_1.jpg

With this name:

Zdjęcie 11



Zdjęcie 12

Jak widać uruchomił się skrypt, w tym przypadku był to zwykły tekst, lecz można wpisać dowolny (niebezpieczny skrypt).

Jak się chronić?

Aby chronić się przed atakami typu Reflected-XSS-Behind-JavaScript, należy:

- 1) Stosować skuteczne walidacje danych wejściowych, takie jak filtrowanie i sanitizacja, aby usunąć lub neutralizować złośliwy kod.
- 2) Unikać bezpośredniego odzwierciedlania danych wejściowych bez ich uprzedniego przetworzenia.
- 3) Zastosować output encoding, kodowanie wyjściowe, które pozwala na kodowanie specjalnych znaków, takich jak < i >, aby zapobiec ich interpretacji przez przeglądarkę jako kod HTML.
- 4) Stosować mechanizmy kontroli dostępu, takich jak uwierzytelnianie i autoryzacja, które pozwolą na ograniczenie dostępu do wrażliwych danych i funkcji tylko dla upoważnionych osób.
- 5) Regularnie monitorować swoje systemy i aplikacje w celu wykrycia anomalii.
- 6) W przypadku wykrycia luki, natychmiast zamknąć lukę i wprowadzić odpowiednie zabezpieczenia.
- 7) Stosować zaktualizowane przeglądarki, które posiadają wbudowane zabezpieczenia przeciwko atakom XSS.

7. Słaby login/hasło

Stopień niebezpieczeństwa

High

Definicja podatności

Słaby login lub hasło to niestrzeżone hasło lub nazwa użytkownika, które są łatwe do zgadnięcia lub złamania przez cyberprzestępców. Słaby login lub hasło zawierają zwykle proste, krótkie lub słabe hasła, takie jak "admin", "password" lub "1234", lub nazw użytkownika, które są łatwe do zgadnięcia, takie jak imię i nazwisko lub adres e-mail.

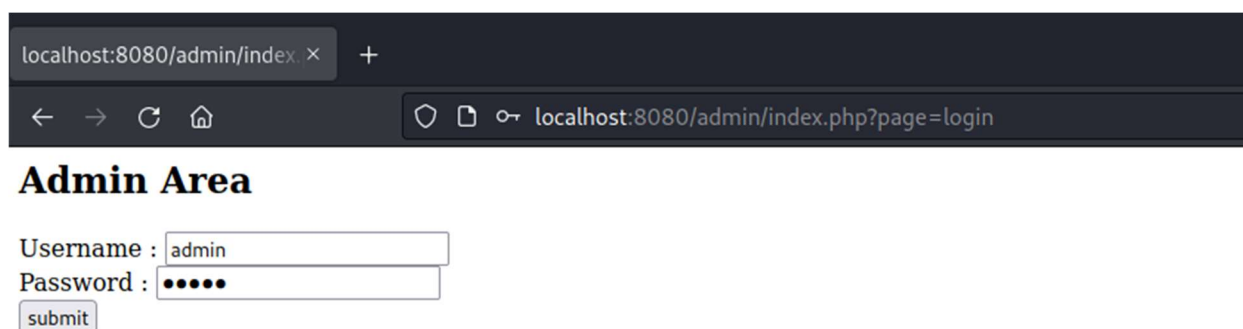
Słaby login lub hasło są często wykorzystywane przez cyberprzestępców do uzyskania dostępu do systemów lub danych. Mogą być również wykorzystywane do przeprowadzenia ataków phishingowych lub innych ataków socjotechnicznych.

Opis działania podatności na stronie

Strona konta administracyjnego zawiera łatwą do odgadnięcia nazwę użytkownika i hasło: admin/admin.

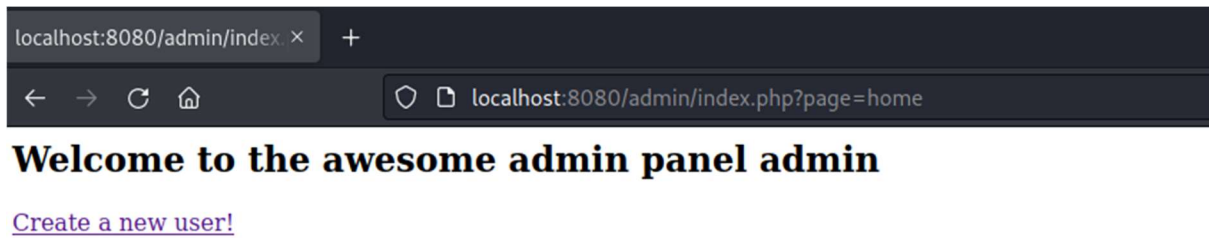
Proof of concept

W WackoPICKO kliknięto zakładkę „Admin” i uzupełniono dane logowania - w polu Username wpisano „admin”, w polu Password wpisano „admin”.



Zdjęcie 13

Udało się zalogować na panel admina:



Zdjęcie 14

Jak się chronić?

Aby chronić się przed atakami związanymi ze słabym loginem lub hasłem, ważne jest, aby:

- 1) Stosować silne hasła i nazwy użytkownika, które są trudne do zgadnięcia lub złamania
- 2) Stosować mechanizmy uwierzytelniania dwuskładnikowego i regularnie zmieniać hasła.

8. Stored SQL Injection

Stopień niebezpieczeństwa

High

Definicja podatności

Stored SQL injection to rodzaj ataku SQL injection, w którym osoba atakująca może wstrzyknąć złośliwy kod SQL do aplikacji internetowej, a następnie ten kod jest przechowywany w bazie danych. Osoba atakująca może później użyć tego kodu do wykonania złośliwych działań w bazie danych. Ten rodzaj ataku jest również znany jako „trwałe” wstrzyknięcie, ponieważ złośliwy kod jest przechowywany w bazie danych i będzie wykonywany za każdym razem, gdy ładowana jest strona, której dotyczy problem.

Opis działania podatności na stronie

Podczas tworzenia konta użytkownicy są proszeni o podanie swojego imienia. Ta podana wartość jest następnie używana bez oczyszczenia na stronie, która pokazuje innych użytkowników o podobnym imieniu. Atakujący może wykorzystać tę lukę, tworząc użytkownika o imieniu „'; DROP users;#” i odwiedzając stronę podobnych użytkowników.

Proof of concept

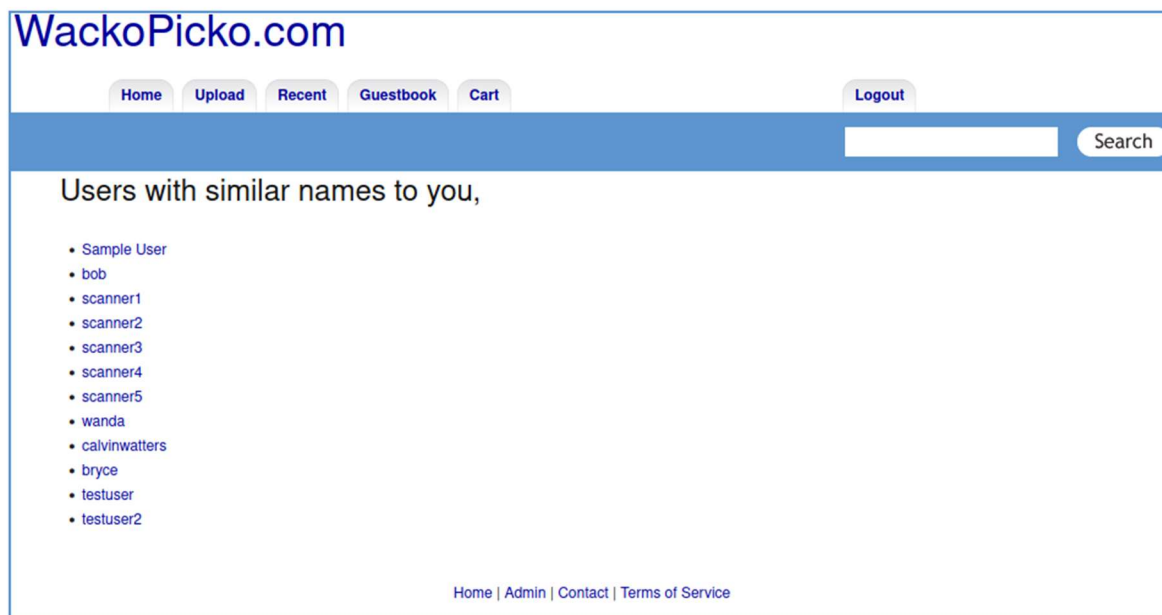
W WackoPICKO kliknięto zakładkę „Register” i wpisano „'; DROP users” w polu First Name.

The screenshot shows the WackoPICKO.com website's registration page. The header includes navigation links: Home, Upload, Recent, Guestbook, and a Login button. A search bar is also present. The main heading is "Register for an account!" with a subtext: "Protect yourself from hackers and check your password strength". A red error message states "All fields are required". The registration form contains the following fields and values:

Field	Value
Username :	testuser3
First Name :	';DROP users;#
Last Name :	user3
Password :	••••••••
Password again :	••••••••

A "Create Account!" button is located below the form. At the bottom of the page, there are links: Home | Admin | Contact | Terms of Service.

Następnie można było zauważyć wszystkich użytkowników odwiedzając zakładkę z użytkownikami o podobnym imieniu.



Zdjęcie 16

Jak się chronić?

Aby chronić się przed stored SQL injection, należy:

- 1) Uzyskiwać dane od użytkownika przez parametry wejściowe zamiast składni SQL.
- 2) Używać mechanizmów walidacji danych wejściowych, takich jak filtrowanie lub sprawdzanie typów danych.
- 3) Upewnić się, że konta bazy danych mają tylko niezbędne uprawnienia do wykonywania operacji.
- 4) Stosować najnowsze zabezpieczenia i aktualizacje dla bazy danych i systemu operacyjnego.

9. Multi-Step Stored XSS

Stopień niebezpieczeństwa

High

Definicja podatności

Multi-Step Stored XSS (Cross-Site Scripting) to rodzaj ataku, który polega na umieszczeniu złośliwego kodu JavaScript w danych przechowywanych na serwerze, takich jak bazy danych lub pliki. Kiedy inny użytkownik odwiedzi stronę zawierającą te dane, złośliwy kod zostanie wykonany na jego komputerze, co pozwala atakującemu na przejęcie kontroli nad przeglądarką ofiary lub uzyskanie dostępu do poufnych danych.

Multi-Step Stored XSS to polega na wykorzystaniu kilku etapów, aby przeprowadzić atak. Na przykład, atakujący może najpierw umieścić złośliwy kod w komentarzu na stronie internetowej, a następnie przekierować innego użytkownika do tego komentarza.

Opis działania podatności na stronie


Podobnie jak w przypadku ataku XSS przechowywanego w zakładce „Guestbook”, komentarze do zdjęć są podatne na atak Stored XSS. Luka ta jest jednak trudniejsza do wykorzystania, ponieważ użytkownik musi być zalogowany i musi potwierdzić podgląd komentarza przed faktycznym rozpoczęciem ataku.

Proof of concept

Treść komentarzy powiązanych ze zdjęciami nie jest odpowiednio oczyszczona. Umożliwia umieszczanie trwałych XSS w bazie danych:

Search

A Preview of what your comment on This grows outside my house will look like



Your Comment

<script>alert('xss');</script>

- by wanda

Cancel


Create

[Home](#) | [Admin](#) | [Contact](#) | [Terms of Service](#)

Zdjęcie 17

Search

A Preview of what your comment on This grows outside my house will look like



Your Comment

```
<script>alert('xss');</script>
```

- by wanda

[Home](#) | [Admin](#) | [Contact](#) | [Terms of Service](#)

Zdjęcie 18

Jak się chronić?

Aby chronić się przed Multi-Step Stored XSS, ważne jest, aby:

- 1) Stosować filtrowanie danych wejściowych
- 2) Sprawdzać typy danych
- 3) Sanitizować, kodować i dekodować dane
- 4) Stosować najnowsze zabezpieczenia i aktualizacje.

10. Forceful Browsing

Stopień niebezpieczeństwa

High

Definicja podatności

Forceful browsing to rodzaj ataku, w którym atakujący próbuje przeglądać i uzyskać dostęp do stron lub plików, które normalnie nie są dostępne dla użytkowników końcowych. Może to być wykonywane przez ręczne wprowadzanie adresów URL, automatyzację lub próby przejścia przez ochronę logowania.

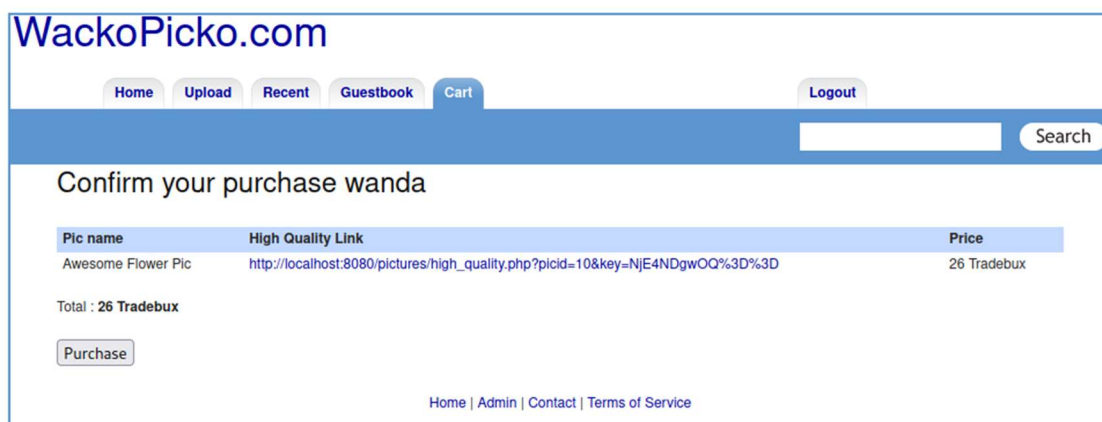
Atakujący może używać Forceful browsing, aby znaleźć i wykorzystać luki w zabezpieczeniach aplikacji, takie jak nieprawidłowe uprawnienia dostępu, niezabezpieczone pliki lub niedoskonałości w kodzie. Celem tego ataku jest uzyskanie dostępu do poufnych danych, dostępu do systemu lub wykonanie innych działań niepożądanych.

Opis działania podatności na stronie

Jedną z głównych idei stojących za WackoPicko jest możliwość zakupu przez użytkowników praw do wysokiej jakości wersji zdjęć. Jednak dostęp do linków do obrazu wysokiej jakości nie jest sprawdzany, a atakujący, który uzyska adres URL obrazu wysokiej jakości, może uzyskać do niego dostęp bez zakładania konta, omijając w ten sposób logikę uwierzytelniania.

Proof of concept

Po zakończeniu procesu kupowania obrazu w WackoPicko, wyświetlany jest pełny adres URL, aby zobaczyć obraz w pełnym rozmiarze:



Jak się chronić?

Aby chronić się przed Forceful Browsing, należy:

- 1) Implementować odpowiednie mechanizmy kontroli dostępu, takie jak autoryzacja i uwierzytelnianie, które pozwolą na dostęp tylko upoważnionym użytkownikom.
- 2) Uaktualniać oprogramowanie i systemy, aby zapewnić najnowsze zabezpieczenia.
- 3) Regularnie monitorować logi, aby wykryć nieprawidłowe lub podejrzaną działalność.
- 4) Edukować użytkowników końcowych, aby uniknąć niechcianych działań i wykluczyć przypadkowe działanie.
- 5) Przetwarzać dane wejściowe przed przekazaniem do serwera.
- 6) Stosować mechanizmy kontroli dostępu oparte na rolach i uprawnieniach.
- 7) Stosować mechanizmy blokowania dostępu po wykryciu nieprawidłowego działania.

11. Logic Flaw

Stopień niebezpieczeństwa

High

Definicja podatności

Logic flaw (luka logiczna) to rodzaj luki bezpieczeństwa, który polega na braku poprawnego działania logiki aplikacji. Może ona pojawić się w kodzie aplikacji, bibliotekach lub systemie operacyjnym. Atakujący wykorzystuje tę lukę, aby przejąć kontrolę nad aplikacją, uzyskać dostęp do poufnych danych lub wykonać inne nieautoryzowane działanie. Logic flaw może pojawić się z powodu błędu w projektowaniu lub implementacji systemu, lub z powodu braku odpowiedniej walidacji danych wejściowych.

Opis działania podatności na stronie

System kuponów cierpi na lukę logiczną, ponieważ kupon może być stosowany wielokrotnie do tego samego zamówienia, co zmniejsza końcową cenę zamówienia do zera. Kupon w danych początkowych to SUPERYOU21.


Proof of concept

WackoPicko.com

Home Upload Recent Guestbook Cart Logout

Search

Welcome to your cart scanner1

Pic name	Sample Pic	Price	Delete?
Our House		30 Tradebux	<input type="checkbox"/>

Coupon Code	Coupon Amount
SUPERYOU21	10% Off
SUPERYOU21	10% Off
SUPERYOU21	10% Off
SUPERYOU21	10% Off
SUPERYOU21	10% Off
SUPERYOU21	10% Off
SUPERYOU21	10% Off

Remove From Cart

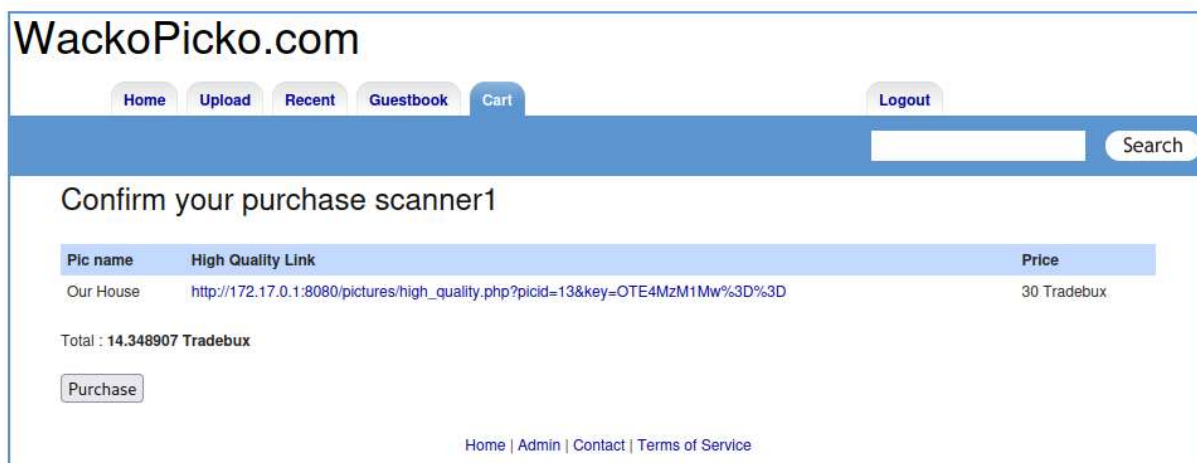
Enter Coupon Code:

Submit Coupon

Continue to Confirmation

Home | Admin | Contact | Terms of Service

Zdjęcie 20



Zdjęcie 21

Jak się chronić?

Aby chronić się przed logicznymi lukami bezpieczeństwa, należy:

- 1) Stosować dobrych praktyk projektowych, takich jak modularność i separacja odpowiedzialności, aby zwiększyć przewidywalność i łatwość testowania aplikacji.
- 2) Regularnie testować aplikację, w tym również testy jednostkowe, aby wykryć potencjalne problemy logiczne.
- 3) Regularnie aktualizować oprogramowanie i systemy, aby skorzystać z najnowszych poprawek bezpieczeństwa.
- 4) Dokumentować i udostępniać informacje na temat luk bezpieczeństwa, aby umożliwić innym dostosowanie swoich systemów.
- 5) Stosować mechanizmy kontroli dostępu, takie jak uwierzytelnianie i autoryzacja, które pozwolą na ograniczenie dostępu do wrażliwych danych i funkcji tylko dla upoważnionych osób.
- 6) Regularnie monitorować swoje systemy i aplikacje w celu wykrycia anomalii.
- 7) W przypadku wykrycia luki logicznej, należy natychmiast zamknąć lukę i wprowadzić odpowiednie zabezpieczenia.

Podsumowanie

Audyt bezpieczeństwa strony internetowej wykazał kilka istotnych podatności, które mogą stanowić zagrożenie dla bezpieczeństwa danych i działania strony. Są to:

- 1) Reflected XSS, polegająca na wykorzystaniu skryptów wprowadzanych przez użytkownika, które są następnie wykonywane przez przeglądarkę
- 2) Stored XSS, polegająca na przechowywaniu skryptów atakujących na serwerze, które są następnie wykonywane przez przeglądarkę
- 3) Podatność na ID sesji, polegająca na braku odpowiedniego zabezpieczenia identyfikatorów sesji
- 4) Stored SQL Injection i Reflected SQL Injection, polegające na wykorzystaniu podatności w bazie danych do wykonania nieautoryzowanego dostępu do danych
- 5) Multi-Step Stored XSS, polegająca na ataku przeprowadzonym w kilku etapach
- 6) Forceful Browsing, polegająca na próbie dostępu do zabezpieczonych zasobów przez przeglądarkę
- 7) Parameter Manipulation, polegająca na wykorzystaniu nieprawidłowych lub nieautoryzowanych parametrów
- 8) Reflected XSS Behind JavaScript, polegająca na ataku przeprowadzonym przez skrypty javascript
- 9) Logic Flaw, polegająca na błędach logicznych w aplikacji
- 10) Słaby login/hasło, polegająca na słabych nazwach użytkownika/hasłach.

Audyt bezpieczeństwa jest ważny, ponieważ pozwala na identyfikację potencjalnych zagrożeń dla bezpieczeństwa systemu i danych. Dzięki temu można podjąć odpowiednie działania mające na celu zabezpieczenie systemu przed atakami i incydentami bezpieczeństwa. Audyt bezpieczeństwa pozwala także na określenie, czy istniejące zabezpieczenia są odpowiednie i skuteczne, co pozwala na ich ewentualną poprawę i uaktualnienie.

Audyt bezpieczeństwa jest ważny również dla organizacji, które chcą spełnić wymagania prawne dotyczące ochrony danych osobowych (np. RODO) oraz standardów bezpieczeństwa (np. ISO 27001). Dzięki regularnym audytom bezpieczeństwa, organizacje mogą zwiększyć swoją odporność na cyberataki i zabezpieczyć swoje dane przed nieautoryzowanym dostępem.