9 Cross-Site-Scripting (XSS)

Teoria

Atak Cross-Site Scripting (XSS) jest jednym z najczęstszych rodzajów ataków w Internecie, który polega na wstrzykiwaniu złośliwego kodu JavaScript do stron internetowych odwiedzanych przez innych użytkowników. Dzięki XSS, atakujący może przejąć kontrolę nad sesjami użytkowników, kraść dane, a także wprowadzać inne formy złośliwego działania.

Reflected XSS

Reflected XSS (zwany także *Non-Persistent XSS*) jest rodzajem ataku, który zachodzi, gdy złośliwy kod JavaScript jest wstrzykiwany w parametrach URL lub w żądaniach HTTP (np. w formularzach). Kod ten zostaje natychmiast przetworzony i odesłany z powrotem do przeglądarki użytkownika w odpowiedzi serwera, a przeglądarka wykonuje go, myśląc, że jest to bezpieczny skrypt. Załóżmy, że mamy aplikację, która przyjmuje dane w parametrze URL, jak np.

$$http://example.com/search?q =< script > alert('XSS') < /script >$$

Jeśli serwer odpowiednio nie sanitizuje danych wejściowych i odsyła je bezpośrednio do przeglądarki, wówczas złośliwy skrypt w parametrze `q` zostanie wykonany w kontekście strony.

Aby zabezpieczyć aplikację przed Reflected XSS, należy:

- **Walidacja i sanitizacja danych wejściowych:** Wszystkie dane pochodzące od użytkownika, w tym parametry URL i dane formularzy, powinny być starannie walidowane i oczyszczane z potencjalnie niebezpiecznych znaków (np. `<`, `>`, `&`, `"`, `'`).
- **Używanie nagłówków Content Security Policy (CSP):** CSP pozwala określić, które źródła mogą ładować skrypty, ograniczając tym samym możliwość wstrzykiwania złośliwego kodu.
- **Escape danych przed wyświetleniem:** Przed wyświetleniem danych na stronie, należy je *escapować* (zamienić specjalne znaki na ich odpowiedniki w HTML).

Stored XSS

Stored XSS (zwany także *Persistent XSS*) zachodzi, gdy złośliwy kod JavaScript jest wstrzykiwany do aplikacji i zapisywany na serwerze, np. w bazie danych, pliku logu lub innym trwałym magazynie danych. Kiedy inny użytkownik odwiedza stronę, która wyświetla te dane, złośliwy skrypt jest automatycznie wykonany przez jego przeglądarkę. Załóżmy, że użytkownik wpisuje w formularzu komentarz, który zawiera złośliwy skrypt:

$$< script > alert('XSS') < /script >$$

Jeśli aplikacja zapisuje ten komentarz w bazie danych bez odpowiedniego filtrowania, a następnie wyświetla go innym użytkownikom, to skrypt zostanie wykonany na komputerze odwiedzającego stronę użytkownika.

Aby zabezpieczyć aplikację przed Stored XSS, należy:

- **Sanitizacja danych przy zapisie:** Wszystkie dane wprowadzane przez użytkowników powinny być sanitizowane przed zapisaniem ich w bazie danych, eliminując wszelkie tagi HTML oraz skrypty JavaScript.
- **Escape danych przy wyświetlaniu:** Zanim dane zostaną wyświetlone w HTML, należy je odpowiednio *escape'ować*, aby zapobiec wykonaniu wstrzykniętych skryptów.
- **Używanie odpowiednich nagłówków HTTP:** Nagłówki takie jak *X-XSS-Protection* mogą pomóc w ochronie przed niektórymi formami XSS, choć nie zastępują one pełnej walidacji i sanitizacji danych.

DOM-based XSS

DOM-based XSS występuje, gdy złośliwy skrypt jest wstrzykiwany do aplikacji przez manipulację Document Object Model (DOM) w przeglądarkce, bez konieczności interakcji z serwerem. Złośliwy kod jest uruchamiany w momencie, gdy aplikacja webowa przetwarza dane wejściowe użytkownika bez odpowiedniego oczyszczania, a te dane trafiają do manipulacji DOM.

W przypadku DOM-based XSS, atak może wyglądać tak:

$$http://example.com/\#q =< script > alert('XSS') < /script >$$

Aplikacja może następnie wziąć parametr `q` z URL i użyć go do manipulacji DOM, np. wstawiając go jako część treści strony, co skutkuje wykonaniem skryptu w przeglądarce.

Aby zabezpieczyć aplikację przed DOM-based XSS, należy:

- **Walidacja danych wejściowych:** Ważne jest, aby wszystkie dane wejściowe, które mogą być użyte w manipulacji DOM, były odpowiednio walidowane i oczyszczane.
- **Używanie bezpiecznych metod manipulacji DOM:** Zamiast bezpośredniego manipulowania HTML za pomocą takich metod jak `innerHTML`, lepiej używać metod takich jak `textContent` lub `createElement`, które nie interpretują danych jako HTML.
- **CSP i nagłówki bezpieczeństwa:** Używanie polityki Content Security Policy oraz innych nagłówków, takich jak `X-XSS-Protection`, może pomóc w ochronie przed DOM-based XSS.

Ataki XSS stanowią poważne zagrożenie dla bezpieczeństwa aplikacji webowych. Dzielą się one na trzy główne typy: Reflected XSS, Stored XSS oraz DOM-based XSS, z których każdy wymaga innych metod zabezpieczeń. Aby skutecznie chronić aplikacje przed tymi atakami, należy stosować odpowiednią walidację i sanitizację danych wejściowych, korzystać z polityk bezpieczeństwa (takich jak CSP) oraz używać bezpiecznych metod manipulacji DOM. Regularne audyty bezpieczeństwa i testy penetracyjne są również kluczowe w zapewnianiu ochrony przed XSS.

Linki

- https://sekurak.pl/czym-jest-xss/
- https://portswigger.net/burp/documentation/desktop/testing-workflow/input-validation/xss
- https://portswigger.net/web-security/cross-site-scripting/cheat-sheet
- https://portswigger.net/support/using-burp-to-find-cross-site-scripting-issues
- https://medium.com/@kaorrosi/xss-discovery-and-exploitation-with-burpsuite-91d98865c1ee
- https://owasp.org/www-community/attacks/xss/
- https://book.hacktricks.xyz/pentesting-web/xss-cross-site-scripting
- https://cheatsheetseries.owasp.org/cheatsheets/XSS_Filter_Evasion_Cheat_Sheet.html
- https://github.com/payloadbox/xss-payload-list
- https://github.com/swisskyrepo/PayloadsAllTheThings/tree/master/XSS%20Injection
- https://github.com/allanlw/svg-cheatsheet
- https://book.hacktricks.xyz/pentesting-web/xss-cross-site-scripting/server-side-xss-dynamic-pdf
- https://developer.mozilla.org/en-US/docs/Web/API/Window/localStorage
- https://www.geeksforgeeks.org/local-storage-vs-cookies/
- https://blog.logrocket.com/localstorage-javascript-complete-guide/
- https://codewithpawan.medium.com/enhancing-security-and-efficiency

Zadania

9.0 *Inside a Docker container, there's a simple Flask webserver written in Python, which accepts user input and prints it.*

(a) Korzystając z przygotowanego obrazu Dockerowego `mazurkatarzyna/xss-example-server-1:latest`, spróbuj wykonać atak XSS. Spróbuj wyświetlić zawartość nagłówka `Cookie`. Sprawdź kod źródłowy serwera. W jaki sposób można zmodyfikować kod, aby przeprowadzenie ataku było niemożliwe?

(b) Korzystając z przygotowanego obrazu Dockerowego `mazurkatarzyna/xss-example-server-2:latest`, spróbuj wykonać atak XSS. Spróbuj wyświetlić zawartość nagłówka `Cookie`. Sprawdź kod źródłowy serwera. Czy atak XSS się wykonał?

9.1 *LimeSurvey is a powerful, open-source survey platform. A free alternative to SurveyMonkey, Typeform, Qualtrics, and Google Forms, making it simple to create online surveys and forms with unmatched flexibility. In version `6.2.11` is vulnerable to CWE-79: Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting', Reflected XSS).*

(a) Zapoznaj się z opisem podatności **CWE-79: XSS**.

(b) Korzystając z przygotowanego pliku Dockerowego (`docker-compose-cwe-79-ex1.yml`), uruchom aplikację `LimeSurvey` w podatnej wersji `6.2.11`.

(c) Pokaż, że podatność istnieje. Podpowiedź: podatność można zaobserwować podczas dodawania nowego menu.

(d) Po udanym potwierdzeniu istnienia podatności, oszacuj poziom jej krytyczności używając kalkulatora CVSS.

9.2 *LimeSurvey is a powerful, open-source survey platform. A free alternative to SurveyMonkey, Typeform, Qualtrics, and Google Forms, making it simple to create online surveys and forms with unmatched flexibility. In version `5.6.4` is vulnerable to CWE-79: Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting', Reflected XSS).*

(a) Zapoznaj się z opisem podatności **CWE-79: XSS**.

(b) Korzystając z przygotowanego pliku Dockerowego (`docker-compose-cwe-79-ex2.yml`), uruchom aplikację `LimeSurvey` w podatnej wersji `5.6.4`.

(c) Pokaż, że podatność istnieje. Podpowiedź: podatność można zaobserwować podczas dodawania nowej ankiety.

(d) Po udanym potwierdzeniu istnienia podatności, oszacuj poziom jej krytyczności używając kalkulatora CVSS.

9.3 *LimeSurvey is a powerful, open-source survey platform. A free alternative to SurveyMonkey, Typeform, Qualtrics, and Google Forms, making it simple to create online surveys and forms with unmatched flexibility. In version `6.2.2` is vulnerable to CWE-79: Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting', Stored XSS).*

(a) Zapoznaj się z opisem podatności **CWE-79: XSS**.

(b) Korzystając z przygotowanego pliku Dockerowego (`docker-compose-cwe-79-ex3.yml`), uruchom aplikację `LimeSurvey` w podatnej wersji `6.2.2`.

(c) Pokaż, że podatność istnieje. Podpowiedź: podatność można zaobserwować importu / eksportu użytkowników.

(d) Po udanym potwierdzeniu istnienia podatności, oszacuj poziom jej krytyczności używając kalkulatora CVSS.

9.4 *Microweber as a Laravel CMS, Microweber provides an open-source, drag-and-drop, and PHP-powered website creation experience. Microweber's simple, yet powerful admin interface puts you in full control over the content in every single area of your website. Microweber is free and open source website builder and CMS, so you can freely download it and use it to build website. Prior to version 1.2.11 Microweber is vulnerable to CWE-79: Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting'), Stored XSS), CVE-2022-0963.*

(a) Zapoznaj się z opisem podatności **CWE-79: XSS**, **CVE-2022-0963**.

(b) Korzystając z przygotowanego obrazu Dockerowego (`mazurkatarzyna/cwe-79-ex4:latest`), uruchom aplikację `Microweber` w podatnej wersji `1.2.11`.

(c) Wiedząc, że w module `Files` występuje podatność XSS, wrzuć na serwer dowolny plik z podatnym kodem JavaScript, który wyświetli dowolną wiadomość.

(d) Wiedząc, że w module `Files` występuje podatność XSS, wrzuć na serwer dowolny plik z podatnym kodem JavaScript, który wyświetli ciasteczka (`cookies`) zalogowanego użytkownika.

(e) Zmodifukuj wrzucany plik, tak aby zamiast wyświetlać ciasteczko użytkownika, wysyłał je na zewnętrzny serwer.

(f) Posiadając przechwycone ciasteczko administratora, dodaj do aplikacji nowego użytkownika z uprawnieniami administratora.

(g) Zaloguj się na konto nowego użytkownika i sprawdź, czy ma uprawnienia administratora.

(h) Po udanym potwierdzeniu istnienia podatności, oszacuj poziom jej krytyczności używając kalkulatora CVSS.

9.5 *Memos is an open-source, self-hosted note-taking solution designed for seamless deployment and multi-platform access. Experience effortless plain text writing with pain-free, complemented by robust Markdown syntax support for enhanced formatting. In version 0.8.3 Memos is vulnerable to CWE-79: Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting'), Stored XSS), CVE-2022-4690.*

(a) Zapoznaj się z opisem podatności **CWE-79: XSS**, **CVE-2022-4690**.

(b) Korzystając z przygotowanego obrazu Dockerowego (`mazurkatarzyna/cwe-79-ex5:latest`), uruchom aplikację `Memos` w podatnej wersji `0.8.3`.

(c) Wgraj na serwer obrazek, którego otwarcie spowoduje wykonanie kodu JavaScript i wyświetlenie dowolnej wiadomości.

(d) Wgraj na serwer obrazek, którego otwarcie spowoduje wykonanie kodu JavaScript i wyświetlenie ciasteczek zalogowanego użytkownika.

(e) Zmodifukuj wrzucany plik, tak aby zamiast wyświetlać ciasteczko użytkownika, wysyłał je na zewnętrzny serwer.

(f) Wykorzystując przechwycone ciasteczko administratora, zmień jego hasło (administrator nie będzie się w stanie zalogować).

(g) Spróbuj zalogować się do aplikacji jako administrator.

(h) Po udanym potwierdzeniu istnienia podatności, oszacuj poziom jej krytyczności używając kalkulatora CVSS.

9.6 *`FlatPress` is a lightweight, easy-to-set-up blogging engine. Plain and simple, just PHP. No database needed! In version `1.2.1`, `FlatPress` is vulnerable to CWE-79: Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting', Stored XSS), CVE-2022-4605.*

(a) Zapoznaj się z opisem podatności **CWE-79: XSS**, **CVE-2022-4605**.

(b) Korzystając z przygotowanego obrazu Dockerowego (`mazurkatarzyna/cve-2022-4605:latest`), uruchom aplikację `FlatPress` w podatnej wersji `1.2.1`.

(c) Wgraj na serwer plik, którego otwarcie spowoduje wykonanie kodu JavaScript i wyświetlenie ciasteczek zalogowanego użytkownika.

(d) Po udanym potwierdzeniu istnienia podatności, oszacuj poziom jej krytyczności używając kalkulatora CVSS.

9.7 *`LibreNMS` is a fully featured network monitoring system that provides a wealth of features and device support. In version `23.8.2`, `LibreNMS` is vulnerable to CWE-79: Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting', DOM XSS), CVE-2023-5060.*

(a) Zapoznaj się z opisem podatności **CWE-79: XSS**, **CVE-2023-5060**.

(b) Korzystając z przygotowanego pliku Dockerowego (`docker-compose-cwe-79-ex7.yml`), uruchom aplikację `LibreNMS` w podatnej wersji `23.8.2`.

(c) Wrzuć na serwer payload XSS, który spowoduje wykonanie kodu JavaScript i wyświetlenie ciasteczek zalogowanego użytkownika.

(d) Po udanym potwierdzeniu istnienia podatności, oszacuj poziom jej krytyczności używając kalkulatora CVSS.

9.8 *`Answer`, in fact, Apache Answer, is a Q&A platform software for teams at any scale. Whether it's a community forum, help center, or knowledge management platform, you can always count on Answer. In version `1.0.2`, `Answer` is is vulnerable to CWE-79: Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting', DOM XSS), CVE-2023-0741.*

(a) Zapoznaj się z opisem podatności **CWE-79: XSS**, **CVE-2023-0741**.

(b) Korzystając z przygotowanego obrazu Dockerowego (`mazurkatarzyna/cve-2023-0741:latest`), uruchom aplikację `Answer` w podatnej wersji `1.0.2`.

(c) Podczas instalacji `Answer` wybierz bazę SQLite. Dodaj konto administratora. Z poziomu konta administratora dodaj do aplikacji zwykłego użytkownika o nazwie `student`.

(d) Zaloguj się w innej przeglądarce / karcie incognito jako użytkownik `student` do aplikacji `Answer`.

(e) Jako użytkownik `student` dodaj do aplikacji pytanie, które spowoduje wyświetlenie zawartości `local storage`. Co znajduje się w `local storage`? Z konta administratora podejrzyj pytanie użytkownika `student` - payload XSS powinien się wykonać.

(f) Jako użytkownik `student` dodaj do aplikacji kolejne pytanie, które spowoduje wysłanie zawartości `local storage` na zewnętrzny serwer. Co znajduje się w `local storage`?

(g) Posiadając zawartość `local storage` wyślij do serwera request, który zmieni e-mail administratora na e-mail usera `student` (Twój własny). Dzięki temu administrator nie będzie się mógł zalogować, a użytkownik `student` uzyska dostęp do konta admina.

(h) Po udanym potwierdzeniu istnienia podatności, oszacuj poziom jej krytyczności używając kalkulatora CVSS.

Cross-Site-Scripting (XSS) - Odpowiedzi

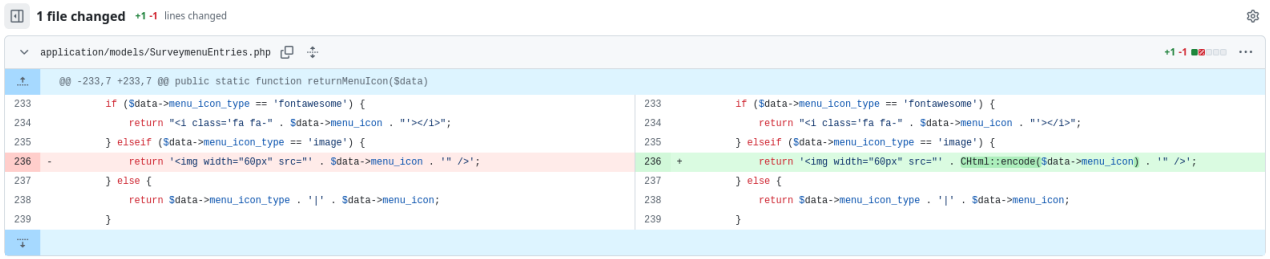
9.0

- Uruchom kontener za pomocą polecenia:
  `docker run -it -p 9901:9901 mazurkatarzyna/xss-example-server-1`.
  Kolejno należy uruchomić serwer: `python3 app.py`, serwer będzie dostępny pod adresem:
  http://127.0.0.1:9901.

- Uruchom kontener za pomocą polecenia:
  `docker run -it -p 9902:9902 mazurkatarzyna/xss-example-server-2`.
  Kolejno należy uruchomić serwer: `python3 app.py`, serwer będzie dostępny pod adresem:
  http://127.0.0.1:9902.

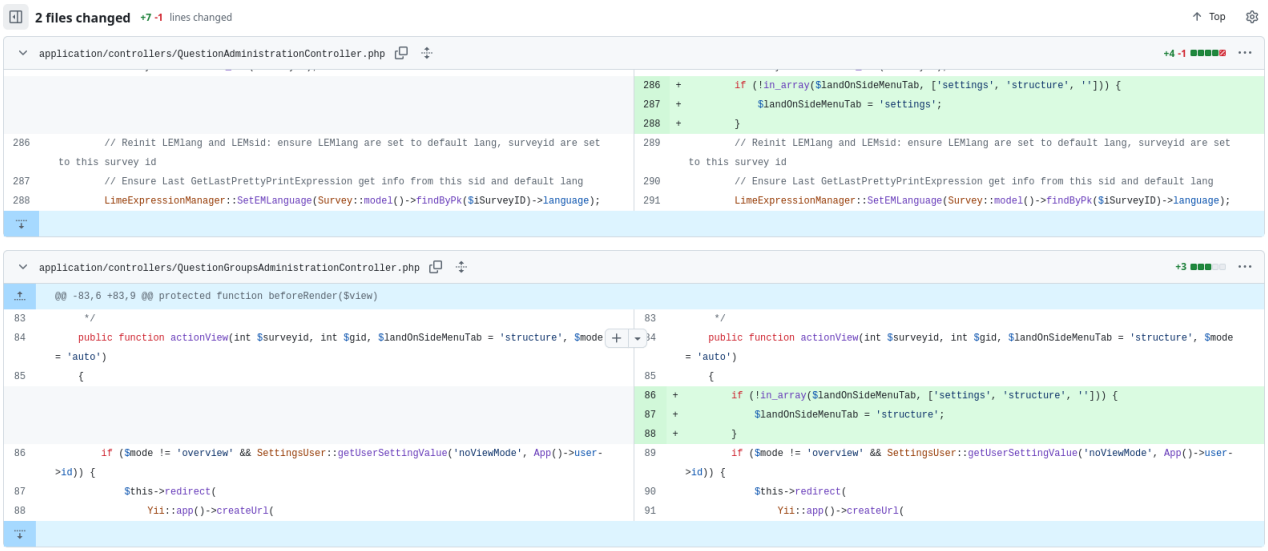
9.1

- Uruchom aplikację za pomocą polecenia: `docker compose -f docker-compose-cwe-79-ex1.yml up`. Serwer będzie dostępny pod adresem http://127.0.0.1:9091.
- Zwróć uwagę, jak twórcy naprawili podatność:



9.2

- Uruchom aplikację za pomocą polecenia: `docker compose -f docker-compose-cwe-79-ex2.yml up`. Serwer będzie dostępny pod adresem http://127.0.0.1:9092.
- Zwróć uwagę, jak twórcy naprawili podatność:

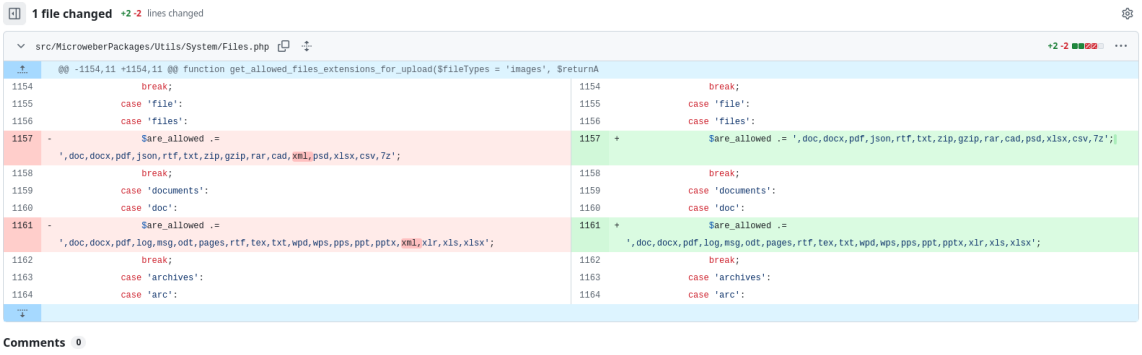


9.3

- Uruchom aplikację za pomocą polecenia: `docker compose -f docker-compose-cwe-79-ex3.yml up`. Serwer będzie dostępny pod adresem http://127.0.0.1:9093.
- Zwróć uwagę, jak twórcy naprawili podatność:

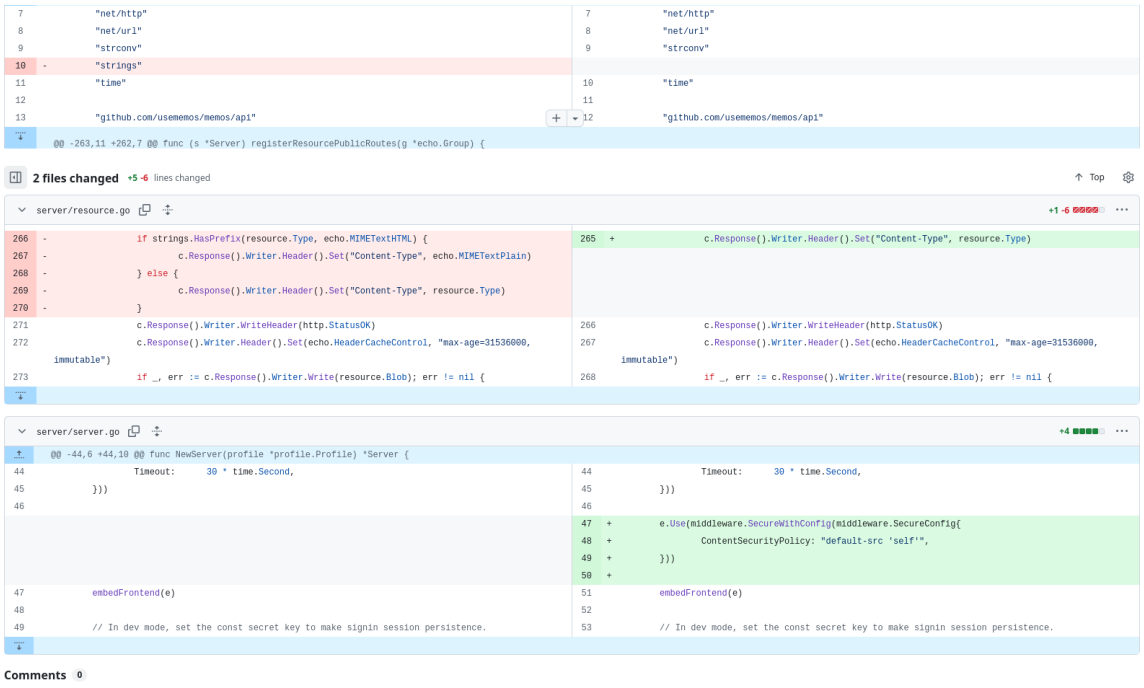
9.4

- Uruchom aplikację za pomocą polecenia: `docker run -dp 9094:80 mazurkatarzyna/cwe-79-ex4:latest`. Serwer będzie dostępny pod adresem http://127.0.0.1:9094.
- Sprawdź znane payloady XSS: https://github.com/swisskyrepo/PayloadsAllTheThings/tree/master/XSS%20Injection.
- Jako zewnętrzny serwer, na który wyślesz ciasteczko, możesz wykorzystać serwer napisany w języku Python: `python3 server.py`. Serwer będzie nasłuchiwał na porcie `9999`.
- Jaki request trzeba wysłać, żeby dodać nowego usera? Sprawdź API aplikacji. Możesz również w `BurpSuite` przechwycić request dodający użytkownika, i zmodyfikować odpowiednie pola. Do wysłania requestu możesz użyć narzędzia `curl` lub `BurpSuite`.
- Zwróć uwagę, jak twórcy naprawili podatność:



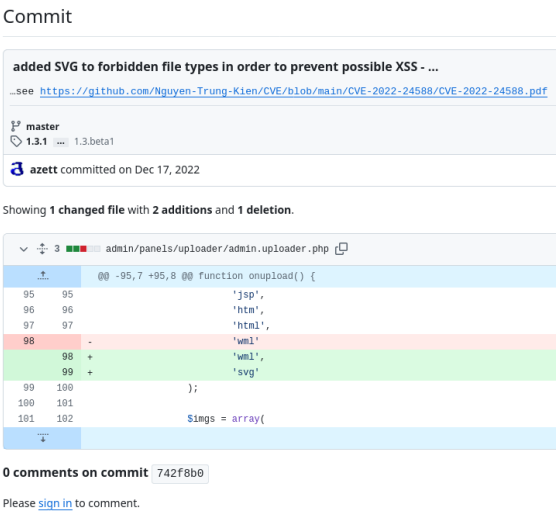
9.5

- Uruchom aplikację za pomocą polecenia: `docker run -dp 9095:5230 mazurkatarzyna/cwe-79-ex5:latest`. Serwer będzie dostępny pod adresem http://127.0.0.1:9095.
- Sprawdź payloady XSS w obrazkach: https://github.com/allanlw/svg-cheatsheet.
- Jako zewnętrzny serwer, na który wyślesz ciasteczko, możesz wykorzystać serwer napisany w języku Python: `python3 server.py`. Serwer będzie nasłuchiwał na porcie `9999`.
- Jaki request trzeba wysłać, żeby zmienić hasło usera? Sprawdź API aplikacji. Możesz również w `BurpSuite` przechwycić request zmieniający hasło użytkownika, i zmodyfikować odpowiednie pola. Do wysłania requestu możesz użyć narzędzia `curl` lub `BurpSuite`.
- Zwróć uwagę, jak twórcy naprawili podatność:



9.6

- Uruchom aplikację za pomocą polecenia: `docker run -dp 9096:80 mazurkatarzyna/cve-2022-4605:latest`. Serwer będzie dostępny pod adresem http://127.0.0.1:9096.
- Sprawdź format `SVG`. Zwróć uwagę, jak twórcy naprawili podatność:



9.7

- Uruchom aplikację za pomocą polecenia: `docker compose -f docker-compose-cwe-79-ex7.yml up`. Serwer będzie dostępny pod adresem http://127.0.0.1:9097.
- Dodaj do aplikacji użytkownika `admin`: `docker exec librenms create_admin` i zaloguj się na konto admina.
- Znajdź miejsce, do którego możesz wstrzyknąć payload XSS.
- Zwróć uwagę, jak twórcy naprawili podatność:

1 file changed +3 -3 lines changed

Top

includes/html/pages/search/ipv6.inc.php +3 -3

```
@@ -74,7 +74,7 @@
                "</select>"+                                                    "</select>"+
                "</div>"+                                                       "</div>"+
                "<div class=\"form-group\">"+                                   "<div class=\"form-group\">"+
-               "<input type=\"text\" name=\"address\" id=\"address\" size=40 value=\"<?php echo $_POST['address']; ?>\" class=\"form-control input-sm\" placeholder=\"IPv6 Address\"/>"+
+               "<input type=\"text\" name=\"address\" id=\"address\" size=40 value=\"<?php echo htmlspecialchars($_POST['address']); ?>\" class=\"form-control input-sm\" placeholder=\"IPv6 Address\"/>"+
                "</div>"+                                                       "</div>"+
                "<button type=\"submit\" class=\"btn btn-default input-sm\">Search</button>"+  "<button type=\"submit\" class=\"btn btn-default input-sm\">Search</button>"+
                "</form></span></div>"+                                         "</form></span></div>"+
@@ -86,8 +86,8 @@
            id: "address-search",                                               id: "address-search",
            search_type: "ipv6",                                                search_type: "ipv6",
            device_id: '<?php echo htmlspecialchars($_POST['device_id']); ?>',  device_id: '<?php echo htmlspecialchars($_POST['device_id']); ?>',
-           interface: '<?php echo $_POST['interface']; ?>',                    interface: '<?php echo htmlspecialchars($_POST['interface']); ?>',
-           address: '<?php echo $_POST['address']; ?>'                         address: '<?php echo htmlspecialchars($_POST['address']); ?>'
+           interface: '<?php echo htmlspecialchars($_POST['interface']); ?>',  interface: '<?php echo htmlspecialchars($_POST['interface']); ?>',
+           address: '<?php echo htmlspecialchars($_POST['address']); ?>'       address: '<?php echo htmlspecialchars($_POST['address']); ?>'
        };                                                                      };
    },                                                                      },
    url: "ajax_table.php",                                                  url: "ajax_table.php",
```

Comments 0