Пірог О. В., канд. техн. наук, доцент кафедри КІ та КБ
Державний університет «Житомирська політехніка»

РЕЦЕПТИ БЕЗПЕЧНОЇ РОЗРОБКИ ВЕБ-ДОДАТКІВ ВІД OWASP ЩОДО ЗАПОБІГАННЯ ВРАЗЛИВОСТІ МІЖСАЙТОВОГО СКРИПТИНГУ (XSS)

Причина використання моделі запобігання проблемам міжсайтового скриптингу від OWASP у тому, що це зменшує складність визначення того, де ми можемо розмістити ненадійні дані. Ідея полягає в тому, щоб розглядати HTML-сторінку як шаблон із слотом, де розробникам дозволено розміщувати ненадійні дані. Розміщення ненадійних даних в інших місцях HTML-сторінки не допускається в цій моделі запобігання загрозам. Це фактично білий список, де можна розміщувати ненадійні дані. Кожен із цих так званих слотів має свої правила безпеки, і це означає, що ненадійні дані не можуть бути інтерпретовані не так, як задумано розробниками.

Ключ вирішення проблеми полягає у використанні безпечних бібліотек кодування. Це допоможе переконатися, що не було зроблено помилок під час кодування даних. Безпечні бібліотеки кодування вже існують, вони вже були багато разів протестовані, тому існує впевненість, що вони є досить безпечними.

Правила запобігання міжсайтовому скриптингу від OWASP:

0) Не вставляти ненадійні дані безпосередньо в код додатку (крім визначених слотів). Ніколи не поміщати недовірені дані безпосередньо між тегами скрипта, у коментарі HTML, в ім'я атрибута, тега, CSS, як показано на прикладах:

```
< script > …NEVER PUT UNTRASTED DATA HERE … < /script >

<!–…NEVER PUT UNTRASTED DATA HERE…-->

<div …NEVER PUT UNTRASTED DATA HERE…>

<NEVER PUT UNTRASTED DATA HERE…>

< style > …NEVER PUT UNTRASTED DATA HERE … < /style >
```

1) Використовувати захищену бібліотеку кодування для екранування HTML перед вставкою ненадійних даних у вміст HTML.

Кодуються наступні символи (&, <, >, ", ') за допомогою кодування сутності HTML, щоб запобігти перемиканню в будь-який контекст виконання.

Ось приклад псевдокода для створення документів на боці серверу:

```
$escaped_data = lib.htmlEs c(input)

$str = "< body >" + $escaped_data + "< /body >"
```

2) Кодувати атрибути перед вставкою ненадійних даних у HTML. Використовувати захищену бібліотеку кодування для кодування ненадійних даних, перш ніж використовувати їх як атрибут HTML.

Приклад:

```
var htmlEsc = new lib.EncHtm lAttrib()

var attribdata = htmlEsc(input)

< span title = attribdata >
```

3) Уникати JavaScript коду при підстановці ненадійних даних у JavaScript значення. Є деякі JavaScript функції, які ніколи не можуть безпечно використовувати ненадійні дані як свої вхідні дані.

Приклад:

```
< script > alert('…ESCAPE UNTRASTED DATA …') < /script >

< script > x ='…ESCAPE UNTRASTED DATA …'< /script >

< div onmousover = " x =' ESCAPE UNTRASTED DATA' " < /div >
```

Застосування представлених рекомендацій від OWASP зводить до мінімуму можливість атак на розроблені веб-додатки з використанням вразливості міжсайтового скриптингу (XSS).

Список використаних джерел

1. OWASP Cheat Sheet Series. – URL: https://cheatsheetseries.owasp.org
2. OWASP Java Encoder. – URL: https://wiki.owasp.org/index.php/OWASP_Java_Encoder_Project#tab=Use_the_Java_Encoder_Project