مجلة الحاسوب والتقانة العلمية
Scientific Journal of Computer and Technology

الهجمات على تطبيقات الويب فى جانب العميل وكيفية الحماية منها: دراسة استقصائية

Client-Side Web Application Vulnerabilities and how to prevent from them: A Survey

انصاف جمعه صالح[1،2]

استاذ محاضر

insaafict@hotmail.com

فخر الدين عباس محمد[1،2]

استاذ مشارك

fakhry00@gmail.com

كلية علوم الحاسوب وتقانة المعلومات – جامعة النيلين[1]

الخرطوم، السودان[2]

المستخلص:

مع ازدياد استخدام تطبيقات الويب فى مختلف المجالات ازدادت الحاجة لمعرفة المهددات التى تواجهها من اجل العمل على حمايتها منها وذلك لتصاعد الهجمات التى تستهدفها. وتصنف هجمات تطبيقات الويب الى هجمات تستهدف الخادم وهجمات تستهدف العميل. تهدف الهجمات فى جانب العميل الى سرقة معلومات العملاء والاستفادة منها بواسطة المهاجم فى انشطتهم غير القانونية وغير الاخلاقية وتحدث هذه الهجمات نتيجة لتفاعل العميل مع تطبيق ويب الكترونى يحتوى على ثغرات مما يجعله معرضا للهجمات. تهدف هذه الورقة الى تحليل عدد من الدراسات والاوراق العلمية فى مجال الحماية من الهجمات التى تستهدف العميل وهى: هجوم تهجين الموقع ((XSS) Cross Site Scripting) وهجوم تزوير طلب إجتياز الموقع ((CSRF) Cross-Site Request Forgery) و

ضعف التحقق من الهوية وإدارة جلسة الإتصال (Broken Authentication and Session Management) .حيث تمت دراسة (20) ورقة علمية فى الفترة من 2010 وحتى 2020. تم التوصل الى عدد من النتائج من خلال تحليل الاوراق العلمية ومن هذه النتائج ان نسبة كبيرة من الادوات المقترحة يجب تطبيقها فى جانب العميل ويتطلب ذلك استخدام متصفحات خاصة او تنزيل ملحقات معينة للمتصفح مما يعنى تقييد تصفح العميل بهذه الادوات فقط كما ان كما ان استخدام ادوات تعمل فى جانب العميل قد يؤثر على سرعة تصفح العميل للمواقع الالكترونية بالاضافة الى ان الاداة ستوفر حماية للعميل الذى يستخدمها فقط. تم اختبار عدد من الادوات بواسطة مواقع الكترونية (فى بعض الاحيان موقع واحد فقط) تم تطويرها بواسطة مؤلف (مؤلفى) الورقة العلمية. وتم التوصل ايضا الى ان بعض الادوات تستطيع فحص المواقع الالكترونية المكتوبة بلغة واحدة فقط (مثلا لغة جافا Java او بى اتش بى PHP).

الكلمات المفتاحية : تطبيقات الويب، هجوم تهجين الموقع، هجوم تزوير طلب إجتياز الموقع، هجوم تعطل آلية المصادقة وادارة الجلسة.

Abstract

With the increase in the use of web applications in various fields, the need to know the threats they face in order to work to protect them from them has increased, due to the escalation of the attacks targeting them. Web application attacks are categorized into server-side and client-side attacks. Client-side attacks aim to steal customer information and make use of it by the attacker in their illegal and unethical activities. These attacks occur as a result of the customer's interaction with an electronic web application that contains vulnerabilities, making the customer vulnerable to attacks. This paper aims to analyze a number of studies and research papers which are aiming to protect from these attacks, namely: Cross Site Scripting Attack (XSS), Cross-Site Request Forgery (CSRF) attack and Broken Authentication and Session Management. (20) scientific papers were studied from 2010 to 2020. It was concluded a large proportion of the proposed tools are applied on the client’s side and this requires the use of special browsers or the installation of specific browser extensions, which means restricting the customer’s browsing with these tools only, the use of tools that work on the customer side might affect the speed of the customer’s browsing of the websites in addition to that the tool will provide protection for the customer who only uses it. A number of tools were tested with websites (sometimes only one website) developed by the author (s) of the paper. It was also concluded that some tools can check websites written in one language only (for example, Java or PHP).

Keywords: Web Applications, Cross Site Scripting (XSS), Cross-Site Request Forgery (CSRF), Broken Authentication and Session Management.

1. المقدمة

تتعرض تطبيقات الويب للعديد من الهجمات وتصنف هذه الهجمات الى هجمات فى جانب الخادم وهجمات فى جانب العميل وهجمات فى الجانبين معا (اى تستهدف الخادم والعميل فى نفس الوقت). نجد ان الهجمات فى جانب الخادم تستهدف تطبيقات الويب على الخادم وعلى العكس من ذلك الهجمات فى جانب العميل والتى تتم من خلال ثغرات فى تطبيقات العميل التى تتفاعل مع خادم معرض للهجمات او يقوم بمعالجة بيانات يمكن ان تكون ضارة. عادة يبدأ العميل الاتصال مع الخادم مما ينتج عنه الهجوم[1] . من ضمن هذه المخاطر هجوم البرمجة عبر الموقع (Cross Site Scripting(XSS)) وهجوم تزوير طلب إجتياز الموقع ((Cross-Site Request Forgery (CSRF)) وضعف التحقق من الهوية وإدارة جلسة الإتصال (Broken Authentication and Session Management) ،حيث تصنف هذه المخاطر ضمن اخطر 10 مهددات للتطبيقات الالكترونية [4] [6]، كما ان هجوم البرمجة عبر الموقع ((Cross Site Scripting(XSS)) يحتل المرتبة الثالثة فى قائمة أمن القبعات البيضاء (White Hat Security) للعام 2019 بينما يحتل هجوم تزوير طلب إجتياز الموقع ((Cross-Site Request Forgery (CSRF)) المرتبة الثالثة عشر فى نفس القائمة [5].تعتبر هذه الهجمات هجمات فى جانب العميل وذلك لانها تحدث فى جانب العميل وذلك اثناء جلسة تصفحة للموقع الالكترونى والغرض منها الحصول على المعلومات التعريفية للعميل (او المعلومات المتعلقة بوسائل الدفع الالكترونية للعميل) وذلك للاستفادة منها بواسطة المهاجم.

يحدث هجوم البرمجة عبر الموقع ((Cross Site Scripting(XSS)) عندما يقوم الموقع بتضمين بيانات غير موثوقة فى صفحة ويب اخرى دون التحقق منها بصورة سليمة او تحديث صفحة ويب موجودة باستخدام بيانات من المستخدم تم الحصول عليها بواسطة المتصفح. يسمح هذا الضعف للمهاجم بتنفيذ سكريبتات على متصفح الضحية وذلك لسرقة جلسة المستخدم او طمس مواقع الانترنت او اعادة توجيه المستخدم الى مواقع ضارة [4]. بعض انواع هذا الهجوم تستهدف الخادم وبعض انواعه تستهدف العميل ولذلك تم ذكره هنا باعتبار ان العميل معرض لنوع من هذا الهجوم.

يحدث هجوم تزوير طلب إجتياز الموقع ((Cross-Site Request Forgery (CSRF))عند اجبار مُتصفح الضحية على إرسال طلبات (HTTP) مزورة تتضمن ملف جلسة الإتصال (session cookie) وأي معلومات تستخدم للتحقق من هوية المستخدم إلى تطبيقات ويب أخرى مصابة. هذا يسمح للمخترق بإجبار متصفح الضحية على إنشاء طلبات تظهر بأنها صحيحة وصادرة من الضحية ويترتب على ذلك تمكن المخترقين من خداع المستخدمين لِإجراء أى من عمليات تغيير الحالة

المصرح لهم بها، على سبيل المثال، تحديث معلومات الحساب، إتمام طلبات شراء، تسجيل الدخول والخروج[6] .

في غالب الأحيان، يتم تطبيق وظائف التطبيق ذات العلاقة بالتحقق من الهوية أو إدارة جلسات الإتصال بطريقة غير صحيحة، ممايسمح ذلك للمخترقين بسرقة كلمات المرور، أو المفاتيح، أو معرِّف جلسة الإتصال، أو بالإمكان كذلك إستغلال ثغرات أخرى بإنتحال هويات مستخدمين آخرين. مثل هذه الثغرات تجعل بعض أو كل الحسابات عرضة للهجوم، ،عندما ينجح الهجوم سيتمكن المهاجم من فعل كل شيء يستطيع فعله الضحية (صاحب الحساب). الحسابات ذات الصلاحيات العالية تكون عادة هي المستهدفة[6] .

اسهمت هذه الدراسة فى التوصل الى ان اغلب الادوات التى تم تناولها بالدراسة تعمل فى جانب العميل وان الادوات التى تعمل فى جانب العميل تشكل عبء على متصفح العميل مما يؤثر على تجربة التصفح بالنسبة له كما انها ستظل موجود فى جانب الخادم. اما بالنسبة للادوات التى تعمل فى جانب الخادم فهى تتطلب اجراء عمليات فحص ومقارنات مما قد يتسبب فى بطء الخادم. كما ان هناك عدد من الادوات تم اختبارها بواسطة مواقع الكترونية مطورة بواسطة مؤلفى الورقة العلمية. بعض الادوات تستهدف نوع واحد من انواع الهجوم موضوع الدراسة كما ان بعض الدراسات تستطيع اكتشاف الثغرات فقط فى المواقع الالكترونية المكتوبة بلغات محددة.

يتناول الجزء الثانى من الورقة الهجمات فى جانب العميل وهى هجوم تهجين الموقع وهجوم تزوير طلب اجتياز الموقع وضعف التحقق من الهوية وإدارة جلسة الإتصال. يوضح الجزء الثالث اهداف الورقة العلمية. ويتناول الجزء الرابع من الورقة تحليل الاوراق العلمية فى مجال الهجمات والمخاطر المذكورة سابقا مع توضيح للنتائج التى تم التوصل اليها من التحليل. يتناول الجزء الاخير من الورقة النتائج التى تم الحصول عليها بعد دراسة وتحليل الاوراق العلمية التى تم الحصول عليها.

2. الهجمات فى جانب العميل:

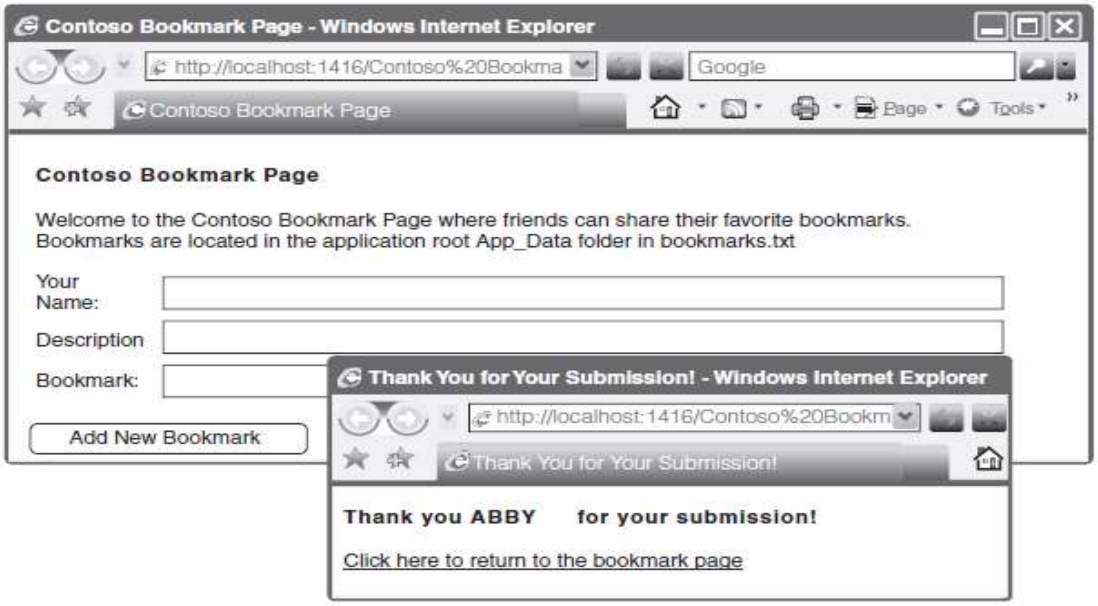
2-1. هجوم تهجين الموقع ((Cross Site Scripting (XSS)

لا تهدف كل الهجمات التى تتم على مواقع الانترنت الى سرقة البيانات او طمس الموقع . على العكس من ذلك بعض الهجمات تستخدم خادم الويب كمنصة لمهاجمة الحواسيب الاخرى التى تتصل به.احد هذه الهجمات هى هجوم تهجين الموقع . حيث يتم حقن خادم الموقع باكواد تقوم بتوجيه الهجوم نحو العملاء الذين يتعاملون معه[1] .

تم تصميم العديد من تطبيقات الويب بحيث تعرض محتوى يتناسب مع المستخدم وذلك من خلال السماح للمستخدم بادخال بيانات يتم الاستعانة بها فى تخصيص محتوى الصفحة له. فمثلا يمكن ان

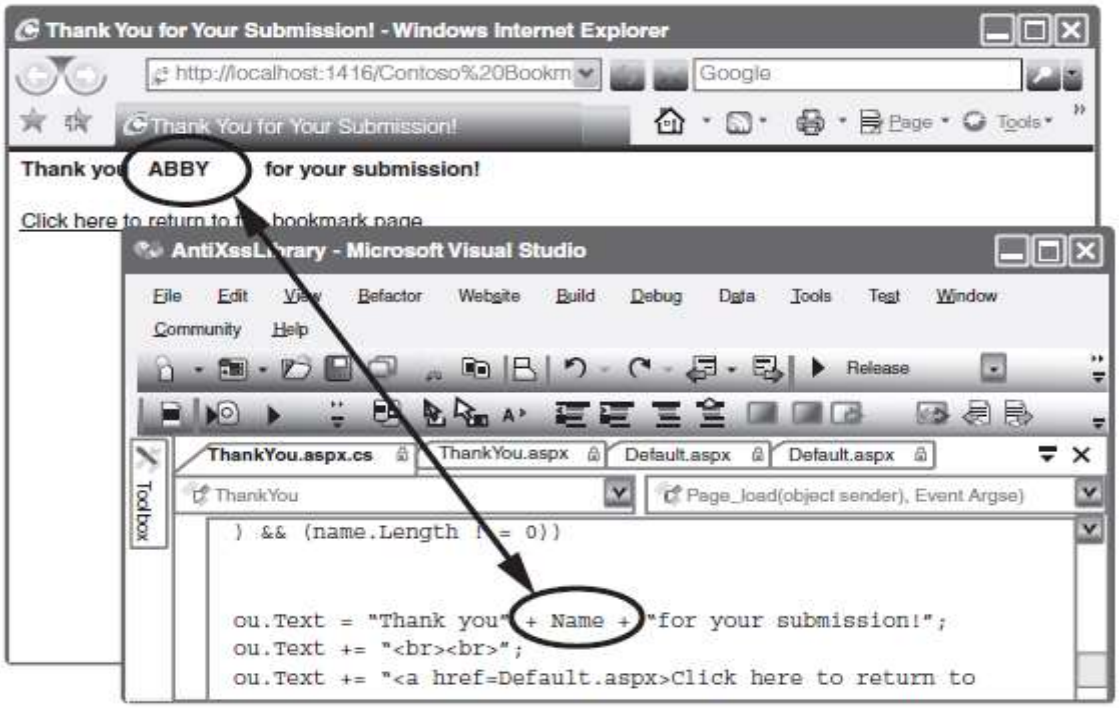
يسمح الموقع للمستخدم بادخال كلمة او جملة بحث ومن ثم يقوم بعرض محتوى عبارة عن نتيجة البحث عن الكلمة التى قام بادخالها المستخدم.[1]

الشكل 1 يوضح تطبيق ويب يسمح للاصدقاء بمشاركة مواقعهم المفضلة . حيث يقوم المستخدم بادخال اسمه ووصف للعنوان ثم العنوان وبعدها يتم عرض رسالة شكر مخصصة للمستخدم. الشكل 2 يوضح كود صفحة الشكر المخصص للمستخدم .[1].



الشكل 1 صفحة ادخال المواقع المفضلة [1]

يحدث هجوم تهجين الصفحة (XSS) عندما يسمح الموقع للمستخدم بادخال بيانات ولا يقوم بالتحقق منها ويقوم بعرضها للمستخدم. عادة ما يستهدف هجوم تهجين الموقع (XSS) المنتديات التى تسمح للمستخدمين باضافة تعليقات. يبدأ الهجوم باضافة المهاجم لتعليق. ويقوم بادراج اكواد برمجيه ضمن التعليق تقوم بافعال ضارة او حتى توجيه المستخدمين الى الموقع الالكترونى للمهاجم. وعندما يقوم الضحية بزيارة المنتدى والضغط على تعليق المهاجم يتم تنزيل الاكواد الضارة الى متصفح الضحية والذى يقوم بتنفيذه.كما يمكن ايضا ان يستفيد المهاجم من هذا الهجوم فى سرقة معلومات المستخدمين الحساسة والتى يحتفظ بها المتصفح عند زيارة موقع ما مثل مواقع التجارة الالكترونية. كما يمكن ان يستخدم المهاجم هذه المعلومات لانتحال هوية صاحبها[1].



الشكل2 المدخلات المستخدمة فى الاستجابة [1]

فى المثال السابق ،عندما يقوم المستخدم بادخال اسمه يتم تمريره بصورة تلقائية الى الكود الذى يقوم بعرضه كاستجابة للمدخلات وذلك دون ان يتم التحقق منه.يمكن ان يقوم المهاجم باستغلال هذه الثغرة وحقن الموقع باكواد ضارة الى متصفح مستخدم آخر، والذى سيقوم بتنفيذها.[1]

2-1-1 انواع البرمجة عبر الموقع (XSS)

1- البرمجة المنعكسة عبر الموقع (Reflected XSS)

يعتبر هذا الهجوم من اكثر الهجمات الاكثر استخدما. يتم هذا الهجوم عندما يقوم التطبيق بقبول مدخل/مدخلات من المستخدم واستخدامها فى صفحة المخرجات التى سيتم انشاءها بواسطة التطبيق. يمكن استغلال هذه الثغرة للقيام بأحد هذه الافعال [2]:

- تنفيذ اكواد جافا سكريبت ضارة.
- تنفيذ اكواد ضارة فى جانب العميل.
- تجاوز وسائل الحماية من هجوم تزوير طلب الصفحة (CSRF).
- طمس الموقع بصورة مؤقتة او غيرها من الاضرار.

تعتبر الحالة الاولى هى الاخطر حيث تسمح للمهاجم بتنفيذ اى كود جافا سكريبت يرغب به على جهاز الضحية. فى هذه الحالة قد يصبح الوضع اسوأ اذا كانت جلسات المستخدم (Sessions) او

الكعكات (Cookies) المهمة متاحة للمهاجم حتى يقوم بسرقتها باستخدام خاصية document.cookie فى لغة جافا سكريبت. اذا اخذنا فى اعتبارنا السطر البرمجى التالى :

window.location='http://evil.example.com/?cookie='+document.cookie

اذا تم تنفيذ هذا الكود بواسطة المتصفح سيتم ارسال كل الكعكات الخاصة بالصفحة المعنية الى صفحة evil.example.com وذلك بمجرد اكتمال تحميل الصفحة. لكن هناك حالة استثنائية اذا ان اى كعكة تحمل خاصية HttpOnly لن يتم ارسالها وذلك ان هذه الخاصية تمنع الوصول الى اى كعكة تحملها بواسطة خاصية document.cookie .[2]

غالبا ما يستهدف هذا الهجوم الخادم ولكن بعض استخدامات هذا الهجوم قد تستهدف العميل ولهذا ورد ذكره هنا باعتبار احتمالية تعرض العميل لهذا الهجوم.

2- البرمجة المخزنة عبر الموقع (Stored XSS)

يختلف الهجوم المخزن (فى بعض الاحيان يسمى بالمستمر) عن الهجوم المنعكس فى انه يستمر فى تكرار نفسه. فبمجرد اضافة الكود الخبيث الى الصفحة فسيظل موجودا ويواصل التنفيذ بصورة دائمة . اى شخص يزور الصفحة سيتأثر بالهجوم. يعتبر الهجوم المخزن شائعا فى المواقع التى يتم تخزين بيانات فيها لفترات طويلة مثل التعليقات والرسائل . [2]

2 -2 هجوم تزوير طلب إجتياز الموقع ((Cross-Site Request Forgery (CSRF)

يجبر هذا الهجوم متصفح المستخدم على ارسال طلبات دون علم المستخدم. يقوم المتصفح باجراء العديد من الطلبات دون علم او موافقة المستخدم ، مثل طلبات الصور والاطارات وغيرها. يقوم هذا الهجوم على ايجاد وصلة تشعبية تقوم باداء افعال مفيدة للمهاجم (وضارة بالمستخدم).[3]

تحتوى صفحات الويب على عشرات – واحيانا مئات – المصادر التى يقوم المتصفح بجلبها تلقائيا لعرض الصفحة. لا توجد قيود على المضيف او النطاق الذى قد تاتى منه هذه المصادر(الصور، ملفات التنسيقات، اكواد جافا سكريبت). فى الواقع ، تقوم بعض المواقع بتخزين محتوياتها الساكنة مثل الصور فى شبكات مخصصة لهذا الغرض ويكون عنوان نطاقها مختلف عن عنوان نطاق الموقع. الشكل 3 يوضح ذلك كما يحتوى ايضا على الكود المصدرى للصفحة.[3]

من هذا المنطلق نجد ان جزئية "اجتياز الموقع" فى هذا الهجوم تقوم بأداء المطلوب من الموقع انجازه. اما التزوير فهو الاستغلال الذى يقوم باضافة اموال الى حساب المهاجم دون التعثر بانظمة اكتشاف التعدى(Intrusion Detection Systems)، او الجدار النارى (Firewall) لتطبيق الويب او اى من انظمة الانذار الاخرى. تمنع سياسات نفس المصدر((SOP) same origin policy) الخاصة

بالمتصفح التفاعل بين المصادر التى يتم جلبها من مصادر مختلفة ولكنها لا تمنع الصفحة من جلب تلك المصادر معا. وعليه فقط يتوجب على المهاجم تزوير الطلب (Request Forgery). ويعتبر محتوى استجابة الموقع ، والمحمى بواسطة سياسة نفس المصدر، غير اساسيا لنجاح الهجوم.[3]

2-2-3العلاقة بين هجوم تهجين الموقع وهجوم تزوير طلب اجتياز الموقع

عادة ما يقوم المهاجمون بدمج هذين الهجومين مع بعضهما البعض. حيث ان كلاهما يستخدم الموقع الالكترونى لتوصيل الاكواد الضارة الى متصفح المستخدم ويجعله يقوم بافعال محددة بواسطة المهاجم. يحتاج هجوم تهجين الموقع (XSS) الى حقن الاكواد الضارة فى اماكن الثغرات فى التطبيق. بينما يقوم هجوم تزوير طلب الموقع (CSRF) باستخدام مواقع اخرى غير مرتبطة ببعضها البعض لتوصيل اكواده الضارة والتى تتسبب فى جعل متصفح الضحية بارسال طلبات الى الموقع المستهدف. ولا يحتاج المهاجم فى تزوير الطلب الى التفاعل مع الموقع المستهدف ولا تحتوى الاكواد المستخدمة فى الهجوم على اى اوامر تحكم مريبة.[3]

الشكل 3 صور تم جلبها من نطاقات مختلفة [3]

تعتبر العلاقة بين الهجومين علاقة تكافلية. يقوم هجوم تزوير طلب الموقع (CSRF) باستهداف وظائف التطبيق والاحتيال على متصفح المستخدم للقيام بطلبات بالنيابة عن المهاجم. بينما تقوم اكواد هجوم تهجين الموقع (XSS) الضارة بحقن نفسها فى متصفح المستخدم واستراق بيانات منه او جعله يتصرف بطريقة معينة. اذا كان الموقع يحتوى على ثغرة تهجين الموقع فهذا يعنى ان كل وسائل الحماية من هجوم تزوير طلب الموقع يمكن تجاوزها. الخلط بين هذين الهجومين قد يقود بعض المطورين الى افتراض ان استخدام وسائل حماية ضد هجوم تهجين الموقع (XSS) سوف تحمى

الموقع ايضا من هجوم تزوير طلب الموقع (CSRF) والعكس صحيح. يعتبر هذين الهجومين منفصلين ويتطلب كل منها حلول مختلفة. [3]

2-3 ضعف التحقق من الهوية وإدارة جلسة الإتصال (Broken Authentication and Session Management)

يعتبر ضعف التحقق من الهوية (Broken Authentication) من الثغرات التى توجد فى تطبيقات الويب الالكترونية وتحدث نتيجة لعدم تهئية متطلبات ادارة الجلسة (Session Management) بصورة سليمة. حيث انه بعد اكتمال عملية المصادقة على المستخدم يتم إنشاء جلسة نشطة لتبادل المعلومات بين الخادم والمستخدم الذى تمت المصادقة عليه. اذا تمكن اى مهاجم من الوصول الى جلسة نشطة خاصة بمستخدم ما وتجاوز خطوات عملية المصادقة فإن هذا يعرف باستغلال تعطل آلية المصادقة فى التطبيق الذى تعرض للهجوم. [8]

يقوم المستخدم بتقديم طلب إنشاء جلسة فى التطبيق الالكترونى من خلال صفحة تسجيل الدخول وفيها يقوم المستخدم بادخال اسمه وكلمة مروره ويتم ارسال هذه المعلومات الى الخادم والذى يقوم بدوره بارسال طلب الى قاعدة البيانات بحثا عن سجل يتطابق مع اسم المستخدم وكلمة المرور التى قام المستخدم بادخالها واذا وجدت هذه البيانات فى قاعدة البيانات يتم انشاء جلسة برقم تعريفى مميز وتخصيصها للاتصال بين المستخدم والتطبيق الالكترونى. وبعد اكتمال هذه العملية يتمكن المستخدم من الدخول الى التطبيق الالكترونى للحصول على خدمات محددة وفقا لصلاحيات يتم تخصيصها له بواسطة مدير التطبيق الالكترونى. وتكون الجلسة مقيدة بفترة زمنية مقيدة يتم تحديدها بواسطة مصمم التطبيق. يقوم المتصفح بحفظ معلومات جلسة المستخدم فى كعكة المصادقة (Authentication Cookie) وذلك طيلة فترة صلاحية الجلسة وعند انتهاء تلك الفترة يتم التخلص من هذه الكعكة. تتم هذه العملية بصورة تلقائية. قد يتمكن المهاجم من الوصول الى جلسات نشطة باستخدام تطبيقات مختلفة مثل : cookie manager, eat my cookie, advanced cookie manager+ وغيرها من البرامج [8] وبالتالى يتمكن من التحكم فى متصفح المستخدم وتنفيذ الاكواد االضارة التى يرغب بتنفيذها.[7]

2-3-1 انواع هجوم تعطل آلية المصادقة وادارة الجلسة

- **هجوم القوى الغاشمة** (Brute Force Attack): يعتمد هذا الهجوم على محاولة تخمين معلومات المستخدم مثل اسم المستخدم وكلمة المرور ورقم بطاقة الائتمان ومفتاح التشفير وذلك بصورة آلية باستخدام برامج. حيث يقوم بارسال قيمة وانتظار استجابة التطبيق الالكترونى واذا كانت القيمة المرسلة غير متطابقة مع القيمة الصحيحة يقوم بارسال قيمة

اخرى وهكذا. تسمح بعض التطبيقات للمستخدمين باستخدام كلمات مرور ضعيفة. يقوم المهاجم بمحاولة كل كلمات القاموس اللغوى كلمة تلو الاخرى حتى يتوصل الى كلمة المرور الصحيحة. قد ينتج عن ذلك الآف وربما ملايين الاحتمالات الخاطئة وعند التوصل الى كلمة المرور الصحيحة يقوم المهاجم باستخدامها للدخول الى حساب المستخدم. يتم استخدام نفس الطريقة لاستنتاج مفاتيح التشفير. [7]

- **اكتشاف الجلسة** (Session Spotting): قد يتمكن المهاجم من التجسس على البيانات التى يرسلها المستخدم(الضحية) على مستوى بروتكول الانترنت (Internet Protocol). فعندما يقوم المستخدم بادخال اسم المستخدم وكلمة المرور فى نموذج تسجيل الدخول وارسال هذه البيانات الى الخادم (وذلك باستخدام بروتكول نقل النصوص المتشعبة الآمن HTTPS). يقوم الخادم بارسال كعكة تحتوى على معرف الجلسة الذى تم تخصيصة للمستخدم ويكون فى صورة نص مشفر. يتمكن المهاجم من الحصول على معرف الجلسة المشفر الخاص بالمستخدم واستخدامه فى انتحال هوية المستخدم الذى كان يقوم بالتجسس عليه. [7]
- **هجوم الاعادة** (Replay Attack): هجوم الاعادة هو احد صور هجمات الشبكات ويتم فيه تكرار او تأخير ارسال البيانات المرسلة لاغراض خبيثة. ويتم ذلك من خلال اعتراض البيانات المرسلة واعادة ارسالها. مثلا اذا افترضنا ان مستخدم ما يرغب بتسجيل الدخول فعليه ادخال اسمه وكلمة مروره ويقوم الموقع او التطبيق الالكترونى بالتحقق من صحة هذه المعلومات. اذا كان المهاجم يتجسس على هذا المستخدم فانه سيتمكن من معرفة معلوماته التعريفية سواء تم ارسالها بصورة مشفرة او غير مشفرة وعلية سيقوم هو ايضا بتسجيل الدخول الى الموقع المستهدف باستخدام المعلومات التعريفية الخاصة بالضحية. [7]
- **هجوم تثبيت الجلسة** (Session Fixation Attack): تثبيت الجلسة هو هجوم يسمح للمهاجم باختطاف جلسة المستخدم المخول. يستغل هذا الهجوم القصور الموجود فى الطريقة التي يدير بها تطبيق الويب معرّف الجلسة ، وبشكل أكثر تحديدًا تطبيقات الويب الضعيفة. حيث انه عند مصادقة مستخدم ، لا يتم تعيين معرف جلسة جديد له، مما يجعل من الممكن استخدام معرّف جلسة موجود مسبقا. يتكون الهجوم من حث المستخدم على المصادقة على نفسه باستخدام معرف جلسة معروف ، ثم اختطاف الجلسة التي تم التحقق من صحتها من خلال معرفة معرف الجلسة المستخدم. يجب على المهاجم توفير معرف جلسة صحيح للضحية. هجوم تثبيت الجلسة هو فئة من فئات هجوم اختطاف الجلسة (Session Hijacking)، والذى يعتمد على سرقة الجلسة المحددة بين العميل وخادم الويب بعد أن يقوم المستخدم بتسجيل الدخول. وبدلاً من ذلك ، يعمل هجوم تثبيت الجلسة على تثبيت جلسة على متصفح الضحية ، وعليه فإن الهجوم يبدأ قبل ان يسجل المستخدم الدخول.[7]

- **اختطاف الجلسة** (Session Hijacking): يُعرف أحياناً باسم اختطاف ملفات تعريف الارتباط (Cookie Hijacking)، وهو استغلال جلسة حاسوب صالحة – يُطلق عليها أحياناً مفتاح جلسة (Session Key) – للحصول على وصول غير مصرح به إلى المعلومات أو الخدمات الموجودة في نظام الحاسوب. على وجه الخصوص ، يتم استخدامه للإشارة إلى سرقة ملف تعريف الارتباط المستخدم لمصادقة مستخدم إلى خادم بعيد. حيث يمكن بسهولة سرقة ملفات تعريف الارتباط الخاصة ببروتكول نقل النصوص المتشعبة (HTTP) والتى تستخدم فى العديد من مواقع الويب للحفاظ على جلسة المستخدم بواسطة مهاجم باستخدام جهاز حاسوب وسيط أو من خلال الوصول إلى ملفات تعريف الارتباط المحفوظة على جهاز حاسوب الضحية. [7]
- **عدم كفاية انتهاء صلاحية الجلسة** (Insufficient Session Expiration): يحدث هذا الهجوم عندما يسمح تطبيق ويب للمهاجم بإعادة استخدام بيانات اعتماد الجلسة القديمة أو معرفات الجلسة للحصول على تخويل. يؤدي هذا الهجوم إلى زيادة تعرض موقع الويب للهجمات التي تسرق أو تعيد استخدام معرفات جلسة المستخدم. ي انتهاء الجلسة نوعين هما: عدم النشاط والثابت. يتم تحديد فترة الانتهاء الثابتة من خلال إجمالي الوقت الذي يمكن أن تكون فيه الجلسة صالحة دون إعادة المصادقة ويتم تحديد فترة انتهاء عدم النشاط بمقدار وقت الخمول المسموح به قبل أن تصبح الجلسة غير صالحة. قد يؤدي عدم وجود انتهاء فترة صلاحية الجلسة المناسبة إلى زيادة احتمالية نجاح بعض الهجمات. تزيد فترة الصلاحية الطويلة للجلسة من فرصة المهاجم في تخمين معرف جلسة صالح بنجاح. [7]

3. اهداف الورقة العلمية

تهدف هذه الورقة الى دراسة وتحليل عدد من الاوراق العلمية التى تتناول اساليب ومنهجيات للحماية من الهجمات والمخاطر التى تستهدف جانب العميل فى تطبيقات الويب الالكترونية وذلك لمعرفة مدى كفاءة وفاعلية تلك المنهجيات فى الحماية من هذه الهجمات والمخاطر وتحديد اوجه القصور (ان وجدت) فى كل منهجية من المنهجيات المقترحة.

4. الحماية من الهجمات فى جانب العميل

تقوم هذه الدراسة على تحليل عدد (20) من الاوراق العلمية فى الفترة من 2010 وحتى 2020 وتم تضمين الاوارق التى تحتوى على الكلمات التالية ضمن عنوان الورقة العلمية:

- Cross Site Scripting
- XSS

- Cross Site Request Forgery
- CSRF
- Broken Authentication and Session Management
- Session Management

تم البحث عن هذه الاوارق والدراسات العلمية باستخدام عدد من قواعد بيانات الاوراق العلمية المشهورة مثل جوجل سكولار (https://scholar.google.com) ودى او اى جاى (https://doaj.org/search/articles) ومحرك البحث جوجل (https://www.google.com). الجدول 1 يوضح الدراسات فى مجالات الهجمات على تطبيقات الويب فى جانب العميل. ويتكون الجدول من الرقم المتسلسل للورقة العلمية وعنوان الورقة العلمية والهجوم المستهدف بالورقة العلمية (هجوم البرمجة عبر الموقع او تزوير طلب إجتياز الموقع او ضعف التحقق من الهوية وإدارة جلسة الإتصال) واسم الاداة او المنهجية المقترحة (ان وجدت) وكيفية الحماية من الثغرة (اما عن طريق الوقاية من الثغرة وذلك لمنع استغلالها بواسطة المهاجمين او اكتشاف الثغرة وذلك للعمل على سدها) ونطاق الاداة (هل تعمل الاداة المقترحة فى جانب العميل ام تعمل فى جانب الخادم – من ناحية تشغيل وتطبيق الاداة) واسم المؤلف (او المؤلفين) وعام اصدار الورقة العلمية.

4-1 الحماية من هجوم البرمجة عبر الموقع (XSS)

الدراسة P1 [9] اقترحت اداة لفحص هجوم التهجين المخزن (Stored XSS) من خلال استخدام نظام مكون من ثلاثة وكلاء(Agents) مستقلين عن بعضهم البعض ويعملون بصورة متكاملة. حيث يقوم الوكيل

جدول 1 الدراسات فى مجالات الهجمات على تطبيقات الويب فى جانب العميل

الدراسة	عنوان الدراسة	الهجوم المستهدف بالاداة/ المنهجية	اسم الاداة/المنهجية	كيفية الحماية	نطاق الاداة/ المنهجية	المؤلف\ المؤلفين	عام الدراسة
P1	A Multi–agent Scanner to Detect Stored–XSS Vulnerabilities	البرمجة عبر الموقع (XSS)	–	اكتشاف الثغرة	فى جانب العميل [9]	E. Gal ‘an وآخرون [9]	2010
P2	A Server- and Browser-Transparent CSRF Defense for Web 2.0 Applications	تزوير طلب إجتياز الموقع (CSRF)	jCSRF [24]	الوقاية من الثغرة	فى جانب الخادم[24]	Riccardo Pelizzi and R. Sekar [24]	2011
P3	Prevention of Cross-Site Scripting Attacks (XSS) On Web Applications in The Client Side	البرمجة عبر الموقع (XSS)	–	الوقاية من الثغرة	فى جانب العميل[10]	S. SHALINI وآخرون [10]	2011
P4	Automated Detection of Session Management Vulnerabilities in Web Applications	ضعف التحقق من الهوية وإدارة جلسة الإتصال	-	اكتشاف الثغرة	فى جانب العميل[25]	Yusuke Takamatsu وآخرون [25]	2012

2013	Yin-Chang Sungوآخرون [23]	فى جانب العميل[23]	الوقاية من الثغرة	Content Box [23]	تزوير طلب إجتياز الموقع (CSRF)	Light-Weight CSRF Protection by Labeling User-Created Contents	P5
2013	RadhaRani Sankuru [21]	فى جانب العميل[21]	الوقاية من الثغرة	-	تزوير طلب إجتياز الموقع (CSRF)	WEB APPLICATION SECURITY - CROSS-SITE REQUEST FORGERY ATTACKS	P6
2014	Raymond Lukanta وآخرون [26]	فى جانب العميل[26]	اكتشاف الثغرة	-	ضعف التحقق من الهوية وإدارة جلسة الإتصال	A Vulnerability Scanning Tool for Session Management Vulnerabilities	P7
2015	Abdalla AlAmeen [22]	فى جانب العميل[22]	الوقاية من الثغرة	RCSR [22]	تزوير طلب إجتياز الموقع (CSRF)	Building a Robust Client-Side Protection Against Cross Site Request Forgery	P8
2015	Wasim Akram Shaikوآخرون [20]	فى جانب الخادم[20]	الوقاية من الثغرة	TwoFish[20]	تزوير طلب إجتياز الموقع (CSRF)	Avoiding Cross Site Request Forgery (CSRF) Attack Using TwoFish Security Approach	P9
2016	D.Kavitha وآخرون [19]	فى جانب الخادم[19]	الوقاية من الثغرة	-	تزوير طلب إجتياز الموقع	Prevention of CSRF and XSS Security Attacks	P10

					(CSRF) و البرمجة عبر الموقع (XSS)	in Web Based Applications	
2016	Jaya Gupta وآخرون [18]	فى جانب الخادم[18]	الوقاية من الثغرة	CSRF Gateway [18]	تزوير طلب إجتياز الموقع (CSRF)	Server Side Protection against Cross Site Request Forgery using CSRF Gateway	P11
2016	Virginia Mary Nadar وآخرون [17]	فى جانب الخادم[17]	الوقاية من الثغرة	-	تزوير طلب إجتياز الموقع (CSRF) وضعف التحقق من الهوية وإدارة جلسة الإتصال	Detection Model for CSRF and Broken Authentication and Session Management Attack	P12
2016	Ankit Shrivastava وآخرون [11]	فى جانب العميل و فى جانب الخادم[11]	الوقاية من الثغرة	–	البرمجة عبر الموقع (XSS)	XSS Vulnerability Assessment and Prevention in Web Application	P13
2016	Shashank Gupta وآخرون [16]	فى جانب الخادم[16]	اكتشاف وسد الثغرة	CSSXC [16]	البرمجة عبر الموقع (XSS)	CSSXC: Context-Sensitive Sanitization Framework for Web Applications against XSS	P14

						Vulnerabilities in Cloud Environments	
2017	M.S. Jasmine وآخرون [12]	فى جانب العميل[12]	اكتشاف الثغرة	XSS-Check add-on [12]	البرمجة عبر الموقع (XSS)	Detecting XSS Based Web Application Vulnerabilities	P15
2017	Rupal R Sharma وآخرون [27]	فى جانب الخادم[27]	اكتشاف الثغرة	-	ضعف التحقق من الهوية وإدارة جلسة الإتصال	Discover Broken Authentication and Session Management Vulnerabilities in ASP.NET Web Application	P16
2018	Bakare K. Ayeni وآخرون [13]	فى جانب الخادم[13]	اكتشاف الثغرة	CrawlerXSS [13]	البرمجة عبر الموقع (XSS)	Detecting Cross-Site Scripting in Web Applications Using Fuzzy Inference System	P17
2018	Virginia Mary Nadarوآخرون [28]	فى جانب العميل[28]	اكتشاف الثغرة و الوقاية من الثغرة[1]	-	تزوير طلب إجتياز الموقع (CSRF) وضعف التحقق من الهوية وإدارة جلسة الإتصال	A Defensive Approach for CSRF and Broken Authentication and Session Management Attack	P18

[1] تعتمد هذه الدراسة اسلوب الوقاية من الثغرة بالنسبة لضعف التحقق من الهوية وادارة جلسة الاتصال واسلوب اكتشاف الثغرة بالنسبة للحماية من تزوير طلب اجتياز الموقع.

2019	Jingchi Zhang وآخرون [14]	فى جانب العميل[14]	اكتشاف الثغرة	–	البرمجة عبر الموقع (XSS)	Cross-Site Scripting (XSS) Detection Integrating Evidences in Multiple Stages	P19
2020	Oluwakemi Christiana Abikoye وآخرون [15]	فى جانب الخادم[15]	اكتشاف وسد الثغرة	-	البرمجة عبر الموقع (XSS)	A novel technique to prevent SQL injection and cross-site scripting attacks using Knuth-Morris-Pratt string match algorithm	P20

الاول ويسمى محلل صفحات الويب Webpage parser بتصفح الموقع الالكترونى الذى نرغب بفحصه بحثا عن المواضع المحتملة (نماذج ادخال البيانات) لحقن الاكواد الضارة فيها وعند العثور على هذه المواضع المحتملة يتم تسجيلها ليتم استخدامها بواسطة الوكيل التالى. يقوم الوكيل الثانى ويسمى حاقن النصوص (Script Injector) بقراءة قائمة المواضع المحتملة للحقن واختيار مجموعة من الهجمات (يتم اختيارها من قائمة تحتوى على مجموعة من الهجمات) وحقن هذه الهجمات فى كل موضع من المواضع المحتملة للحقن ومن ثم يقوم بتسجيلها فى قائمة الهجمات المنفذة. واخيرا يقوم الوكيل الثالث ويسمى المدقق (The Verificator) باسترجاع قائمة الهجمات المنفذة وتصفح الموقع مرة اخرى بحثا عن هذه الهجمات وبعد الانتهاء من التصفح يقوم باستخراج تقرير بالهجمات التى تم العثور عليها. توجد العديد من اوجه القصور فى هذه الطريقة وهى انه عند تنفيذ الوكلاء الثلاثة بصورة متزامنة قد تفشل فى اكتشاف بعض الهجمات المنفذة كما انه عند اكتشاف الهجوم بواسطة هذه الطريقة لن نستطيع معرفة الصفحة التى تحتوى على هذه الثغرة وكذلك اذا كانت نفس الصفحة تحتوى على اكثر من نموذج لادخال البيانات فلن نتمكن من معرفة ايهم يحتوى على الثغرة المكتشفة. بالاضافة الى انه تم اختبارهذه الطريقة على موقع الكترونى واحد (يوجد موقع الكترونى اخر لكن لم يتم الحصول على نتائج عن فحصه باستخدام الاداة المقترحة) وبالتالى تعتبر النتائج بحاجة الى مزيد من التدقيق.

الدراسة P3 [10] اقترحت اداة ملحقة تعمل فى جانب العميل على متصفح موزيلا Mozilla Firefox 1.5. وتقوم هذه الاداة بمنع هجوم التهجين (XSS) من خلال عدم تمرير اى اكواد يشتبه فى انها عبارة عن هجمات الى محرك جافا سكريبت الخاص بالمتصفح وبالتالى عدم تنفيذها. تمت مقارنة اداء هذه الاداة الملحقة بمتصفح موزيلا مع 4 متصفحات اخرى واثبتت النتائج التى تم الحصول عليها ان الاداة المقترحة قد قامت بمنع وازالة العديد من انواع هجوم التهجين (XSS) كما ان تطبيق المنهجية المقترحة لا يؤثر على اداء المتصفح (بعض الحلول التى تعمل فى جانب العميل ينتج عنها بطء فى التصفح). من عيوب هذه المنهجية بما انها تعمل فى جانب العميل فستتمكن من اكتشاف الثغرة للعميل الذى يستخدم هذه الاداة فقط اما العملاء الذين لا يستخدمونها فلن يتمكنوا من معرفة الثغرات الموجودة فى المواقع التى يتصفحونها. وكذلك فإن الثغرة ستظل موجودة فى جانب الخادم مما يجعل الخادم والعملاء الذين يتعاملون معه معرضين للهجوم باستغلال هذه الثغرة.

الدراسة P13 [11] اقترحت هذه الدراسة منهجية هرمية تتكون من العديد من المراحل (عبارة عن ارشادات وخطوات وتقنيات) وتقترح استخدام هذه المنهجية فى عملية تطوير تطبيقات الويب الآمنة فى جانب الخادم وفى جانب العميل وذلك لان استخدام منهجية واحدة يعتبر غير كافي لتأمين

التطبيقات الالكترونية من هجوم التهجين (XSS). لكن بما انها تعتمد على خطوات وتقنيات وادوات وارشادات تستخدم معا للوقاية من الثغرة ولا توجد آلية موحدة (او واحدة) لتطبيق هذه المنهجية لذلك فهى تعتبر معقدة.

الدراسة P14 [16] اقترحت هذه الدراسة اطار جديد للحماية من هجوم البرمجة عبر الموقع (XSS) يسمى CSSXC ويستهدف هذا الاطار البيئة السحابية (Cloud environment). يقوم هذا الاطار باكتشاف كل نقاط الضعف فى التطبيق الالكترونى والتى يتم من خلالها استقبال بيانات من المستخدم (قد تكون ضارة) حيث يعمل من خلال الزحف فى التطبيق الالكترونى واستخلاص كل الصفحات الموجودة فى التطبيق الالكترونى وتحديد المواضع المحتملة لحقن الاكواد الضارة فى الصفحات المستخلصة ومن ثم تتم اعادة صياغة وكتابة الاكواد (الاكواد الخاصة بالمواضع المحتملة للهجوم) بصورة آمنة وذلك لسد الثغرات المحتملة ومنع استخدامها. تم اختبار هذا الاطار باستخدام 4 تطبيقات الكترونية واظهرت النتائج ان تمكنه من اكتشاف وتحييد هجوم البرمجة عبر الموقع بنسبة دقة عالية ونسبة خطأ منخفضة. يعمل هذه الاطار على البيئات السحابية ولم يتم اختبار فاعليته على التطبيقات الالكترونية العادية.

الدراسة P15 [12] تم فى هذه الدراسة تطوير اداة XSS-Check add-on والتى تعمل فى جانب العميل لاكتشاف ثغرات هجوم التهجين فى جلسة المتصفح الحالية للموقع الذى نرغب باكتشاف الثغرات فيه من خلال تحديد هل المدخلات التى تم ادخالها بواسطة المستخدم تم ارجاعها فى صفحة الاستجابة لطلب المستخدم. لكن بما انها تعمل فى جانب العميل فستتمكن من اكتشاف الثغرة للعميل الذى يستخدم هذه الاداة فقط اما العملاء الذين لا يستخدمونها فلن يتمكنوا من معرفة الثغرات الموجودة فى المواقع التى يتصفحونها وبالتالى ستظل الثغرة موجودة فى الخادم.

الدراسة P17 [13] اقترحت اداة تسمى CrawlerXSS لاكتشاف هجوم التهجين الذى يستهدف نموذج كائن المستند (DOM) باستخدام الاستدلال الضبابى (Fuzzy Inference) وتعمل هذه الاداة فى جانب الخادم. اظهرت الدراسة ان الاداة المقترحة افضل من حيث الدقة بمعدل 15% كما ان معدل الايجابية الخاطئة (False Positive) فى هذه الاداة اقل بمعدل 0.01% مقارنة مع 4 ادوات اخرى. تستطيع الاداة المقترحة اكتشاف هجوم التهجين الذى يستهدف نموذج كائن المستند (DOM) فقط ولا تتمكن من اكتشاف الانواع الاخرى من هجوم التهجين.

الدراسة P19 [14] تقوم بتجميع بيانات من الطلب (Request) والاستجابة (Response) لهذا الطلب وذلك لتصنيف هجوم التهجين (XSS) وتمييزه من التعاملات الطبيعية للموقع الالكترونى. حيث تم جمع مجموعات بيانات عن هجوم التهجين (XSS) والتعاملات الطبيعية للموقع الالكترونى واستخلاص خصائص من هذه المجموعات البيانية باستخدام تقنية word2vec ومن ثم استخدام هذه

الخصائص فى تدريب نموذجين من خلال خوارزمية Gaussian mixture، النموذج الاول لتعاملات هجوم التهجين والنموذج الثانى للتعاملات الطبيعية. يقوم كل نموذج من النموذجين بتوليد درجة احتمال لكل تعامل جديد مع الموقع الالكترونى وبناءا على ذلك يتم تحديد مدى مماثلة هذا التعامل الجديد مع التعاملات الطبيعية وتعاملات هجوم التهجين (على حسب النموذج المستخدم) واخيرا يتم تجميع درجات الاحتمال معا لتحسين معدل الاكتشاف. اظهرت الدراسة ان استخدام الاكتشاف الثنائى ومتعدد المراحل يمكن ان يحسن دقة اكتشاف هجوم التهجين.بالاضافة الى تقليل عدد الايجابيات الخاطئة (False Positive) والسلبيات الخاطئة (False Negative). يعاب على هذه الطريقة انها تتمكن من اكتشاف هجوم التهجين المنعكس (Reflected XSS) فقط ولا تتمكن من اكتشاف الانواع الاخرى من هجوم التهجين. بالاضافة الى انها بحاجة الى اجراء مزيد من الاختبارات باستخدام بيانات واقعية منتقلة عبر الشبكة.

الدراسة P20 [15] اقترحت هذه الدراسة منهجية جديدة لاكتشاف هجوم البرمجة عبر الموقع XSSحيث تمت دراسة انواع وانماط مختلفة لهذا الهجوم ومن ثم تم تصميم شجرة تحليل (Parse Tree) بناءا على هذه الانماط. تمت صياغة دالة للتنقية ((Filter) باستخدام خوارزمية KMP لمقارنة السلاسل الحرفية وذلك بالاعتماد على الانماط السابقة. تقوم دالة التنقية باكتشاف ومنع هجوم البرمجة عبر الموقع حيث يتم تمرير كل المدخلات التى يقوم المستخدم بادخالها عبر هذه الدالة واذا كانت نتيجة هذه الدالة بالايجاب (True) يتم حجب المستخدم المعنى وحظر الطلب الذى قام بارساله وعرض رسالة تحذيرية تفيد بذلك. تم اختبارهذه المنهجية على تطبيق تم تطويره بواسطة مؤلفين الورقة العلمية واظهرت النتائج مقدرة هذه الطريقة على اكتشاف ومنع هجوم البرمجة عبر الموقع وتسجيل وحفظ اى محاولة لهذا الهجوم فى قاعدة بيانات مصممة لهذا الغرض وحجب الجهاز المستخدم فى الهجوم باستخدام عنوانه الفيزيائي (MAC). من عيوب هذه الطريقة انه تم اختبارها بواسطة تطبيق ويب الكترونى واحد كما ان هذا التطبيق المستخدم تم تطويره بواسطة الباحثين . بالاضافة الى اسلوب المقارنة المستخدم بحاجة الى مزيد من التوضيح.واخيرا نجد ان هذه الطريقة ستتمكن فقط من اكتشاف هجوم البرمجة الذى يعتمد على مدخلات المستخدم ولن تتمكن من اكتشاف الصور الاخرى من الهجوم التى لا تعتمد على مدخلات المستخدم.

4-2 الحماية من هجوم تزوير طلب إجتياز الموقع (Cross-Site Request (CSRF) Forgery)

الدراسة P2 [24] تم تطوير اداة تسمى jCSRF وهى تعمل كخادم بروكسى فى جانب الخادم ويغنى ذلك عن التعديل فى متصفح العميل او فى الخادم. ويتم تطبيقها بواسطة مدير الموقع الالكترونى ولا تتطلب هذه الاداة من العميل تنزيل اى ملحقات خاصة بالمتصفح او استخدام متصفح محدد كما انها لا تحتاج الى الوصول الى اكواد الموقع الالكترونى للتعديل فيها. عندما يقوم المستخدم بتسجيل الدخول يتم توليد متسلسلة رموز خاصة بهذا المستخدم وعندما يرغب باجراء اى معاملة مع الخادم يتم ارسال طلبه الى هذه الاداة بالاضافة الى المتسلسلة الخاصة به وتقوم الاداة بالتحقق من ان الطلب من مستخدم مخول (الصفحة مستخدمة بواسطة المستخدم المخول) بواسطة المتسلسلة وبناءا على ذلك يتم تمرير طلب المستخدم الى الخادم والا يتم منع الطلب من الوصول الى الخادم (عدم تمريره).

الدراسة P5 [23] تم تطوير اداة Content Box والتى تعتمد على استخدام علامات (Label) لتمكين الخادم من تحديد الطلبات الضارة من الطلبات غير الضارة دون الحاجة الى تغيير المحتويات التى يتم إنشاءها بواسطة المستخدمين وبناءا على هذا التصنيف يتم منع الطلبات الضارة من الوصول الى الخدمات الحرجة فى تطبيق الويب والتى يتم تحديدها بواسطة مدير الموقع الالكترونى. عندما يقوم المستخدم بتسجيل الدخول الى الموقع يتم تخصيص كعكة (Cookie) لمتصفح المستخدم الحالى وعندما يقوم المستخدم بارسال طلب الى الموقع الالكترونى فإن متصفح المستخدم يقوم بالحاق الكعكة الخاصة بالمستخدم مع الطلب الذى قام بارساله بصورة تلقائية ويتم الرجوع الى هذه القيمة المضمنة داخل الطلب لتحديد هل المحتوى موثوق ام غير موثوق ويتم التعامل معه بناء على تصنيفه.فاذا كانت موثوقة يسمح له بالوصول الى الخدمات الحرجة ولا يتم منعه من الوصول اليها.

الدراسة P6 [21] تعتمد هذه الورقة على توليد متسلسلة فريدة (Token) يتم الحاقها بكل طلب يقوم به المستخدم المخول الى الموقع الالكترونى.. وتتكون المتسلسلة من رقم معرف الجلسة وزمن الطلب بالاضافة الى الطابع الزمنى للطلب (Timestamp). وبناءا على هذه المعلومات تكون كل متسلسلة فريدة . وبالتالى يصعب على المهاجم التنبؤ بمتسلسلة صحيحة وعليه لن يتمكن من مهاجمة الموقع الالكترونى.

الدراسة P8 [22] اقترحت هذه الدراسة اداة تسمى RCSR للحماية من هجوم تزوير طلب اجتياز الموقع المنعكس وتعمل هذه الاداة عن طريق تحديد مصدر طلب برتكول نقل النصوص المتشعبة (HTTP) هل هو من نفس التبويب (tab) الخاص بالمستخدم المخول ام انه من تبويب آخر. وتقوم

بمراقبة واعتراض اى يطلب يتم ارساله بواسطة متصفح المستخدم واستخلاص معلومات جلسة المستخدم وارسالها الى الخادم وبناءا على هذه المعلومات يقوم الخادم بتوليد متسلسلة رموز خاصة لجلسة المستخدم. تم تصميم هذه الاداة ليتم استخدامها مع متصفح موزيلا (Mozilla). اظهرت النتائج تمكن الاداة من اكتشاف هجوم تزوير طلب اجتياز الموقع المنعكس ولكن نجد من جانب آخر انها تكون محدودة فقط بجهاز المستخدم الذى يستخدم متصفح موزيلا ويتضمن هذه الاداة كما انها توفر حماية من نوع واحد من انواع هذا الهجوم.

الدراسة P9 [20] بما ان هجوم تزوير طلب اجتياز الموقع (CSRF) يحدث نتيجة لان المصادقة على المواقع تتم بواسطة المتصفح وليس المستخدم تم اقتراح طريقة TwoFish والتى تستخدم للمصادقة على المواقع الالكترونية والتأكد من انها مواقع موثوقة وليست مواقع تم اعدادها بواسطة المهاجمين. عندما يرغب المستخدم بالتأكد من موقع ما فانه يقوم بادخال عنوان هذا الموقع وادخال صورة هذا الموقع وبناءا على ذلك يتم حساب القيمة الهاشية لعنوان الموقع الالكترونى بواسطة MD5 وبعد ذلك يتم تشفير صورة الموقع الالكترونى ومن ثم تتم مقارنة النتائج لتحديد هل الموقع المعنى موثوق وآمن او غير ذلك. ويتم استخراج تقرير بالنتيجة.

الدراسة P10 [19] تم اقتراح نموذج يقوم بتوليد متسلسلة رموز (Token) فريدة لكل حالة من حالات الجلسة ويتم تشفير المتسلسلات باستخدام خوارزمية MD5 وفى كل مرة يرغب فيها العميل بالتعامل مع الخادم يتم ارسال المتسلسلة المشفرة الى الخادم واذا كانت مطابقة للمتسلسلات التى تم توليدها بواسطة الخادم لذلك العميل يتم اكمال الطلب اما اذا كانت غير متطابقة فلا يتم تلبية طلب العميل. اما بالنسبة لهجوم البرمجة عبر الموقع فتتم تنقية المدخلات التى يقوم المستخدم بادخالها وذلك قبل تمريرها الى الخادم. واذا كانت المدخلات تحتوى على وسوم خاصة مثل <script>.....</script> يتم حذفها وبعد ذلك يتم ازالة الرموز الخاصة من مدخلات المستخدم ومن ثم تتم مقارنة المدخلات المنقحة مع انماط محددة للمدخلات الصحيحة واذا تطابقت يسمح بتمريرها والا يتم حجبها.

الدراسة P11 [18] تم تطوير اداة تسمى CSRF Gateway وتعمل هذه الاداة فى جانب الخادم. عندما يبدأ العميل التفاعل مع الخادم يتم توليد متسلسلة رموز (Token) عشوائية خاصة بهذا المستخدم وتضمينها فى كل نموذج ادخال بواسطة وسم خاص تم ابتكاره فى هذه الطريقة وهو وسم <CSRFToken> . تعتمد الاداة على استخدام طبقتين للحماية حيث يتم فى الطبقة الاولى تضمين متلسللة من الرموز فى كل صفحة من صفحات التطبيق الالكترونى ويتم فى الطبقة الثانية تضمين

متسلسلة رموز اخرى فى الجلسة. عندما يقوم العميل بارسال طلب الى الخادم يتم التحقق من متسلسلات الرموز التى تم تخصيصها لجلسة المستخدم واذا تطابقت يسمح للعميل باجراء المعاملة التى يرغب بها والا يتم التعامل مع طلب العميل كهجوم ويتم اغلاق جلسة العميل بصورة تلقائية ويطلب منه اعادة تسجيل الدخول لاجراء اى معاملات اخرى. تم اختبار فعالية هذه الاداة على تطبيق الكترونى تم تطويره بواسطة المؤلفين وبالاستعانة بـــ(OWASP Zed Attack Proxy (ZAP وذلك للقيام بفحص التطبيق الالكترونى من خلال محاولة مهاجمة التطبيق الالكترونى وذلك للتأكد من فعالية الطريقة المقترحة. واظهرت النتائج تمكن الاداة المقترحة من حماية التطبيق الالكترونى من الاشكال المختلفة من هجوم تزوير طلب إجتياز الموقع (CSRF).

الدراسة P12 [17] اقترحت نموذج لمنع هجوم تزوير طلب إجتياز الموقع (CSRF) وضعف التحقق من الهوية وإدارة جلسة الإتصال من خلال تطبيق واتباع عدد من القواعد واجراء عدد من الاختبارات عند استقبال طلبات من العملاء. بالنسبة لهجوم تزوير اجتياز الموقع (CSRF) يتم اختبار طلبات العملاء بالاضافة اختبار مدخلات المستخدم بواسطة نماذج ادخال البيانات. اما بالنسبة لضعف التحقق من الهوية وإدارة جلسة الإتصال فيتم تحديد قواعد لاختيار كلمات المرور بواسطة المستخدمين وادارة البيانات المرتبطة بجلسات المستخدمين. تقدم هذه الطريقة حل لاكثر من مشكلة ولكن نجد ان تطبيقها يتطلب اجراء عدد من الاختبارات والمقارنات فى جانب الخادم لكل طلب يتم ارساله بواسطة المستخدم وبالتالى فقد تؤدى الى حدوث تأخير فى استجابة الطلب بواسطة الخادم.

4-3 الحماية من ضعف التحقق من الهوية وإدارة جلسة الإتصال

الدراسة P4 [25] تهدف هذه الدراسة الى اكتشاف الثغرات المتعلقة بالجلسة فى الموقع الالكترونى (تحديدا تهدف الى اكتشاف هجوم تزوير طلب اجتياز الموقع وهجوم تثبيت الجلسة (Session Fixation)) وذلك من خلال محاكاة وتنفيذ هجمات على الموقع الالكترونى بصورة تلقائية وتم تضمين هذه التقنية ضمن Amberate وهو اطار لفحص تطبيقات الويب. لاكتشاف هجوم تثبيت الجلسة يتم ادخال اسم الجلسة واسم المستخدم وكلمة المرور لكل من المهاجم والضحية. تقوم الاداة بالتفاعل التلقائى (ارسال طلبات واستقبال استجابة الموقع الالكترونى لهذه الطلبات) مع الموقع الالكترونى باتباع عدد من الخطوات مثل تسجيل الدخول والخروج وشن هجوم تثبيت الجلسة على الموقع الالكترونى ومن ثم استخلاص المعلومات الضرورية وبناءا على ذلك يتم تحديد اذا كان الموقع الالكترونى معرض لهذا الهجوم ام لا. لاكتشاف هجوم تزوير طلب اجتياز الموقع يتم ادخال اسم الجلسة واسم المستخدم وكلمة المرور لكل من المهاجم والضحية واسم المتسلسلة السرية والوظيفة التى

سيتم اختبارها فى الموقع الالكترونى بعد ذلك تقوم الاداة بتكرار نفس الخطوات التى تم اتباعها لاكتشاف هجوم تثبيت الجلسة بالاضافة الى التفاعل مع الوظيفة المطلوب اختبارها وشن الهجوم عليه وبناءا على تحليل البيانات التى يتم الحصول عليها من التفاعل ومقارنتها مع البيانات التى تم الحصول عليها من مراقبة تفاعل المختبر (Tester) مع الموقع الالكترونى واذا كان هناك اختلاف فهذا يعنى ان الموقع الالكترونى معرض لهذا الهجوم. تتطلب هذه الطريقة ان يقوم الشخص المكلف باختبار النظام باستخدام الموقع الالكترونى وكل وظيفة يرغب باختبارها فى الموقع الالكترونى وذلك حتى تتمكن الاداة من اكتشاف الثغرات بصورة صحيحة وفعالة هذا مع العلم انه يجب ادخال المعلومات الضرورية المتعلقة بكل هجوم فى الاداة حتى تتمكن من التفاعل مع الموقع الالكترونى بصورة تلقائية.

الدراسة P7 [26] تهدف هذه الدراسة الى اكتشاف هجوم تزوير طلب اجتياز الموقع وهجوم تثبيت الجلسة (Session Fixation) وعدم كفاية خصائص الكعكات (insufficient cookies attributes). تم فى هذه الدراسة تطوير اداة وتضمينها داخل الاداة المسماة بـــ Nikto وهى اداة مفتوحة المصدر. يتكون الحل المقترح فى هذه الدراسة من جزئين، الجزء الاول عبارة عن ملحق للمتصفح (Browser Extension) يعمل فى متصفح كروم (Google Chrome) ويقوم باكتشاف الثغرات الموجودة فى التطبيق الالكترونى بالاضافة الى شن هجمات على الموقع الالكترونى للتأكد من وجود الثغرات التى تم اكتشافها وايضا يقوم بتسجيل كل التفاعلات التى تتم بين المستخدم والمتصفح. كما يقوم ايضا بتوليد اكواد اختبارية (testing script) وتزويد الاداة بهذه الاكواد ليتم استخدامها فى اختبار وفحص الموقع الالكترونى وبعد الانتهاء من ذلك يقوم باستخراج التقرير النهائي للفحص. الجزء الثانى يتمثل فى الاداة Nikto، حيث تم تطوير ملحق لهذه الاداة يحمل اسم Session Management Plugin ويقوم باداء نفس المهام التى يقوم بها ملحق المتصفح عدا المهام المتعلقة بشن الهجمات على التطبيق الالكترونى.

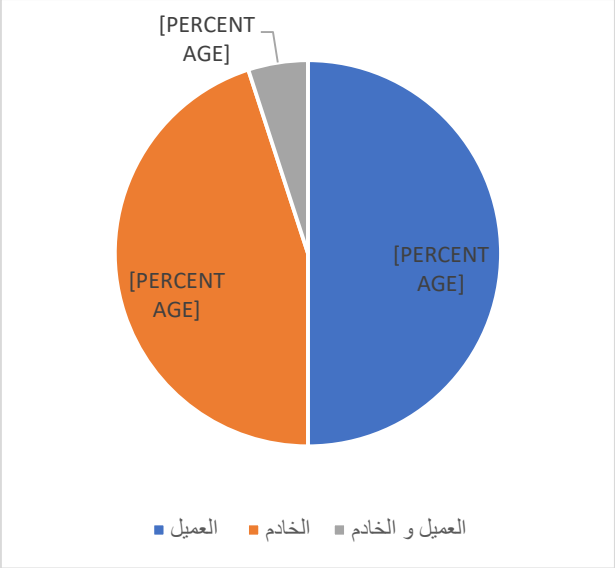
الدراسة P16 [27] اقترحت هذه الدراسة خوارزمية لاكتشاف ضعف التحقق من الهوية وإدارة جلسة الإتصال من خلال فحص الموقع الالكترونى والملفات المصدرية لاكواد الموقع الالكترونى المكتوبة بلغة ASP.NET . تتكون الخوارزمية المقترحة من 11 خطوة وتمثل هذه الخطوات اختبارات يتم اجراؤها على ملفات الاكواد بحثا عن صفات وخصائص معينة فى هذه الملفات والقيم المسندة لهذه الخصائص مثلا يتم البحث عن خاصية الاكمال التلقائي (autocomplete) فى نماذج ادخال البيانات وهل هذه الخاصية مفعلة ام معطلة واذا كانت مفعلة فيتم اعتبارها ثغرة وتضمينها فى تقرير الفحص النهائي. تمت برمجة هذه الخوارزمية بلغة بايثون.

الدراسة P18 [28] تهدف هذه الدراسة الى اكتشاف تزوير طلب إجتياز الموقع (CSRF) وضعف التحقق من الهوية وإدارة جلسة الإتصال. لاكتشاف تزوير طلب إجتياز الموقع (CSRF) تم تطوير Packet tracker module لاكتشاف اى طلبات تحتوى على اكواد ضارة يتم ارسالها بواسطة المهاجمين واذا كان لايحتوى على اكواد ضارة يتم التحقق من الطلب بواسطة request checker policy والتى تقوم بالتحقق من مطابقة الطلب لمجموعة من القواعد والسياسات وبعد ذلك يتم تنفيذ الطلب. اما بالنسبة لضعف التحقق من الهوية وإدارة جلسة الإتصال فيتم استخدام authentication module وهو يساعد المستخدم من خلال توليد كلمات مرور قوية للمستخدمين وذلك لان كلمات المرور الضعيفة يسهل اكتشافها وتخمينها بواسطة المخترقين، كما تتم متابعة محاولات المخترقين الحصول على كلمات المرور وايقاف اى تعاملات لهم باستخدام كلمات مرور المستخدمين المخولين وذلك لمنع اى ضرر يمكن ان ينجم عن ذلك. واخيرا يتم استخراج تقرير عن الثغرات التى تم اكتشافها.

4-4 نتائج التحليل

تم التوصل الى النتائج التالية من خلال تحليل الاوراق والدراسات العلمية:

- الادوات التى تعمل فى جانب العميل قد تتطلب من العميل تنزيل ملحقات معينة واستخدام متصفحات معينة مما يشكل تقييدا للمستخدم كما انها توفر حماية فقط للمستخدم الذى يتصفح الموقع الالكترونى بواسطتها بالاضافة الى انها قد تتسبب فى بطء عملية التصفح بالنسبة للمستخدم. واذا كانت الثغرة موجودة فى الخادم فلن يتم التعرف عليها فى جانب الخادم وبالتالى ستظل موجودة وتسبب فى الحاق اضرار بسمتخدمين آخرين.
- الادوات التى تعمل فى جانب الخادم قد تتسبب فى بطء استجابة الخادم لطلبات المستخدمين وذلك لانها تتطلب اجراء عمليات واختبارات معينة قبل القيام بتلبية طلبات المستخدمين ويشكل ذلك عبئا اضافيه علي الخادم.
- بعض الادوات المقترحة تستطيع اكتشاف الثغرات الموجودة فى مواقع الكترونية مكتوبة بلغة برمجية واحدة فقط وبالتالى لا يمكن استخدامها لفحص مواقع الكترونية مكتوبة بلغات برمجية اخرى.



الشكل 4 مواضع تطبيق الادوات المقترحة

- بعض الادوات يتم اختبارها بواسطة مواقع الكترونية (غالبا يكون موقع واحد او عدد من الصفحات المترابطة) يتم تطويرها بواسطة مؤلفى الورقة العلمية مما يجعل هناك ضرورة لاختبارها بواسطة تطبيقات ومواقع الكترونية اخرى لتأكيد النتائج التى تم التوصل اليها بواسطة الدراسة.
- بعض الادوات تستهدف نوع واحد او صورة واحدة من صور الهجوم المستهدف بالدراسة.
- لازالت الهجمات فى جانب العميل تمثل تحديا بالنسبة للمواقع الالكترونية الى يومنا هذا وانها لازالت من المواضيع التى تحظى باهتمام الباحثين فى مجال امن التطبيقات الالكترونية.
- 50% من الدراسات العلمية اقترحت ادوات تعمل فى جانب العميل بينما اقترحت 45% من هذه الدراسات ادوات تعمل فى جانب العميل و5% من الدراسات اقترحت ادوات تعمل فى جانب الخادم والعميل معا. الشكل 4 يوضح مواضع تطبيق الادوات المقترحة.

5. الخلاصة

تتعرض تطبيقات الويب الالكترونية للعديد من الهجمات وتستهدف هذه الهجمات اما جانب الخادم او جانب العميل. تعرضت هذه الورقة لعدد من الادوات المقترحة للحماية من الهجمات التى تستهدف العميل وهى هجوم تهجين الموقع (Cross Site Scripting (XSS)) وهجوم تزوير طلب إجتياز الموقع (Cross-Site Request Forgery (CSRF)) و ضعف التحقق من الهوية وإدارة جلسة الإتصال (Broken Authentication and Session Management). حيث تمت دراسة (20) ورقة علمية تم التوصل الى ان 50% من الادوات التى تم تناولها بالدراسة تعمل فى جانب العميل

و45% من الادوات تعمل فى جانب الخادم و5% من الادوات تعمل فى الجانبين معا. وايضا ان الادوات التى تعمل فى جانب العميل تشكل عبء على متصفح العميل مما يؤثر على تجربة التصفح بالنسبة له كما انها ستظل موجود فى جانب الخادم. اما بالنسبة للادوات التى تعمل فى جانب الخادم فهى تتطلب اجراء عمليات فحص ومقارنات مما قد يتسبب فى بطء الخادم. كما ان هناك عدد من الادوات تم اختبارها بواسطة مواقع الكترونية مطورة بواسطة مؤلفى الورقة العلمية. بعض الادوات تستهدف نوع واحد من انواع الهجوم موضوع الدراسة كما ان بعض الدراسات تستطيع اكتشاف الثغرات فقط فى المواقع الالكترونية المكتوبة بلغات محددة.

المصادر والمراجع

1. Mark Ciampa, CompTIA® Security+ Guide to Network Security Fundamentals, Cengage Learning, Fifth Edition, 2015.
2. Prakhar Prasad, Mastering Modern Web Penetration Testing, Packt Publishing, First Edition, 2016
3. Mike Shema, The Seven deadliest Web Application Attacks, Syngress, 2010.
4. OWASP, OWASP top 10 application security risks - 2017, https://www.owasp.org/index.php/Top_10-2017_Top_10, 2019.
5. WhiteHat Security, Top 10 vulnerabilities of 2019, https://info.whitehatsec.com/Content-2020-Top10Vulnsof2019WP_LPNew.html, 2020.
6. OWASP, OWASP top 10 application security risks - 2013, https://www.owasp.org/www-pdf-archive/OWASP_Top_10_-_2013.pdf, January 10 2021.
7. Bharti Nagpal, Nanhay Singh, Naresh Chauhan, Pratima Sharma. Preventive Measures for Securing Web Applications Using Broken Authentication and Session Management Attacks: A Study. International Conference on Advances in Computer Engineering & Applications, 2014.
8. Md. Maruf Hassan, Shamima Sultana Nipa, Marjan Akter, Rafita Haque, Fabiha Nawar Deepa, Mostafijur Rahman, Md. Asif Siddiqui, Md. Hasan Sharif. Broken Authentication and Session Management Vulnerability: A Case

Study of Web Application. International Journal of Simulation: Systems, Science & Technology, 2018.

9. E. Gal´an, A. Alcaide, A. Orfila, J. Blasco, A Multi–agent Scanner to Detect Stored–XSS Vulnerabilities, 2010 International Conference for Internet Technology and Secured Transaction, 2010.
10. S. SHALINI, S. USHA, Prevention of Cross-Site Scripting Attacks (XSS) On Web Applications in The Client Side, International Journal of Computer Science Issues, Volume 8, Issue 4, 2011.
11. Ankit Shrivastava, Santosh Choudhary, Ashish Kumar, XSS Vulnerability Assessment and Prevention in Web Application, 2nd International Conference on Next Generation Computing Technologies, 2016.
12. M.S. Jasmine, Kirthiga Devi, Geogen George, Detecting XSS Based Web Application Vulnerabilities, International Journal of Computer Technology & Applications, Vol 8(2),291-297, 2017.
13. Bakare K. Ayeni, Junaidu B. Sahalu, and Kolawole R. Adeyanju, Detecting Cross-Site Scripting in Web Applications Using Fuzzy Inference System, Journal of Computer Networks and Communications, Volume 2018, 2018.
14. Jingchi Zhang, Yu-Tsern Jou, Xiangyang Li, Cross-Site Scripting (XSS) Detection Integrating Evidences in Multiple Stages, Proceedings of the 52nd Hawaii International Conference on System Sciences, 2019.
15. Oluwakemi Christiana Abikoye, Abdullahi Abubakar, Ahmed Haruna Dokoro, Oluwatobi Noah Akande and Aderonke Anthonia Kayode, A novel technique to prevent SQL injection and cross-site scripting attacks using Knuth-Morris-Pratt string match algorithm, EURASIP Journal on Information Security, 2020.
16. Shashank Gupta, B. B. Gupta, CSSXC: Context-Sensitive Sanitization Framework for Web Applications against XSS Vulnerabilities in Cloud Environments, Procedia Computer Science, 2016.
17. Virginia Mary Nadar, Madhumita Chatterjee, Leena Jacob, Detection Model for CSRF and Broken Authentication and Session Management Attack,

International Journal of Computer Science and Information Technologies, Vol. 7 (4), 1801-1804, 2016.

18. Jaya Gupta and Suneeta Gola, Server Side Protection against Cross Site Request Forgery using CSRF Gateway, Journal of Information Technology & Software Engineering, 2016.
19. D.Kavitha, M.R.Akshaya, M.Karthick, K.Baghya, K.Gomathi Raja Eswari, Prevention of CSRF and XSS Security Attacks in Web Based Applications, International Journal of Innovative Research in Science, Engineering and Technology, Vol. 5, Issue 3, March 2016.
20. Wasim Akram Shaik, Rajesh Pasupuleti, Avoiding Cross Site Request Forgery (CSRF) Attack Using TwoFish Security Approach, International Journal of Computer Trends and Technology (IJCTT) – volume 25 Number 2 – July 2015.
21. RadhaRani Sankuru, WEB APPLICATION SECURITY -CROSS-SITE REQUEST FORGERY ATTACKS, International Journal of Computer Science & Engineering Technology, Vol. 4 No. 08 Aug 2013.
22. Abdalla AlAmeen, Building a Robust Client-Side Protection Against Cross Site Request Forgery, International Journal of Advanced Computer Science and Applications, Vol. 6, No. 6, 2015.
23. Yin-Chang Sung, Michael Cheng Yi Cho, Chi-Wei Wang, Chia-Wei Hsu, Shiuhpyng Winston Shieh, Light-Weight CSRF Protection by Labeling User-Created Contents, 7th International Conference on Software Security and Reliability, 2013.
24. Riccardo Pelizzi and R. Sekar, A Server- and Browser-Transparent CSRF Defense for Web 2.0 Applications, Proceeding of the 27th Annual Computer Security Applications Conference, P257-P266, 2011.
25. Yusuke Takamatsu, Yuji Kosuga, Kenji Kono, Automated Detection of Session Management Vulnerabilities in Web Applications, Tenth Annual International Conference on Privacy, Security and Trust, 2012.

26. Raymond Lukanta, Yudistira Asnar, A. Imam Kistijantoro, A Vulnerability Scanning Tool for Session Management Vulnerabilities, International Conference on Data and Software Engineering, 2014.
27. Rupal R Sharma, Ravi K Sheth, Discover Broken Authentication and Session Management Vulnerabilities in ASP.NET Web Application, International Journal of Scientific Research in Science and Technology, Volume 3, Issue 1, 2017.
28. Virginia Mary Nadar, Madhumita Chatterjee and Leena Jacob, A Defensive Approach for CSRF and Broken Authentication and Session Management Attack, Advances in Intelligent Systems and Computing, volume 696, 2018.