جلسة مطورة لدعم أمن تطبيقات الويب

د.عمار ريماوي

الملخص

تستخدم تطبيقات الويب تقنية الجلسات لضمان أمن التواصل مع المستخدم لحماية المستخدم والتطبيق. وبالرغم من ذلك تعاني بعض هذه التطبيقات من اختراقات ومشاكل أمنية عديدة.

في هذه المقالة ألقينا الضوء على مشكلات أمن تطبيقات الويب الهامة جداً من أجل حماية تطبيقاتنا من الاختراقات أو أية أمور مزعجة تؤثر على عمل هذه التطبيقات ومتابعة كل ما يتم على هذه التطبيقات من تطورات لحمايتها بالشكل الأمثل. كما عرضنا أهم استراتيجيات أمن تطبيقات الويب وتحديد المسؤول عن البيانات الحساسة الموجودة في التطبيقات أو من الذي يقوم باختراق المواقع والتخريب المتعمد لها. وقدمنا تطويراً هاماً على جلسة تطبيقات ويب إذ تقوم هذه التقنية بجلب بيانات عن المستخدم ورصد نشاطاته وتسجيل كل ذلك في قاعدة بيانات منفصلة بهدف تأمين المحتوى من الاختراقات وإجراء إحصائيات متنوعة ورصد تفاعلات المخترقين في حال اكتشافها أو عند ورود شكوى تتعلق بها.

كلمات مفتاحية: تقنية الجلسات، كوكيز، أمن مواقع الويب.

A Developed Session to Support Web Applications Security

Abstract

Web applications usually use sessions' technique to ensure user access security to protect both users and applications. Nevertheless, some of these applications suffer of hijacking and many security problems. In this paper, we preview the most important web applications problems to protect web applications from hijacking or any annoying things may affect on these applications. Website should be paid attention to follow up all changes to protect them optimally. Also, we preview the most important web security strategies. After that, a web applications session was developed to fetch some user information and trace his activities, and then record them in an individual database. This was made to secure the content from hijacking, perform various statistics and trace hijackers' activities in case of detection them or when a complaint was came.

Key words: Sessions, Web Application Security, Cookies

1. مقدمة

كان لانتشار الانترنت الواسع دوراً كبيراً في انتشار تطبيقات الويب والتي أصبحت تلبي كافة احتياجات الناس مثل إمكانية تبادل الآراء والحوار والنقاش مع أعداد كبيرة جدا من الناس مما يساعد على التشارك بالمعلومات وتبادل الآراء، في مختلف أنحاء العالم مهما بعدت المسافات كما أصبحت الخدمات الأساسية تقدم من خلال هذه التطبيقات كالبيع والشراء والحسابات المصرفية مما جعلها تواجه أخطار عديدة أهمها اختراق أمن هذه التطبيقات [1,2] مما يهدد بالوصول إلى البيانات الهامة المتعلقة بالمستخدمين كاسم المستخدم وكلمة المرور مما يسمح بالوصول إلى البطاقات الائتمانية والحسابات المصرفية وغيرها.

2. أهمية البحث وأهدافه

تأتي أهمية البحث مع أهمية أمن المواقع حيث أصبح من الضروري محاولة الحماية المطلقة لبعض المواقع الهامة مثل مواقع البنوك ومواقع التعليم الالكتروني ومواقع الخدمات الأخرى منعاً للاختراقات والسرقات التي يمكن أن تحدث من خلال هذه المواقع. ويمكن تلخيص أهداف البحث بما يلي [1,2,3]:

- إمكانية حماية التطبيقات من الـ Griefers , Trools , Pranksters.
- إمكانية حماية التطبيقات من الـ Malware.
- تأمين محتوى التطبيقات من الاختراقات.
- تسجيل جميع تحركات المستخدم على هذا الموقع.
- إمكانية القيام بإحصائيات متنوعة.

3. طرائق البحث و مواده

القينا الضوء في هذه الفقرة على أهم مشكلات أمن تطبيقات الويب الهامة لحماية تطبيقاتنا من الاختراقات أو أية أمور مثيرة للقلق والتي قد تؤثر على عمل هذه التطبيقات ومتابعة كل ما يتم على هذه التطبيقات من تطورات لحمايتها بطريقة نأمل أن تكون الأمثل. كما عرضنا أهم استراتيجيات أمن تطبيقات الويب وتحديد المسؤول عن البيانات الحساسة الموجودة في التطبيقات أو من الذي يقوم باختراق المواقع والتخريب المتعمد لها. وقدمنا تطويراً هاماً على جلسة تطبيقات ويب والتي هي الفكرة الأساسية من هذه المقالة. تقوم جلسة تطبيقات ويب بجلب بيانات عن المستخدم ورصد نشاطاته وتسجيل كل ذلك في قاعدة بيانات منفصلة بهدف تأمين المحتوى من الاختراقات وإجراء إحصائيات متنوعة ورصد تفاعلات المخترقين في حال اكتشافها أو عند ورود شكوى تتعلق بها.

3.1 مشكلات أمن مواقع الويب:

نعرض في هذه الفقرة بعض القضايا التي ينبغي أن ننوه عليها والتي لها صلة كبيرة بالموضوع.

أولا: استحالة تحقيق الأمن المطلق

عندما ترسل الأرقام الثنائية للمعالج لا يفرق بين 1 الأولى و1 الأخيرة مثلاً، فهما متماثلان، ولا يمكن تميزها بسبب عدم وجود كتابة يدوية تحليلية أَو بصمات أَو شهادة الأصالة. ويتم الاختراق، إذا استبدل المهاجم أحد تلك الأرقام 1 بشكل سري مع الـ0، وتكون البرامج العادية للمعالجة ليست ذات سلطة لمعرفة فيما إذا كانت أن هذه الـ0 حقيقيا أم لا. إن البرامج الجيدة (التي تأخذ هذه القضية بالحسبان) التي كتبت من قبل المبرمجين المحترفين هي التي تميز

ذلك، إذ تقوم هذه البرامج بمقارنة الموقع الآخر في الذاكرة، ومعرفة أن هذه الأرقام عدلت أم لا. فإذا طُبّقت هذه المراقبة بشكل سيئ، أو لم تنفذ نهائياً، فإن عملية الاختراق قد تمت دون اكتشافها.

ثانيا: توقف الخدمة

هناك العديد من أنواع الهجوم DOS نذكر أهمها [4]:

- **استهلاك عرض الحزمة :**

يقوم المهاجم باستهلاك كامل عرض الحزمة في نظام شبكة الضحية وذلك بإغراق شبكة الضحية بكمية هائلة من الطلبات مثل ...GET, SYN : وغيرها، مما يؤدي إلى استهلاك كامل لعرض الحزمة وإيقاف الموقع المستهدف أو النظام المهاجم تماماً عن العمل، ويكون ذلك إما بالهجوم مباشرة Direct Attack حيث ينتصر في هذه الحالة من لديه عرض حزمة أكبر (مثلاً 56 Kbps في مواجهة 56 Kbps)، أو القيام بربط العديد من المواقع من أجل إغراق نظام الضحية إذ يقوم المهاجم باستخدام أنظمة البث broadcast في شبكات أخرى من أجل تضخيم الهجوم، ويستفيد في هذه الحالة من عرض حزمة تلك الشبكة .

- **استهلاك الموارد:**

يعتمد هذا النوع من الهجوم على استهلاك الموارد في نظام الضحية عوضا عن استهلاك عرض الحزمة. وأهم الموارد المستهدفة : CPU, Memory, Kernel, File system وغيرها، ويؤدي انخفاض الموارد في النظام إلى عدم استقراره وانهياره في نهاية الأمر .

- **الثغرات البرمجية في مكونات النظام :**

لا يوجد نظام أو برنامج خالٍ تماماً من الثغرات مهما بلغت دقة تصميمه. هناك عدة طرق لاستغلال هذه الثغرات كأن نقوم مثلا بإرسال Packets غير متوافقة مع المعايير القياسية لـ بروتركول TCP/IP المحددة من قبل RFC إلى نظام الضحية مما يؤدي إلى نتائج تختلف حسب نوع البرنامج من توقف الخدمة، أو توقف النظام، أو ضياع المعلومات، أو فيضان المكدس stack وغير ذلك .

ثالثا: تأثيرات Griefers , Trools , Pranksters على تطبيقات الويب:

المستخدمون المعروفون بالـPranksters Trools , Griefers هم الأكثر إزعاجاً على الرغم من قلة خطرهم ويستطيعون القضاء على المتعة في تطبيقات الويب بشكل مباشر وفوري ولابد من توضيح المفاهيم التالية:

- **الـ Griefers:** هم المستخدمون الذين يستمتعون بإيذاء الآخرين ومهاجمتهم. فعندما تلاحظ وجود بعض المستخدمين المجهولين على الشبكة Online والذين يختبئون تحت اسم مجهول Screen Name فإن هؤلاء هم الـGriefers إذ يقومون بإزعاج شديد ومؤذٍ جداً وذلك بإرسال التعليقات غير المفيدة مما يقلق المستخدم المتابع بغرض الاستفادة.
- **الـ Trools:** يستمتعون بكونهم مهاجمين ويقوم هؤلاء بسرقة اسمك أثناء التعليقات في المنتديات بالإجابة عنك على أسئلة موجهة إليك مما يجعل إجاباتك غير صحيحة مما قد تعرضك المسائلة القانونية.

- **الـ Pranksters:** يقوم هؤلاء بإدخال تعليمات بلغات مثل Html أو Javascript إلى مايسمى Plain Text، وذلك بغية تشويه مظهر صفحة الانترنت أو قد يأتون بطرق عديدة بهدف إلهاء الآخرين عما كان من المفروض أن يكون عملاً مهماً.

رابعا: تأثير الافتراءات وسوء التخزين:

- **الافتراء (Defamation):** هو استخدام المحترفين للتطبيقات الخاصة بك لإيذاء الآخرين سواء على مستوى الأشخاص أو المؤسسات عن طريق الدمى المتحركة sock puppets (وهي إجراءات تستخدم غالباً في عمليات التصويت). إن عملية التعليق Posting أمر اعتيادي لا مشكلة فيه، إلا أن المشكلة هي حصول عملية Posting باسمك ويحوي الكثير من المواضيع غير الصحيحة وإن استطعت إزالتها إثر ملاحظتك لها، فقد تعرضك للمحاكمة والمسؤولية، على الرغم من إنك لست من قام بتعليق الرسالة.
- **سوء التخزين (Abuse of Storage):** قد تتيح تطبيقات الويب أماكن خاصة للتخزين مقدمة للمستخدمين (كأن تسمح لهم باستخدام مساحات مجانية لتخزين مواقع إنترنت تجريبية) ومواقع كهذه تجذب العديد من هؤلاء المزعجين الذين يرغبون بتخزين الوثائق غير الشرعية والمثيرة للفوضى، كما يستخدمون المساحات المجانية التي توفرها هذه المواقع عوضاً عن الدفع من أجل الحصول عليها.

خامسا: تأثيرات (Malware) (الفيروسات والديدان و برمجيات التجسس وحصان طروادة

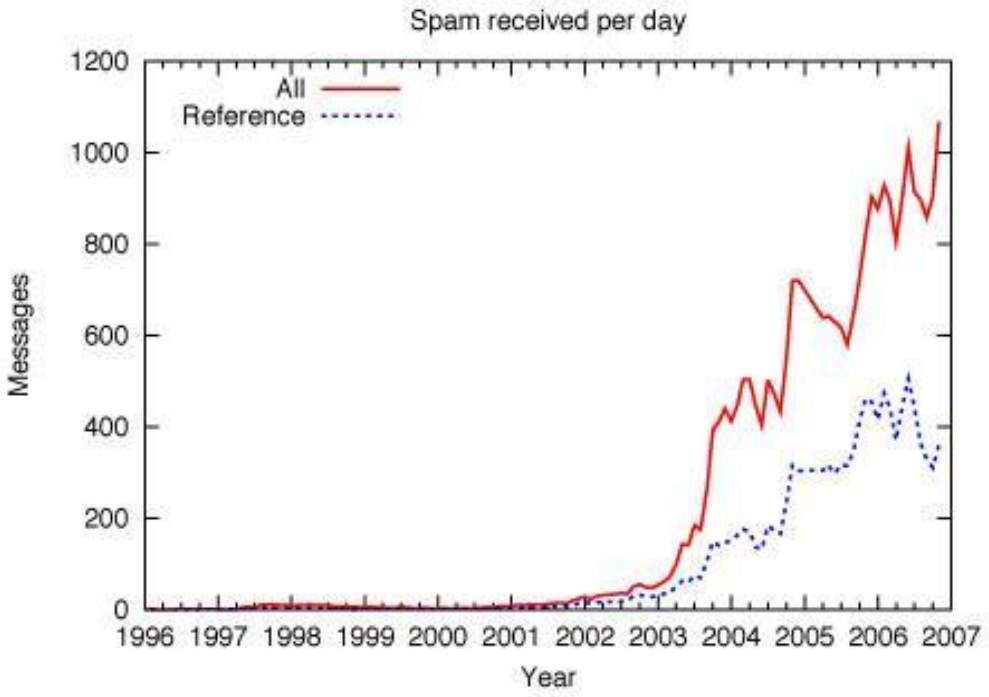
الدودة Worm أو الفيروس Virus هي برامج صغيرة تقوم بإنزال نفسها على الكمبيوتر وقد يحصل ذلك في أغلب الأحيان من خلال ملحقات الرسائل الالكترونية Attachments أو من خلال تحميلها بطريقة ضمنية عند تحميل التطبيقات أو البرامج، والهدف الرئيسي للفيروس أو الدودة هو أنه يعمل على نسخ نفسه ويتضاعف محاولاً الانتقال إلى أجهزة أخرى، والهدف الثانوي هو نشر الفوضى على جهاز المضيف، من خلال حذف وتعديل الملفات، وفتح الباب لتدفق الملفات والرسائل غير المرغوبة إلى جهاز المضيف، أو الرسائل المنبثقة Popping up من مختلف الأنواع، مما يجعل الجهاز المصاب عرضة لإرسال هذه الفيروسات أو الديدان إلى أجهزة أخرى. أما Spyware فهي البرمجيات التي تتجسس على الأجهزة وترسل معلومات عنها لتسهل عمليات الاختراق فيما بعد و Trojans برامج خبيثة تختفي وراء برمجة أو بيانات غير مؤذية بحيث تتمكن من السيطرة ومن ثم تعمل أشكالها المختارة للضرر، مثل تخريب ملفات الإقلاع على قرصك الصلب. وأخيراً جميع الـMalware تجعل من الجهاز المصاب مركزاً لانتشارها.

سادسا: تأثير السبامات (Spams):

الـSpam: هو إرسال الرسائل غير المجدية بكميات كبيرة وهذا غالباً غير مرحب به. هذا النوع من الهجوم هو هجوم أوتوماتيكي من نمط مختلف، لأنها تبدو لمتلقيها رسائل عادية وإن كانت كثيرة، وقد لا يستغرق من المستخدمين وقتاً طويلاً ليبدؤوا بالتعرف على هذه الرسائل أو أغلبها على الأقل. تستغرق المخدمات Servers والتي تتحمل عبء نقل هذه الرسائل وقتاً أطول للتعرف عليها. تجعل الـSpams كلاً من المخدم والمستخدم يعاني من خدمات غير مرحب بها، وقد يؤدي التصفح غير المقصود لها بإعادة إرسالها ثانية من خلال عنوانك الالكتروني مما يجعلها آمنة للطرف الآخر وهي محملة بالفيروسات.

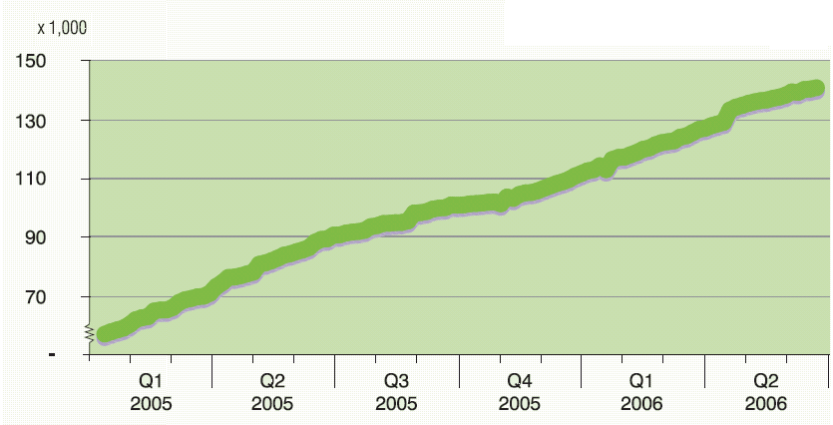
3.2. بعض الإحصائيات والمخططات التي تبين تزايد عمليات الاختراق

نعرض في هذه الفقرة بعض الإحصائيات التي تبين تزايد عمليات الاختراق من العام 1996 و حتى 2007.



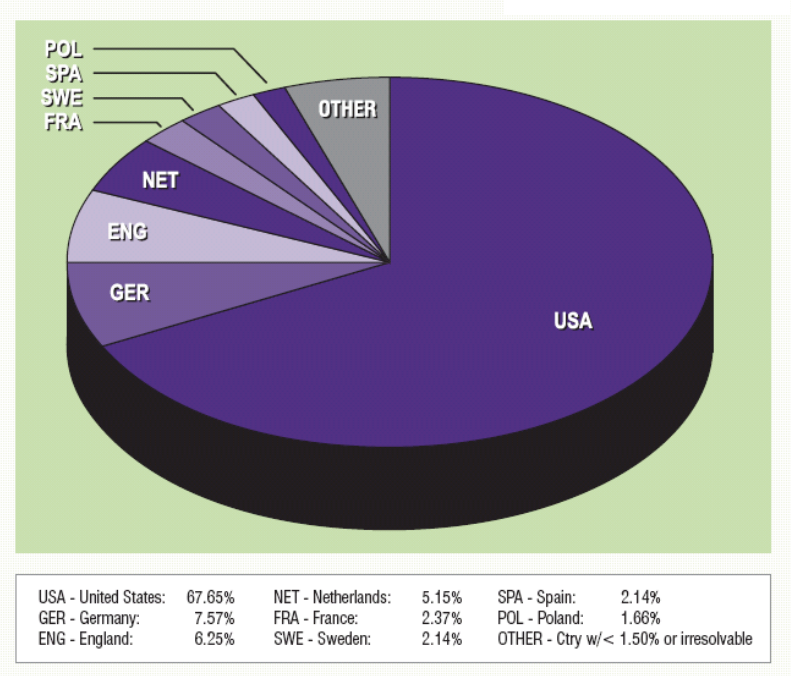
الشكل (1): كمية السبامات Spams الواصلة لمستخدم واحد يومياً ولمعدل شهر من 1996 وحتى 2007 ومن خلال عناوين بريد إلكتروني متعددة.

يبين المخطط السابق معدل السبامات في اليوم الواصلة لمستخدم واحد وعلى مختلف حسابات بريده الالكتروني خلال شهر واحد وذلك منذ عام 1996، ونلاحظ تزايد ملحوظ في عدد السبامات في الأعوام الخمسة الأخيرة. الخط المنقط يشير إلى عدد السبامات الواصلة لحساب مستخدم واحد فقط من حساباته البريدية.



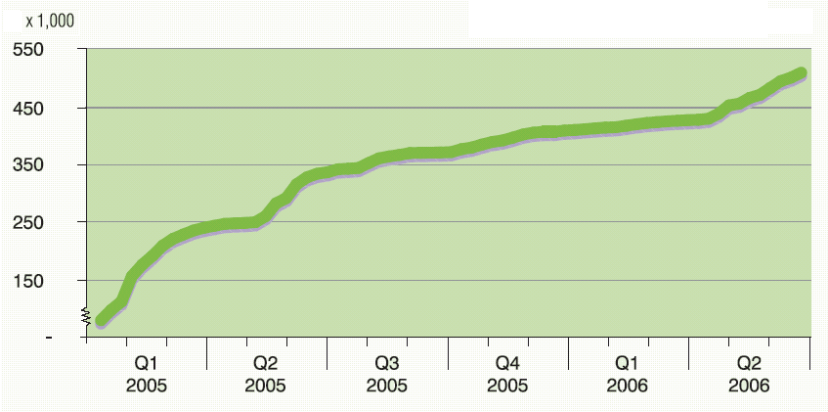
الشكل (2): تزايد عدد برامج الـ Spyware

يبين المخطط السابق تزايد عدد برامج التجسس على الأجهزة والممكن نشرها من خلال البريد الالكتروني وذلك منذ الربع الأول من عام 2005.



الشكل (3): الدول المستضيفة لـ Spyware

يبين المخطط السابق أكثر الدول استضافة للمواقع الناشرة لبرامج الـ Spyware. وكما يبين الشكل فإن الولايات المتحدة الأمريكية أكبر مصدر لبرامج التجسس في العالم.



الشكل (4): تزايد عدد مواقع الـ Malware

يبين المخطط التالي تزايد عدد المواقع الناشرة لبرامج الـ**Malware** وذلك منذ الربع الأول من عام 2005.

3.3. أهم استراتيجيات أمن تطبيقات الويب (Website Security Strategies)

(1) إدارة المخدمات ذات البيانات الهامة بأحدث نسخ من البرامج

الإدارة الصحيحة للمخدمات يجب أن تضمن دائماً جلب الحزم البرمجية الحديثة وآخر التحديثات البرمجية بشكل مباشر لضمان الحفاظ على أمن النظام وقواعد البيانات [4,5].

(2) تهيئة المخدمات بأجهزة وبرامج حائط الصد:

تراقب برامج صد الشبكة (الحماية) نشاط الشبكة ويمكن أن تسمح أو تمنع العمليات على المخدم، وتراقب أي نشاط مريب. ومجموعة الأجهزة ذات المستوى والنوعية العالية وبرامج صد الشبكة يزوّدان بمستوى قوي من الحماية ضدّ أكثر أنواع الهجمات والاختراقات على الشبكة.

(3) استخدام برامج الفيروسات والايميلات المزعجة من جانب المخدمات:

المخدمات يجب أن تعد بطريقة مناسبة لكي تتمكن من تتبّع مصادر الرسائل البريدية الإلكترونية بحيث تحمي مستخدميها من نقل الرسائل المزعجة ورسائل الدعاية. يمكن أن تساعد البرامج المتوفرة على المخدم على تمييز الرسائل المزعجة ورسائل الدعاية وكذلك الأمر بالنسبة لبرامج الحماية من الفيروسات وبالطبع يجب دائما إبقائها على آخر التحديثات. وأيضا هناك الأدوات التي تمنع عناوين أجهزة مرسلي الرسائل المزعجة ورسائل الدعاية، بغية منع بعض الأنواع الأخرى للهجوم (ومثال على ذلك الهجمات من نوع DOS).

(4) إلغاء عملية الاستعراض داخل المخدمات:

العديد من مخدمات الويب تسمح بتصفّح دليل الموقع من خلال HTTP. قد يتمكن لصوص المعلومات من معرفة إعدادات النظام والبرامج المستخدمة على المخدم، ويسمح لهاكر الكومبيوتر باستكشاف أجزاء الموقع غير المتاحة للعموم. يمكن وبسهولة إلغاء تفعيل تصفّح المجلدات وتمكينها وقت الضرورة.

(5) حماية ملفات الإعداد في المخدمات وجعلها ما أمكن ملفات للقراءة فقط:

تستخدم الحزم البرمجية الخاصة بالمخدمات الملفات النصية العادية والتي تتضمن ترتيب الملفات التي استخدمت لتحديد الإعدادات الأساسية لهذه البرامج على المخدم. وتكون هذه الملفات قابلة للقراءة فتعطي للهاكر معلومات هامة جدا تساعده في اختراق الموقع والأسوأ من ذلك أن هذه الملفات النصية قد تكون قابلة للتعديل مما يتيح للهاكر إمكانية تغيير الإعدادات وهنا تقع الأضرار الجسيمه على الموقع. لذلك يجب إخفاء ما أمكن هذه الملفات عن المستخدمين العاديين، وإذا كان وضع الإعداد يحتاج إلى تعديل الملف يعدل مؤقتاً ويعاد مباشرة بعدها للقراءة فقط.

(6) إلغاء تقارير الأخطاء داخل المخدمات والاستعاضة عنها بملفات الأخطاء:

بشكل اعتيادي ، عندما نهيئ PHP على المخدم ، يقوم بعرض جميع مستويات الخطأ ورسائل تحذير إلى المستخدم. وهذا أمر طيب ومفيد عند وضع النظام ، ولكن يمكن أن تحتوى رسائل الخطأ معلومات هامة ومرعبة عن عمل الإدخال والإعدادات على المخدم. فبدلا من إظهار الأخطاء إلى المستخدم، يمكن لـ PHP إعدادها بسجل للأخطاء في ملف على المخدم. وهذا هو أفضل خيار لإنتاج الموقع!. ويجب التأكد من أن الأخطاء مسجلة في الملف بشكل آمن وأن هذا الملف غير مقروء للجميع.

(7) التأكد من السماحيات قبل تشغيل البرامج على المخدمات:

قد تكون ملفات البرمجة النصية الصغيرة على مخدم الويب ضرورية لأداء وظيفة شرعيه، ولكن في كثير من الأحيان ومن الحكمة أن تتأكد من أن المستخدم قد اتبع الطريق الصحيحة من خلال الموقع قبل السماح بإدخال نصه. يوجد بعض الاختبارات لفحص طلبات الـ Script قبل السماح بتنفيذها وذلك للتأكد من أن هذه الطلبات

مشروعة (مثل التحقق من أن المستخدم لديه الترخيص للقيام بذلك). وانه لا يوجد قراصنة تُتدخل طلبات غير شرعية بهدف الإيذاء، وتوفر هذه المصادقة هي خط آخر للدفاع ضد الهجمات الخبيثة.

(8) حذف ملفات التنصيب والملفات الحاوية على تفاصيل هامة:

يؤدي إعداد التطبيقات على المخدمات غالبا إلى ترك مختلف الملفات الواصفة للمصداقية، على سبيل المثال، ملاحظات الإصدار والتي تحتوي جميع التفاصيل التي تم إدخالها على البرامج منذ آخر إصدار أو سجل الملفات التي تورد جميع العمليات التي يؤديها تركيب البرنامج. قد تحتوي هذه الملفات على الأدلة التي ستساعد أي هاكر على تحديد نقاط الضعف في هذه الخدمات. لذلك ينبغي حذفها. ويجب الحرص من جانب مدير المخدم بعدم ترك أي من الملفات السابقة على المخدم لكي لا يراها الأخرون إلا إذا كانت معدة للاستهلاك العام.

(9) التحقق من عمليات الإدخال من صفحات الويب منعاً لدخول كود برمجي تخريبي:

البيانات التي يتم معالجتها على الانترنت وإرسالها إلى المخدم يمكن أن تشكل ضرراً كبيراً إذا كانت خادعة وغير حقيقية. وهذا يظهر بشكل جلي عندما تكون البيانات المستخدمة هي أوامر لغة الاستعلامات البنيوية، أو إذا كانت هناك قيمة تتدخل ليتم عرضها في المتصفح (الرد على المواضيع في المنتديات)، فإذا طُعمت بأوامر لغة الاستعلامات البنيوية فقد يكون هناك خطر كبير. لذلك ينبغي القيام بشكل دوري بفحص جميع مدخلات المستخدم، بغض النظر عما أدخل.

(10) جعل المخدمات يقوم بعملية المعالجة بدلاً من جهاز المستخدم:

الأكثر أمانا لمعالجة البيانات أن يتم ذلك دوماً من قبل المخدم، لأنه لو تركت المعالجة إلى الزبون، لوجد لصوص الكومبيوتر فسحةً للاختراق إذ يمكن أن يعدّلوا بالسماحيات ليتمكنوا من المساهمة غير الصحيحة، حيث يتم تجاوز أيّ معالجة هم لا يرغبون بها. وأحياناً من المفيد في بعض الأوقات تنفيذ سيكربتات من جانب المستخدم والذي يستخدم تقنية مثل javascript للمعلومات غير الحساسة. على سبيل المثال، يمكن أن يستعمل javascript لعرض نوافذ منبثقة popup للتقويم مثلاً، مما يجعل الأمر أكثر سهولة لإدخال التاريخ.

(11) جعل الكوكيز يحوي رقم الجلسة وليس معلومات المستخدم:

الكوكيز مفيد جداً لتطبيقات الويب وذلك للحفاظ على التواصل بين المخدم والمستخدم أثناء عمل تلك التطبيقات ولكن لا يمكن ضمان الحفاظ على تلك الكوكيز لذلك يجب عدم كتابة معلومات هامة داخل الكوكيز ولو كان من الضروري ذلك فيجب أن يكون مشفر. والأفضل من ذلك كتابة رقم الجلسة داخل الكوكيز والتي تمكن المخدم من عملية التواصل مع المستخدم بمطابقة هذا الرقم داخل المخدم مع ذاك الموجود داخل الكوكيز عند المستخدم وأيضاً لو وقع هذا الرقم مع أي لص من لصوص الكمبيوتر فلن يكون هذا الرقم مفيد له وبالتالي لن يستفيد منه.

(12) إلغاء التعقب والاستعاضة عنه بإعادة الكتابة والتعديل كما في المخدم المشترك:

من الميزات القياسية لمخدم الويب الأباتشي مثلاً السماح بتنقيح التطبيقات فوق HTTP. ويمكن أن تستعمل هذه الميزّة بشكل خبيث لمعرفة سماحيات المخدم وحتى الوصول إلى جذر النظام، أي يعطي الحرية الكاملة للهاكر (لصوص الكمبيوتر). من غير الممكن دائماً ضمان إبقاء خيار التعقب TRACE ملغى في الأباتشي، خصوصا عند استخدام مخدم مشترك يُدار من قبل طرف آخر وهو الشركة المضيفة. على أية حال، يمكن أن تلغى الميزّة

عمليا لنطاق محدد domain name باستخدام وظيفة mod_rewrite إعادة الكتابة والتعديل، حيث تعيد توجيه أي شخص يحاول الوصول إلى وظيفة التعقب بشكل غير مؤذٍ.

(13) استخدام المسح الدقيق والأدوات التحليلية الأساسية لتعقب برامج الهاكرز:

هناك أدوات متاحة، البعض منها يكون من مصدر مفتوح، والتي تمكن من مسح لأكثر نقاط الضعف شيوعا في المخدم أو موقع الويب ، ويزودوا بنصيحة تصحيحية. عادة، هذه الأدوات تجدّد بانتظام بآخر الاكتشافات الخاصة بـ ' لصوص القبعة البيضاء ' لذلك يمكن أن يُختبر الموقع بانتظام ويتم إجراء التصحيح اللازم له إذا احتاج لذلك، و' لصوص القبعة السوداء ' يستعملون هذه الأدوات أيضا لتقصّي الضعف في مخدمات الويب وبمعرفة ذلك يمكن تفادي بعض الأخطار.

(14) استخدام تقنية أمن طبقة المقابس (SSL Socket Security Layer) :

تقنية تستعمل لتزويد رزم HTTP بتشفير يعمل فوق الـHTTPS. لكي يتم التشفير يجب الحصول على شهادة ضمان من هيئة شهادات مؤتمنة. هناك مستويات مختلفة من التشفير، الأقوى أن يكون 256 بت. لكن الكثير من المتصفّحات لا تدعم تشفير 256 بت لذلك معظم شهادات طبقة أمن المقبس ستخفّض نفسها إلى 128 بت على مثل هذه المتصفّحات. تشفير الـ 128 بت ما زال يعتبر آمن جداً، ولكن زيادة معالجة الكمبيوتر يمكن أن تساعد على اختراقها.

(15) تشفير كلمات السر باتجاه واحد:

كلّ كلمات السر الموجودة في قواعد البيانات يجب أن تكون مشفّرة بإتجاه واحد، وهذا يعني أنّها غير ممكنة لإعادة فك التشفير. ولكي يتم التحقق من أن المستخدم وصل إلى جزء النظام الخاص به يتم مقارنة كلمة السر المكتوبة عند المستخدم والتي يتم تشفيرها مع كلمة السر المخزنة وعند تطابق الكلمتين يسمح للمستخدم بالوصول إلى جزء النظام الخاص به.

(16) حفظ تاريخ ووقت واسم المستخدم الذي يقوم بعمليات التعديل:

أي عمل يتضمّن التحديث على المعلومات المهمة في قواعد البيانات يجب تتبعه وتسجيله في قاعدة البيانات كتاريخ التعديل ووقت التعديل وطبيعة التعديل، وهوية المستخدم الذي قام بالتحديث.

(17) استخدام الحمايات الإضافية التي تتيحها برامج السيرفرات مثل Apache:

إذا استعملنا مخدم ويب أباتشي، فإن تركيب وحدة تعديل الأمن يمكن أن تزوّد بحماية إضافية ضدّ العديد من الأشكال الهجوم الشائعة فوق الـHTTP كاستعمال المآثر في تطبيقات الويب.

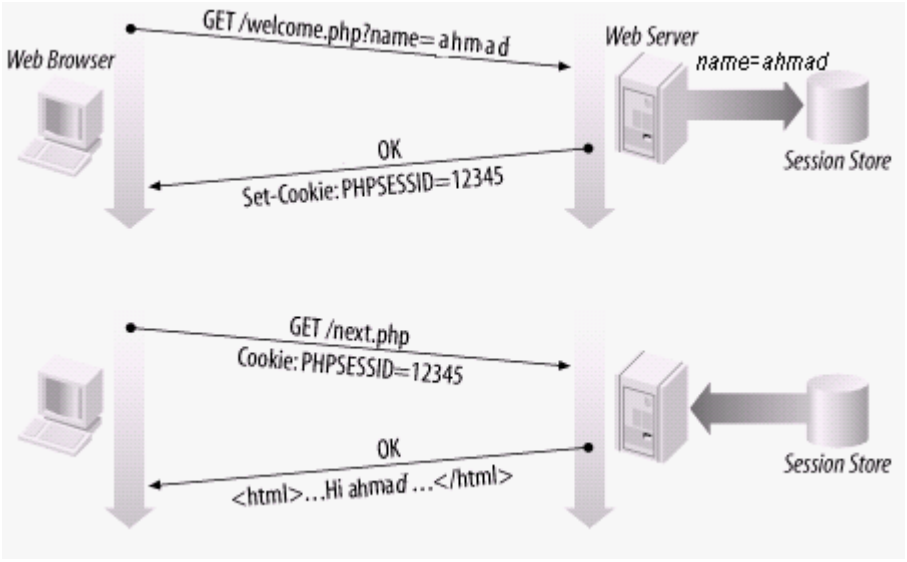
في هذه المقالة، تم التركيز على تقنية الجلسات التي تضمن استمرار التواصل بين المخدم والمستخدم لمنع الاختراقات وقمنا بتطوير هذه التقنية للحد ما أمكن من الاختراقات بمختلف أنواعها وتسجيل كافة الاختراقات في قاعدة بيانات خاصة لمتابعة عمليات الاختراق أو للقيام بالإحصائيات المناسبة والتي تسمح بالوصول إلى مستوى من الأمن أعلى.

3.4. تقنية الجلسات Sessions

عند الانتقال من صفحة الى أخرى في موقع معين فإن بروتوكول الـ HTTP لا يمكنه معرفة أن تلك الصفحات قد تم تصفحها من قبل نفس الشخص. الـ Session هي ملف على المخدم يمكن من خلاله تخزين قيمة معينة للرجوع

اليها في حال قام نفس الشخص بالانتقال من صفحة الى أخرى من خلال الـ cookies المزروع في جهاز ذلك الشخص [10-6].

إذاً التعرف على الشخص الذي يقوم بتصفح الموقع هو الهدف الرئيسي للـ Session.



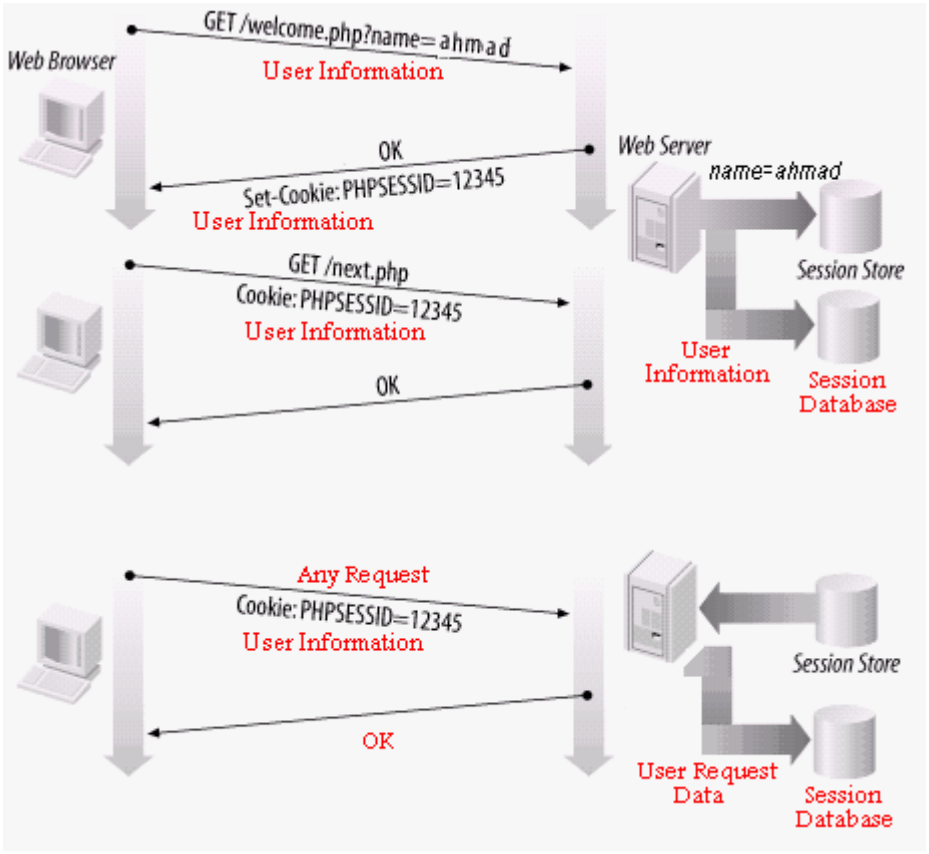
الشكل (5): مخطط يصف طريق عمل الجلسة العادية.

4. النتائج و المناقشات

نقدم في هذه الفقرة تطوير هاما على الجلسات العادية بهدف رفع مستوى الأمن وهي كما يلي:

4.1. تقنية الجلسات Sessions المطورة لرفع مستوى الأمن:

تمكن هذه التقنية مدير الموقع من ضبط أمنه بمراقبة المستخدمين عند قيامهم بالدخول إلى المحتوى أو تعديله أو إدخال أي نوع من البيانات إليه، وكشف أي تلاعب يمكن أن يحدث خلال ذلك. بعد التحقق من اسم المستخدم وكلمة المرور يقوم الموقع بإنشاء جلسة عادية في مخزن الجلسات على المخدم تحتوي على بيانات تتوافق مع البيانات الموجودة في الكوكيز التي قام الموقع بزرعها في جهاز المستخدم. وفي كل طلب يتم مطابقة بيانات الجلسة الموجودة على المخدم مع بيانات الكوكيز الموجودة على جهاز المستخدم لتحقيق هذا الطلب له. وفي الجلسة المطورة يقوم الموقع بالإضافة إلى عمل الجلسة العادية بتخزين معلومات إضافية عن جهاز المستخدم. وعند أية طلب يقوم الموقع بالتحقق من المعلومات الموجودة لديه ومحتويات الكوكيز والجهاز وعندما تتم المطابقة يتم تحقيق الطلب وتسجيل نسخة عن الطلب في قاعدة بيانات مخصصة بذلك ، كما هو مبين في الشكل التالي:



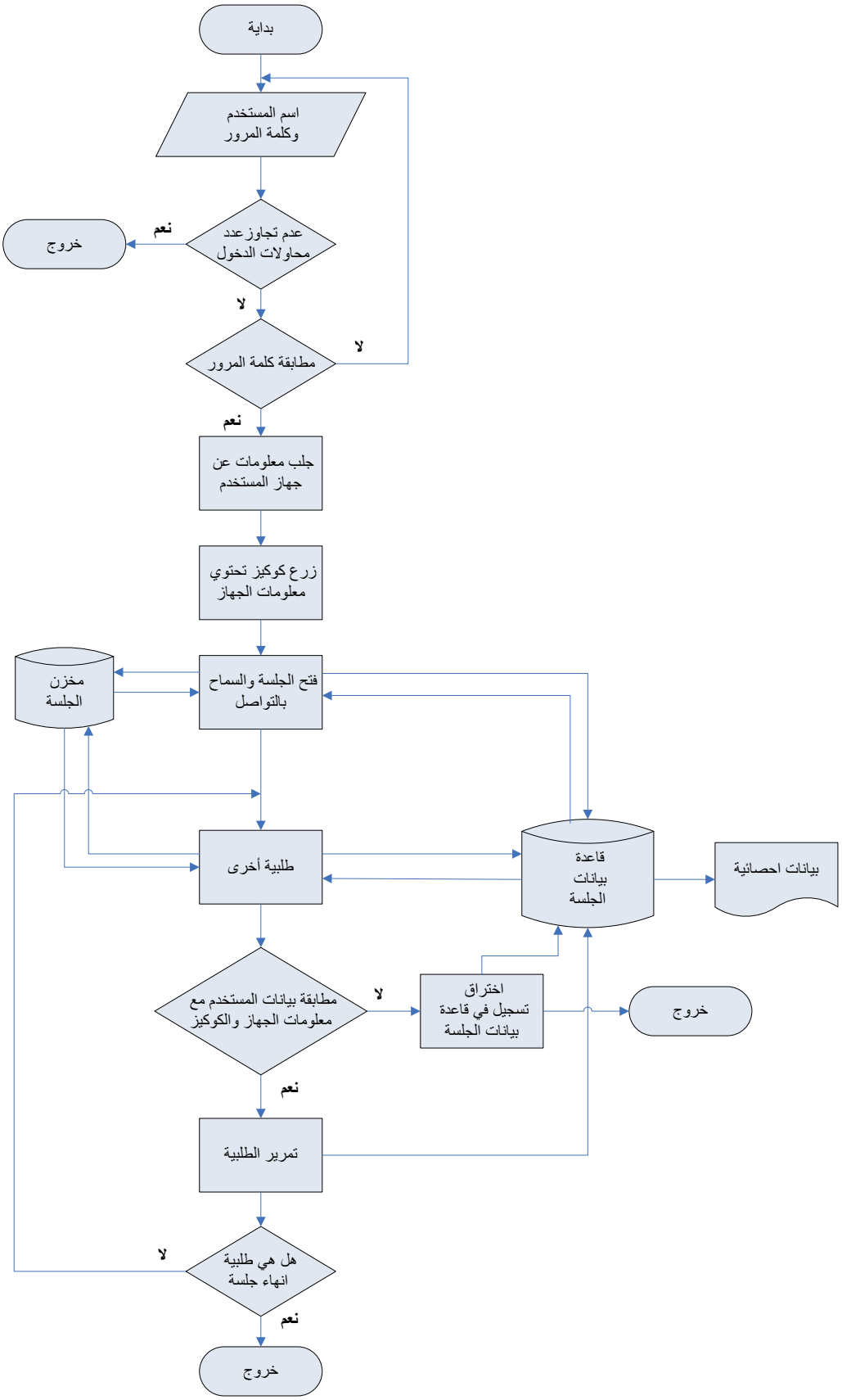
الشكل (6): مخطط يصف طريق عمل الجلسة المطورة في المقالة.

4.2. ميزات تقنية الجلسات المطورة:

نبين فيما يلي ميزات تقنية الجلسات المطورة:

- إمكانية حماية التطبيقات من الـ Griefers , Trools , Pranksters.
- إمكانية حماية التطبيقات من الـ Malware.
- تأمين محتوى التطبيقات من الاختراقات.
- تسجيل جميع تحركات المستخدم على هذا الموقع.
- إمكانية القيام بإحصائيات متنوعة.

في المخطط التالي (الشكل **(6)**) طريقة العمل التفصيلية لآلية الدخول إلى الموقع ومتابعة المستخدم ومراقبة حركاته غير الاعتيادية والتي تسجل في قاعدة البيانات المخصصة لذلك مع إمكانية تسجيل كافة الاختراقات التي تطرأ على هذه الجلسة.



الشكل (7) :مخطط يمثل طريقة الحل باستخدام تقنية الجلسات المطورة في هذه المقالة.

نقدم فيما يلي جزء برمجي خاص بتخزين الجلسة في قواعد البيانات:

```
/* بداية الجلسة */
session_start();
/* صف الجلسة */
class session
{
/* اسم الجدول الخاص بالتخزين */
   var $ses_table = "sessions";
/* متحول التأكد من الاتصال بقواعد البيانات */
   var $db_con = "Y";
/* متحولات التصال بقواعد البيانات */
   var $db_host = "localhost";
   var $db_user = "root";
   var $db_pass = "root";
   var $db_dbase = "web application";


/* إنشاء الاتصال بقواعد البيانات */
   function db_connect() {
      $mysql_connect = @mysql_pconnect ($this->db_host,
                                        $this->db_user,
                                        $this->db_pass);
      $mysql_db = @mysql_select_db ($this->db_dbase);


      if ($db_con) {
         return FALSE;
      } else {
         return TRUE;
      }
   }
/* فتح قاعدة البيانات وبدأ وضع البيانات */
   function _open($path, $name) {
      if ($this->db_con == "Y") {
         $this->db_connect();
      }


      return TRUE;
   }


/* إغلاق الجلسة */
   function _close() {
      $this->_gc(0);
      return TRUE;
```

```
    }
/* قراءة البيانات من قواعد البيانات */
    function _read($ses_id) {
        $session_sql = "SELECT * FROM " . $this->ses_table
                    . " WHERE ses_id = '$ses_id'";
        $session_res = @mysql_query($session_sql);
        if (!$session_res) {
            return '';
        }


        $session_num = @mysql_num_rows ($session_res);
        if ($session_num > 0) {
            $session_row = mysql_fetch_assoc ($session_res);
            $ses_data = $session_row["SES_VALUE"];
            return $ses_data;
        } else {
            return '';
        }
    }
/* كتابة بيانات جديدة في القاعدة */
    function _write($ses_id, $data) {
        $session_sql = "UPDATE " . $this->ses_table
                    . " SET ses_time='" . time()
                    . "', ses_date='" .date('d/m/Y')
                    . "', ses_value='$data' WHERE ses_id='$ses_id'";
        $session_res = @mysql_query ($session_sql);
        if (!$session_res) {
            return FALSE;
        }
        if (mysql_affected_rows ()) {
            return TRUE;
        }
        $session_sql = "INSERT INTO " . $this->ses_table
                    . " (ses_id, ses_time, ses_start, ses_date, ses_value)"
                    . " VALUES ('$ses_id', '" . time()
                    . "', '" . time() . "','" .date('d/m/Y'). "', '$data')";
        $session_res = @mysql_query ($session_sql);
        if (!$session_res) {
            return FALSE;
        }       else {
            return TRUE;
        }
    }
/* حذف بيانات الجلسة */
    function _destroy($ses_id) {
        $session_sql = "DELETE FROM " . $this->ses_table
```

```
                . " WHERE ses_id = '$ses_id'";
        $session_res = @mysql_query ($session_sql);
        if (!$session_res) {
            return FALSE;
        }       else {
            return TRUE;
        }
    }
/* حذف بيانات الجلسة التراكمية */
    function _gc($life) {
        $ses_life = strtotime("-5 minutes");


        $session_sql = "DELETE FROM " . $this->ses_table
                . " WHERE ses_time < $ses_life";
        $session_res = @mysql_query ($session_sql);
        if (!$session_res) {
            return FALSE;
        }       else {
            return TRUE;
        }
    }
}
```

5. الاستنتاجات والتوصيات

عرضنا في هذه المقالة أهم التهديدات الأمنية و الاستراتيجيات الأمنية التي قد يواجهها أي تطبيق يعمل عن طريق الشبكة إذ قدمنا طريقة مطورة لتقنية الجلسات العادية للحد من اختراقات المواقع. بينا ميزات التقنية المطورة التي تبشر كما هو واضح بنتائج جيدة تساهم في الحد من التهديدات التي يتعرض لها مستخدموا شبكة الانترنت.

المراجع

[1] SNYDER C and SOUTHWELL M. *Pro PHP Security*, 1st edition, A Press, USA, 2005, 500.

[2] MICHELLE M. *A Hacker's Guide to Protecting Your Internet Site and Network*, 1st edition, Ange l722 Computer Publishing, USA , 2003, 670.

[3] CASTAGNETTO J , RAWAT H, SCHUMANN S, SCOLLO C, VELIATH D. *PROFESSIONAL PHP Programming*, 1st edition, Wrox Press Ltd, USA, 1999, 1102.

[4] GREENSP J and BULGER B. *MySQL/PHP Database Applications*. 1st edition, IDG Books, USA , 2001, 622.

[5] WILLIAMS H and LANE D. *Web Database Applications with PHP, and MySQL*. 1st edition, O'reilly, USA, 2002, 582.

[6] 21-6-2007 <http://www.webroot.com>.

[7] 14-8-2007 <http://www.netshinesoftware.com>.

[8] *PHP Security Guide*. 12-9-2007 <http://phpsec.org>.

[9] Fuecks H. *Notes on PHP Session Security* 20-10-2007 <http://www.sitepoint.com>.

[10] *The practical solution of requirements using PHP* 4-7-2007 <http://www.wellho.net>.