Setting up IIS and securing streamed content (from Sam)

I have to admit that nowadays, the live of a security adviser is very tuff.
I have come across many misleading articles in Internet, which have brought me to write this article.

Sorry, but I do not intend to teach you here any thing about html, javascript, php or ASP.
You have to have enough knowledge about IIS, Apache and PHP or ASP to understand this article!
For better understanding, I recommend to see the following pages to have some idea about of what I am talking here!

1) http://motion4u.sytes.net
2) http://www.moviefix.com

How to prevent downloads but still letting visitors only from your own site to read a file on your server or serving a steamed film without letting others to download them.

Introduction

If you have looked to secure your files on your server, you have probably found dozens of options like ISAPI-filters for IIS or other locking tools, which eventually 99% of them do not work or they do some thing but not what you really wanted. Eventually, after few tries, you will reinstall the bloody server once more.

In this chapter at the beginning, I will explain about using PHP and ASP and later concentrate only on using PHP.
On the next chapter, I will enhance this method for ASP.

About IIS

IIS 5 has been programmed so dry that it will not let you in anyway to tweak URS or hiding them in anyway!
(I hope that Microsoft will not sue me for this ☺)

Lets assume that you have a hidden subdirectory containing “nice pictures”, films or documents which, you want to let your visitors to view but not to download or leech them!

Now, the problem is that on the page where you will place the HIDDEN LINK to whatever you want to show, could always be seen through the source code.
You may already have tried the below options:

1) To prevent the user to read the source code, you may have tried to install a no-right-mouse javascript (but for example, it doesn’t work with Netscape!)

2) Open the download page in frames (This also does not work with Netscape)

3) Locking the access page only for Authorized users with an ISAPI filter
(but it is cumbersome because, you have to release access for each individual visitor)

The Only way to hide a URL is to sending it by reference to your page like this for example:

Programming

This is a normal HTML link:
```
<a href="http://www.motion4u.sytes.net/software/">download</a>
```

This is a Hidden Link with PHP:
```
<a href="hidden_url.php?id=<?=$id?>">download</a>
```

The content of the hidden_url.php file may look like this:
```
<?
$filename=”suckmeup.zip”;
$link       ="http://localhost/".$downloadme;
echo $link;
?>
```

This is a Hidden Link with ASP:

```
<a href="hidden_url.asp?id=<%=$id%>">download</a>
```

The content of the hidden_url.asp file may look like this:

```
<%
filename=”downloadme.zip”
response.write("http://localhost " &filename)
%>
```

What did we learn out of this?

If you will write your links to load a URL out of another file, the URL will not be visible in the source code!
So, if by now you think that your problem is solved, you are wrong!

Why?
Because, anyone with basic knowledge would open the source code and see something like this for example:
download

So, he will type in the following URL to get the content of your hidden file!

with PHP:
http://serveraddress/ hidden_url.php?id=1
or with ASP:
http://serveraddress/ hidden_url.asp?id=1

What did we achieve until now?

The URL is not visible in the source code, but the address of the hidden file is still visible!

The idea and the Solution

Now, if we could program our server to differ and return the URL only to our own server back and not to anybody else who would type the above URLs, we would be save!

How?
By using session Variables!

From now on, I will explain only about PHP to make things easy to understand.
In the next chapter, I will enhance this for ASP too.
Do we agree?
Sure, you do not have any other choice ☺
Anyway, where was I? ….. O yeh,

To prepare sessions in PHP you have to insert this code on the top of each php document:

```
<?
session_start();
?>
```

For ASP: there is no need for any extra code

This is how you start a session in PHP:

```
<?
//Start the bloody session
 session_start();

 // Register a session named SESSION

   session_register('SESSION');

// Now push a variable named 'its_me' with a value of 0 !

   if(!isset($SESSION))
    {
     $SESSION = array();
      // now save variables into this session
      // for example

     $SESSION['its_me']    = "0";

     // $SESSION['someother'] = "0";
     }
?>
```

If you cannot guess why the above code is so weird, is because we do not want to initiate the session more than once, even if we will revisit the page and for that, there is an commented example, how to add more than one variable into a session.

If I have made you too dizzy, we could also diminish the above code to something like this:

```
session_start();
session_register('SESSION');

if(!isset($SESSION))
{
 $SESSION = 0;
 }
```

Are you satisfied now? Ok, so lets march on

The variable: *$SESSION['its_me']* (or for the diminished example *$SESSION*) exists and carries the initial value of 0!

So, if you are lost now, it is time to put all peaces together!

Summary

We have a file with the name of *hidden_url.php*.
This file returns a hidden URL to a link and this link can open for example a site with a picture!
Now, we wanted that only our server would be able to open that file and in order to do so,
we intend to post a hidden session variable to the page which will return the hidden URL and immediately after that, alter this session variable.

Or in a more simple way of explanation:
You have the key to open the stable (it is the session variable)
So, open the stable, pull the stinky cow (your hidden URL) out and close it as fast as possible to prevent the flies from going into the stable! Got it? ☺

The practical solution

Now we want to have a link that, when you click on it, will do the following tasks:

1) Set a Session variable to some value (lets say 1) to allow our hidden file to send the URL to our new page (open the stable door).
2) Open a new window which, will call within it a hidden URL from our hidden file

Hmmm…., a bit complicated. I know!
Maybe make a short brake and come later ☺
Are you with me now? Ok,

Note!
All the following javascripts will work correctly if you will insert them in between the <head> and </head> tags!

The below javascript-function opens a new browser window (800x600 pixels) and posts a single variable (id) to the opened file, not more and not less.
If you have already looked at the URL, you have understood that, the file *hidden_url.php* resides in the script directory.
It means, can be executed but never downloaded or opened!

```
<script Language="Javascript">
function Win (id)
{
window.open("/scripts/hidden_url.php?
id="+id","window","toolbar=no,width=800,height=600,directories=no,status=no,scrollbars=auto,resizable=no,menubar=no");
}
</script>
```

If you wonder how this window looks like, you could go to any of the below links and find it out ☺
1) http://motion4u.sytes.net
2) http://www.moviefix.com

Now, lets write another javascript function named open_win().
This function fills the following variables in the below form and submit it back to the same page!

```
function open_win ()
{
document.Form.session_val.value=1;
document.Form.id.value=1;
document.Form.submit();
return true;
}
```

Here is the form which will be also somewhere at the end of your php page

```
<form name="Form" method="post" action="">
<input type="hidden" name="session_val">
<input type="hidden" name="id">
</form>
```

And the last modification will be to add an “onload” function in the body tag of the page wrapped with php code
This is a hybrid of PHP, Javascript and HTML (We have salad now ☺)

```
<BODY   onLoad="<?If(isset($session_val)&&$ session_val >0){?> Win (<?=$id?>)<?}?>">
```

The Idea here is when the variable *($session_val)* exists and has a value of 1 the body tag will look like this:
```
<BODY   onLoad=" Win (1)">
```

This will also, automatically call our Win() function which, will open a popup window of 800x600 pixels.

Else, the body tag will look like this:
```
<BODY   onLoad="">
```

And who closes the stable? Yes! The *hidden_url.php* file will close it for us!

If you ask yourself, how the hell, do I call this function in my HTML document?
Like this:
```
<a href="javascript: open_win()">Sesame open</a>
```

Now, It is the right time to see how our *hidden_url.php* file looks like:

```
<?
session_start();
if (isset($SESSION)&&$SESSION  == "1") // is my stable open?
{
//pull the stinky cow out (out put the hidden URL)
$link       ="http://localhost/my_hidden_dir/my_hidden_file.rm";
// Pass it on
echo $link;
//close the stable
 $SESSION  = "0";
}
?>
```

Puhhhhh, I don’t know how you are doing but, I am finished ☺

Don’t worry, I will clear all and put all together for you.

In order to test these functions open a new blank file, paste the below code into it and save it as *hidden_url.php*

```
<?
session_start();
if (isset($SESSION)&&$SESSION  == "1")
{
// Output the hidden URL
$link       ="http://localhost/my_hidden_dir/my_hidden_file.rm";
// Pass it on
echo $link;
//close any access
 $SESSION  = "0";
}
?>
```

Open a second file, paste the below code into it and save it as *test.php*

```
<?
session_start();
session_register('SESSION');

if(!isset($SESSION))
{
 $SESSION = 0;
}
if(isset($session_val))
{
$SESSION = $session_val;
}
```

```
?>
<html>
<head>
<title></title>

<script Language="Javascript">

function Win (id)
{
window.open("hidden_url.php?
id="+id,"window","toolbar=no,width=800,height=600,directories=no,status=no,scrollbars=auto,resizable=no,menubar=no");
}

function open_win()
{
document.Form.session_val.value=1;
document.Form.id.value=1;
document.Form.submit();
return true;
}
</script>

</head>

<BODY  onLoad="<?If(isset($session_val)&& $session_val >0){?> Win (<?=$id?>)<?}?>">

<a href="javascript: open_win()">Sesame open</a>

<form name="Form" method="post" action="">
<input type="hidden" name="session_val">
<input type="hidden" name="id">
</form>

</body>
</html>
```

Lets see what we have achieved here:
If you would type the following URL in your browser

http://your_domain/hidden_url.php?id=1

You would get a blank page!

Nevertheless, the same URL called through the test.php will reveal its content.

What you have learned until here is only the top of the iceberg I would say.
There are still lots of holes to be sealed, but those are the basics.
Stay tuned since, in the future chapters; I will enhance this issue for different situations and needs.

Problems
This type of security has its own disadvantages too.

If you are serving Realmedia type of files, make sure that the streamed files do exist. For example,
Realmedia player programmed so wisely that it would present an error popup-window showing the full non-existing URL that, you have worked so hard to hide. (With friends like this, who needs enemies ☺)

Few last words

The next articles would be about:

1) How to implement the above security issue with ASP
2) Security issues with Apache server such as URL tweaking.

Sam

Any comments, ideas or questions, Please send to sambukkaa@hotmail.com