Essential ASP for Web Professionals by Elijah Lovejoy, © 2001 Prentice Hall PTR, ISBN 0-13-030499-9, http://www.phptr.com/essential/asp/.

Advanced Projects

Below are some more specifics on some of the ways the scripts presented in the book can be hacked, based on the section in each chapter entitled “Ideas for Hacking This Script.”

Chapter 1

Modified Arrays

Perhaps the easiest way to hack the "fancy" date script in this chapter is to put new values in the arrays.

Pour example, en francais:

```
1.  <%@ Language=JavaScript %>
2.
3.  <HTML><HEAD><TITLE>Fancy Date</TITLE></HEAD>
4.  <BODY BGCOLOR="#FFFFFF" >
5.
6.  <%
7.  var now, days, months, day_of_week_int, day_of_month_int, month_int, year, today_date;
8.  now = new Date();
9.
10. days = new Array ("dimanche","lundi","mardi","mercredi","jeudi","vendredi","samedi");
11. months = new Array
    ("janvier","fevrier","mars","avril","mai","juin","juillet","aout","septembre","octobre
    ","novembre","decembre");
12. day_of_week_int  = now.getDay();
13. day_of_month_int = now.getDate();
14. month_int = now.getMonth();
15. year = now.getFullYear();
16.
17. today_date = days[day_of_week_int] +", "+ months[month_int] + " " + day_of_month_int +
    ", " + year;
18.
19. Response.Write (today_date);
20. %>
21.
22. </BODY></HTML>
```

Il faut dire que je n'ais pas fait d'etude en francais depuis la septieme, et meme a ce temps la, mon orthographe ete completement nul. Whatever.

More Detail

More detail can be added by inserting code to display the current hour and minute (bold):

```
days = new Array
("Sunday","Monday","Tuesday","Wednesday","Thursday","Friday","Saturday");
months = new Array
("January","February","March","April","May","June","July","August","September","Oc
tober","November","December");
day_of_week_int  = now.getDay();
day_of_month_int = now.getDate();
```

```
month_int = now.getMonth();
year = now.getFullYear();
hour = now.getHours()
minutes = now.getMinutes()
today_date = hour + ":" + minutes + " - " + days[day_of_week_int] +", "+
months[month_int] + " " + day_of_month_int + ", " + year;
```

Chapter 2

Echoing Guest Book Contents to Visitors

Modifying the guest book to display its contents after a submission is simply a matter of redirecting a user to the script with the `to_do` variable set to `print_contents` once the submission has been written to the file system. Since by this point html has already been sent to the browser, the code below does this using client-side JavaScript:

```
Response.Redirect ("<script>location=\"" + var this_script_url +
"?to_do=print_contents\"</script>")
```

Adding Email

Adding email to the guest book is also something that can be done with a small amount of code. For example, the code below adds email after someone has submitted an entry to the guest book, which includes a form variable called `Email`:

```
if (String(Request("Email")).search(/@/) > 0){
        mail_ob = Server.CreateObject("CDONTS.NewMail");
        mail_ob.Send("guestbook@lovejoy.com",
        String(Request("Email")),"Thanks","Thanks for contributing to the
        Lovejoy.com guestbook");
}
```

The regular expression is used to make sure that the `Email` field contains, if not an entire email address, at least the `@` symbol that seems to pop up in so many email addresses.

Chapter 3

Adding Keyword Search

The first thing you need is a form that lets users enter a keyword. For example:

```
<form action = ch3_list_records.asp>
Keyword Search: <input type=text name=keyword_search value="">
</form>
```

You can put this form wherever it makes sense for you to do so on your site. However, do not try to put the code inside an existing form, since that's a pretty effective way to confuse browsers.

The next step is to check for the `keyword_search` variable and use the appropriate SQL string depending on whether or not a keyword search is being performed (which can be determined based on whether or not the `keyword_search` variable is set). For example:

```
if (String(Request(keyword_search)) != "undefined"){
    // sql for keyword search
    sql = "SELECT * FROM data where data_category = " + current_category + " order
by data_subcategory, data_name";

}
else{
```

```
    // normal sql
    sql = “select * from data where data_category = “ + current_category + “ and
where data_name like ‘%” + search_term + “ %’”;
}
```

Chapter 4

Fancy SQL

It makes a lot of sense not to delete a category record if there are any records left in that category. And in fact, it's the kind of thing that SQL is pretty good for. For example, the following SQL will do the trick.

```
1.  delete_sql = "delete from categories where category_uid = " + current_category AND
    (Select count(*) from data where data_category = " + current_category + ") = 0;
```

Basically, the way this works is that the `Select count(*)` statement will return zero only if there are no records left in the category. If any records are found, the `count()` function (this is a SQL function) will return an integer greater than zero, and nothing will be deleted.

SQL does lots of other neat stuff too. But that's another book.

Chapter 5

Make Sure Prices Were Not Hacked

As written in the book, the shopping cart has a security vulnerability: a malicious shopper could build an html page and set the price of an item to less than it should be. The code below fixes this. You could put this code, for example, in the checkout process, perhaps after collecting shipping information:

```
1.  var abs_path =  String(Request.ServerVariables("PATH_TRANSLATED"));
2.  file_to_open = abs_path.replace(/\\\w*\.asp/,"\\") + option_file; fso = new
    ActiveXObject("Scripting.FileSystemObject");
3.      fs_stream = fso.OpenTextFile(file_to_open);
4.      while (! fs_stream.AtEndOfStream){
5.              temp = fs_stream.ReadLine() ;
6.              // if there are two pipes next to each other, add a space.
7.              temp = temp.replace(/\|\|/g,"| |");
8.              temp_array = temp.split(/\|/);
9.              if (temp_array[0] != "//id" && String(items[temp_array[0]]) !=
    "undefined"){
10.                     temp_item_id = items[temp_array[0] + temp_array[1]
11.                     for (key in cart_items){
12.                             if (cart_items[key].uid == temp_item_id){
13.                                     cart_items[key].uid = temp_array[2]
14.                             }
15.                     }
16.             }
17.     } // end while
```

This is a "stealthy" approach: if prices were changed, it changes them back, and whoever tried to crack your cart just wasted their time. I suppose you could modify it to say "ha, I caught you" by comparing the price in `cart_items` to the price retrieved from the option file. But being confrontational with this kind of person is probably not a good idea. Bad enough that there are people like that out there. Let them focus their attention elsewhere.

Chapter 6

Collect More Data

The code below collects all the data that it can find and sends it to you:

```
1.  temp_body = ""
2.  for ( field_number = 1; field_number <= Request.QueryString.count ; field_number++ )
3.      {
4.      temp_field = String(Request.QueryString.Key(field_number));
5.      temp_content = String(Request.QueryString.item(temp_field))
6.
7.      // make sure field is one we want to use
8.      if (temp_field.search(/toss/) < 0 ){
9.              temp_body += temp_field + ": " + temp_content + "\n\n"
10.     }
11.
12. mail_ob = Server.CreateObject("CDONTS.NewMail");
13. mail_ob.Send("form@lovejoy.com","recipient@lovejoy.com","form information",temp_body);
```

The code above works by looping through incoming form information, finding all the form fields, and concatenating them into a variable called `temp_body`. The only exception occurs when the variable is called `toss`, which I like to use for submit buttons whose value I don't want recorded.

That done, `temp_body` goes into an email which is sent to `recipient@lovejoy.com`. Please don't use this code without changing that email address – my junk mail filter is overworked already.

Chapter 7

Have a Nice Day

Turn your computer off. If you have a cell phone, take it out of your pocket and leave it at home. If it's nice out, consider going for a walk. If it's raining, you might go to a museum or a library or a coffee house. Visit a friend, or stay at home. If you see a baby, give it a big smile. If you run into someone you like, wish them well.

You don't have to do this for a whole day. You can do it for a few minutes in the morning or during any part of the day really. You can even do this a little bit every day.

And even though it has nothing to do with writing code, if you take breaks regularly, you will probably end up writing better code. Go figure.