HACKING ENCRYPTED HTML

By Edward Stoever
http://www.database-expert.com

This article will show you a hack you can use to decrypt the html that has been encrypted by three popular software programs. There are a number of reasons why a webmaster would want to encrypt his or her html markup. The most obvious reason is to protect the markup from your curious eyes or to prevent you from directly downloading images or flash movies from a website. There are even better reasons for using encrypted html that don't involve encrypting the entire page. I will explain these shortly.

In this article, I am going to show you how to decrypt nearly any html that is encrypted with javascript. Then, at the end of this article, I will show you a couple of websites that create encrypted html for free. In the event that you ever need to use encrypted html, you will know its strengths and its inherent weakness, and you will know where on the web to get it done gratis!

For our first challenge, let's open this website with a browser: http://www.protware.com. To see protware in action, click the "Demonstration" link. In the center of the next page is a link "Click here to open the encrypted demo page." Click that and a new window opens. In this new window, right-clicking has been disabled. So, view the source through the menu: view, page source. Now you see a huge javascript but no recognizable html. We want a source that we can read and understand.

To hack this source, you will need to edit the html markup. You can use any html editor, even notepad or gedit. Select the entire page source, copy it then paste it into your editor. At the top of the page, immediately after the `<script>` opening tag, type in:

```
document.write('<textarea cols="80" rows="40" name="whatever">');
```

Then, at the bottom of the page, immediately after the `</script>` closing tag, type in:

```
</textarea>
```

Now, save the page, and open it in a browser or click the browse tab in your html editor if you have one. You will see a big text area and inside of it, you will see the html that you can recognize. All the paths and filenames of the images and other objects are readable. If you want to view that in a browser, select that text and save it. The javascript at the top will prevent you from viewing it. Just cut it out of the text and the page will view just fine. That was just too easy!

From this point on, I will refer to this technique as "wrapping" because we are surrounding the javascript output in a textarea. As we move to the next examples, you will see javascripts that encrypt javascripts. You will be able to wrap any of these in a textarea to see the underlying decrypted code.

The following examples use javascripts that open like this:

```
<SCRIPT LANGUAGE="JavaScript"><!--
```

The html comment tag `<!--` requires that we add a carriage return before we open the wrapping with

```
document.write('<textarea cols="80" rows="40" name="whatever">');
```

Our next challenge can be found here: http://www.antssoft.com/htmlprotector. In the middle of that page is a link: "Click here to view sample page protected by HTML Protector." In the sample page that opens, right-clicking has been disabled so view the source from the menu. Copy and paste the html text into your editor. Here you will see three javascripts. If you wrap the first one by itself, you will find that it hides another javascript. You have the option of replacing the encrypted javascript in your editor with the decrypted one in

your browser. If you don't replace the encrypted version with the decrypted version, remove the wrapping so it will function. You can decrypt the source simply by wrapping the final javascript.

Finally, we get to our third challenge: http://www.aevita.com/web/lock/samples.htm. This website has taken some extra steps to make their content harder to decode. Click the link for STRONG encryption scheme. A new page will open that will look just like the Google homepage. View the source through the browser's menu. At first, the source looks like it is empty, but that is just because of a bunch of added carriage returns. Scroll down! Copy the source and paste it into an editor.

The source has four javascripts. If we use our wrapping hack on the first one, we find that it is a javascript that just disables mouse clicking. Simply delete that first javascript. Next, look for a javascript that includes the text `src="encrypt.js"`. Here is the tricky part. We need that bit of code to complete our job. Go back to the browser, and change the URL for the page to this:
http://www.aevita.com/web/lock/samples/encrypt.js.

The text we need either appears in the browser or can be saved as a text file depending on the browser you use. Copy all of the text from encrypt.js and paste into the text editor between the script open and close tags as shown here:

```
<SCRIPT src="encrypt.js" type="text/javascript">paste it here!</SCRIPT>
```

Next, delete the text `src="encrypt.js"` out of the script open tag. Then, on the last javascript on that page do a wrapping hack. Now view the page in a browser and you will see the html source you wanted to see.

The wrapping technique shown above can be used on nearly any javascript html encryption to view the true html markup.

There are a couple of reasons I use encrypted html, neither of which is to prevent people from reading the source of the page. In either of these cases, I only encrypt the small portion of the html markup that I want to hide.

The first reason I use encryption is to hide email addresses from spambots, programs that search the internet hunting for email addresses to send spam to.

The second reason I use encryption is to hide `<DIV>` tags that I use to layer divisions in web pages. I use the `<DIV>` tags to conceal text from the user's eyes and at the same time make the text available to search engines. Search engines know we can use `<DIV>` tags to do this, and can be programmed to eliminate text strings found in divisions that are not visible to people. By encrypting the `<DIV>` tags, a search engine will have a harder time eliminating the concealed text from its search index. For an example of hiding `<DIV>` tags, visit my homepage: http://www.database-expert.com.

My personal favorite way of encrypting text strings can be found here:
http://www.guymal.com/nospam_email_link.php
Guymal's utility is easy to use, quick and free.

Another package for encrypting html markup for free can be found here:
http://javascript.about.com/library/blenc.htm

Happy decrypting!
-Edward