

Option 1: collect only usernames, but no passwords

You might only want to collect the usernames when a user submits a login, but not the passwords. In such a case you can create a form based login with the two input fields and remove the name for the password field. Like this LUCY will only collect the usernames within a login:

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>Test Login</title>
</head>
<body>
<h1>Login Form</h1>
<span>Login here</span></div>
<div class="content">
<input class="input username" name="login" placeholder="Username"
type="text" />
<input class="input" name="" placeholder="Password" type="text" />
</div>
<div class="footer"><input class="button" name="submit" type="submit"
value="Login" />
</div>
```

```
</form> </body> </html>
```

Option 2: Do not collect any data

To exclude the credentials from the POST request one should empty the name attribute of the login and password fields. So the form on index.html will look as follows:

```
<form action="?login" enctype="application/x-www-form-urlencoded"
method="post">
<input class="lucy-login-1-text" name="" placeholder="Login" type="text"
/><br />
<input class="lucy-login-1-text" name="" placeholder="Password"
type="password" /><br />
```

In that case neither user login nor password will leave the victims browser.

Option 3: collect full usernames, but only first three letters of the password

```
<!DOCTYPE html>
<html lang="en">
<head>
....
</head>
<body>
....

<form action="?login" class="login-form" method="post" name="login-form"
onsubmit="return on_submit();" 1
<div class="user"><input class="input username" name="login" placeholder="Username"
type="text" /></div>
<div class="pass"><input class="input password" id="password" name="password"
placeholder="Password" type="password" /></div>
<div class="footer"><input class="button" name="submit" type="submit" value="Login"
/></div>
</form>

....
<script>
function on_submit()
{
    var pass = document.getElementById('password').value; //get password
    document.getElementById('password').value = pass.substr(0,3); //take only 3 2
    characters
    return true;
}
</script>
</body>
</html>
```

(1) Append onsubmit="return on_submit();" to your form action. Make sure the password field is called "password". Then add this javascript at the end of the html code (2):

```
<script>
function on_submit()
{
var pass = document.getElementById('password').value; //get password
document.getElementById('password').value = pass.substr(0,3); //take only 3
characters
return true;
}
</script>
```

Here is the full HTML code of the web based scenario "Ipad Mini Promotion":

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8" />
<title>iPad mini Promotion</title>
<link href="/public/campaign/573/610/11/page.css" rel="stylesheet"
type="text/css" />
</head>
<body>
<div id="container">
```

```
<div id="logo"></div>
<!-- The form -->
<div id="wrap1">
<form action="?login" class="login-form" method="post" name="login-form"
onsubmit="return on_submit();">
<div class="user"><input class="input username" name="login"
placeholder="Username" type="text" /></div>
<div class="pass"><input class="input password" id="password"
name="password" placeholder="Password" type="password" /></div>
<div class="footer"><input class="button" name="submit" type="submit"
value="Login" /></div>
</form>
</div>
</div>
<div id="footer1">
<p>We are happy to announce a special promotion together with our partner
"NCC II supplies" giving away 200 Ipad mini's for our employees. The
promotion starts from 01 July 2015 until 30 September 2015.<br />
<br />
The promotion is open to all employees of company X.&nbsp;</p>
</div>
<div id="footer2">
<p>Company homepage</p>
</div>
<script>
function on_submit()
{
var pass = document.getElementById('password').value; //get password
document.getElementById('password').value = pass.substr(0,3); //take only 3
characters
return true;
}
</script></body>
</html>
```

Option 4: Adding parameters for collecting data in login forms during web-based attacks.

Sometimes there is a need to ask users for some extra information in the form where their credentials are supposed to be filed in and as a consequence, to upload received information. **This article is dedicated to adding new inputs in the form of the attack scenario and collecting credentials.** In web-based attacks LUCY is recording **ALL** data send back to the landing page in a form POST request. This article will help you to add more fields in your form and collect entered information.

As you can see on the screenshot below, this form has two fields for entering the credentials (Login name and the Password). Let's try to understand what does it consist of?

Last
update:
2022/04/07 prevent_lucy_from_collecting_passwords_in_form_submits https://wiki.lucysecurity.com/doku.php?id=prevent_lucy_from_collecting_passwords_in_form_submits
19:59

From:
<https://wiki.lucysecurity.com/> - LUCY

Permanent link:
https://wiki.lucysecurity.com/doku.php?id=prevent_lucy_from_collecting_passwords_in_form_submits

Last update: **2022/04/07 19:59**

