**bWAPP**

**Broken Authentication**

Tổng quan về Broken Authentication:

**- Broken** **Authentication là gì?**

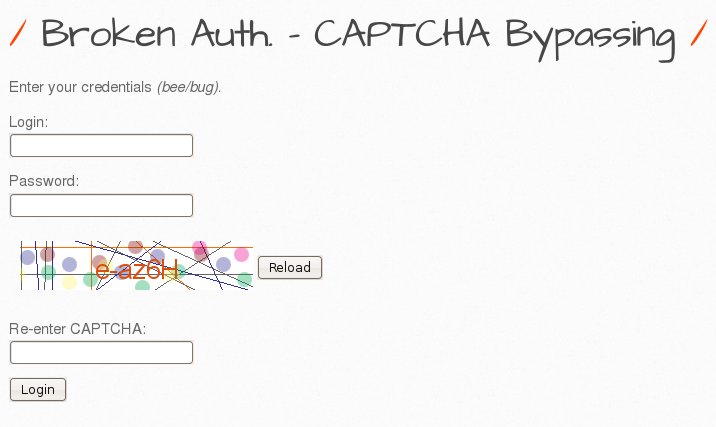
Broken Authentication là một thuật ngữ dùng chung cho một số lỗ hổng mà kẻ tấn công khai thác để mạo danh người dùng hợp pháp. Broken Authentication đề cập đến điểm yếu trong các lĩnh vực “quản lý phiên” và “quản lý thông tin xác thực”

Những kẻ tấn công có thể tận dụng nhiều cách để khai thác lỗ hổng này, và trong trang web bWAPP có 6 bài liên quan đến Broken Authentication đó là: “CAPTCHA Bypassing”, “Forgotten Function”, “Insecure Login Forms”, “Logout Management”, “Password Attacks”, “Weak Passwords”

Và ở bài viết này chúng ta sẽ đi qua toàn bộ các bài liên quan đến Broken Authetication trong trang web bWAPP

**Broken Auth. - CAPTCHA Bypassing**

Đây là giao diện của trang web



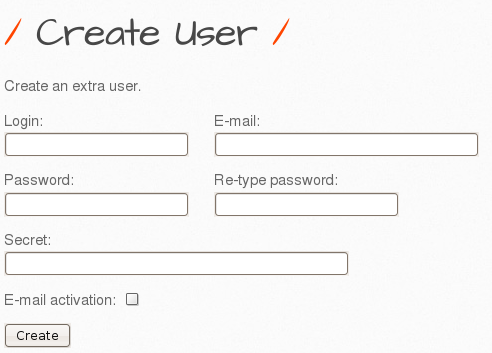
Tôi đã thử nhiều cách để có thể loại bỏ Captcha mà dường như trang web này không có lỗ hổng để ta có thể bypassing qua Captcha của nó. Tất cả kết quả thu được chỉ là tên users là bee và password của nó là bug nhưng những thứ này đều đã hiển thị ở trên giao diện của trang web nên có vể như trang web này không có lỗ hổng liên quan đến captcha bypassing

**Broken Auth. - Forgotten Function**

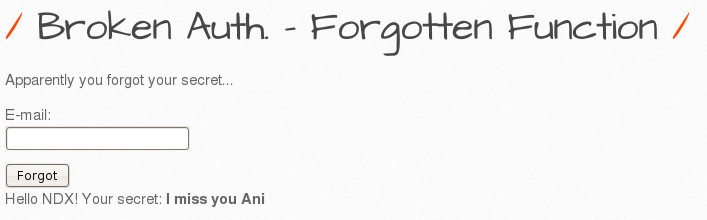
Đây là giao diện của trang web

****

Ở đây trang web bắt chúng ta nhập một e-mail vào vậy chúng ta sẽ tạo một tài khoản mới



Sau khi tạo tài khoản xong chúng ta sẽ quay lại trang web kia để nhập gmail của chúng ta vừa tạo vào



Và sau khi nhập xong thì thông tin secret của chúng ta đã bị lộ, vậy nên khi bạn biết gmail của người khác mà họ có sử dụng trang web này thì có thể thông tin bí mật của họ sẽ bị lộ

Với level Medium hay level High của trang web này sau khi bạn nhập e-mail



Nó hiện một thông báo là đã gửi một e-mail với reset code nhưng khi chúng ta cố gắng bắt tệp tin được gửi đi thì không nhận được nên chúng ta sẽ xem qua source code của trang web

// Debugging

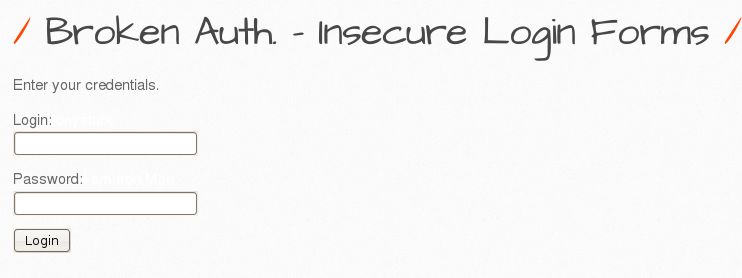
// die("Error: mail was NOT send");

// echo "Mail was NOT send";

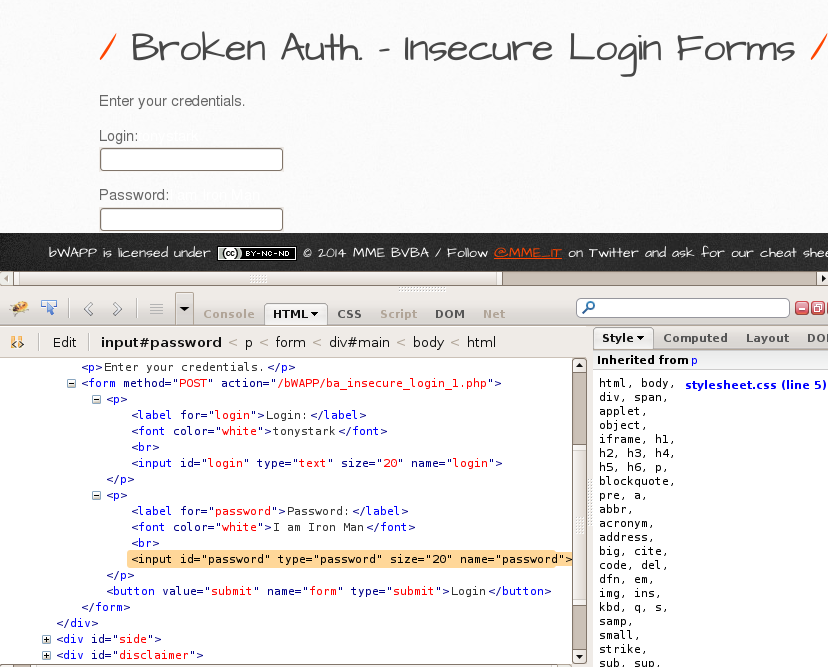
Có vẻ như tập tin mà chúng ta cần được gửi sẽ không bao giờ được gửi bởi debugging của trang web

**Broken Auth. - Insecure Login Forms**

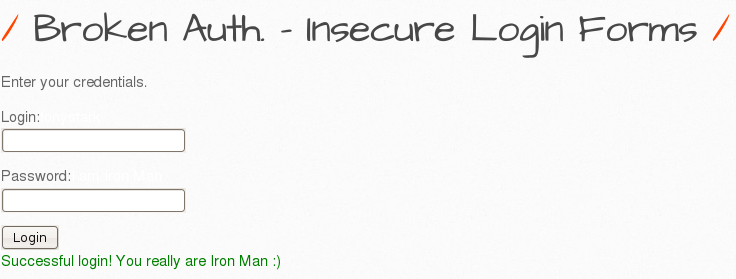
Đây là giao diện của trang web



Lỗ hổng ở level Low của trang web này đó là nó đã hiển thị những thông tin đăng nhập ở ngay trong source code được public

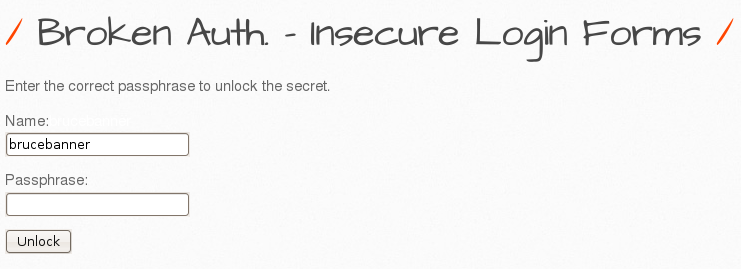


Ta có thể thấy được tên user là tonystark và password là I am Iron Man và ta sẽ đăng nhập thử với những tài nguyên ta có



Và chúng ta đã thành công qua level Low của bài này

Tiếp theo ta sẽ xem level Medium của nó như thế nào



Trang web đã không còn để lộ password cho chũng ta có thể xem như ở level Low nữa nhưng chúng ta có một thứ khá đặc biệt ở trong source code

function unlock\_secret(){

var bWAPP = "bash update killed my shells!"

var a = bWAPP.charAt(0); var d = bWAPP.charAt(3); var r = bWAPP.charAt(16);

var b = bWAPP.charAt(1); var e = bWAPP.charAt(4); var j = bWAPP.charAt(9);

var c = bWAPP.charAt(2); var f = bWAPP.charAt(5); var g = bWAPP.charAt(4);

var j = bWAPP.charAt(9); var h = bWAPP.charAt(6); var l = bWAPP.charAt(11);

var g = bWAPP.charAt(4); var i = bWAPP.charAt(7); var x = bWAPP.charAt(4);

var l = bWAPP.charAt(11); var p = bWAPP.charAt(23); var m = bWAPP.charAt(4);

var s = bWAPP.charAt(17); var k = bWAPP.charAt(10); var d = bWAPP.charAt(23);

var t = bWAPP.charAt(2); var n = bWAPP.charAt(12); var e = bWAPP.charAt(4);

var a = bWAPP.charAt(1); var o = bWAPP.charAt(13); var f = bWAPP.charAt(5);

var b = bWAPP.charAt(1); var q = bWAPP.charAt(15); var h = bWAPP.charAt(9);

var c = bWAPP.charAt(2); var h = bWAPP.charAt(2); var i = bWAPP.charAt(7);

var j = bWAPP.charAt(5); var i = bWAPP.charAt(7); var y = bWAPP.charAt(22);

var g = bWAPP.charAt(1); var p = bWAPP.charAt(4); var p = bWAPP.charAt(28);

var l = bWAPP.charAt(11); var k = bWAPP.charAt(14);

var q = bWAPP.charAt(12); var n = bWAPP.charAt(12);

var m = bWAPP.charAt(4); var o = bWAPP.charAt(19);

var secret = (d + "" + j + "" + k + "" + q + "" + x + "" + t + "" +o + "" + g + "" + h + "" + d + "" + p);

if(document.forms[0].passphrase.value == secret){

// Unlocked

location.href="/bWAPP/ba\_insecure\_login\_2.php?secret=" + secret;

}

else{

// Locked

location.href="/bWAPP/ba\_insecure\_login\_2.php?secret=";

}

}

Có vẻ như khi chúng ta có thể giải xong hàm này thì ta sẽ ra được passphase của user kia

Và ta sẽ bắt đầu giải mã nó. Ta có var secret như vậy

var secret = (d + "" + j + "" + k + "" + q + "" + x + "" + t + "" +o + "" + g + "" + h + "" + d + "" + p);

Vậy kí tự d sẽ tương ứng với = var d = bWAPP.charAt(3) = “h”

Vậy kí tự j sẽ tương ứng với = var j = bWAPP.charAt(5) = “u”

Vậy kí tự k sẽ tương ứng với = var k = bWAPP.charAt(14) = “l”

Vậy kí tự q sẽ tương ứng với = var q = bWAPP.charAt(12) = “k”

Vậy kí tự x sẽ tương ứng với = var x = bWAPP.charAt(4) = “ ”

Vậy kí tự t sẽ tương ứng với = var t = bWAPP.charAt(2) = “s”

Vậy kí tự o sẽ tương ứng với = var o = bWAPP.charAt(19) = “m”

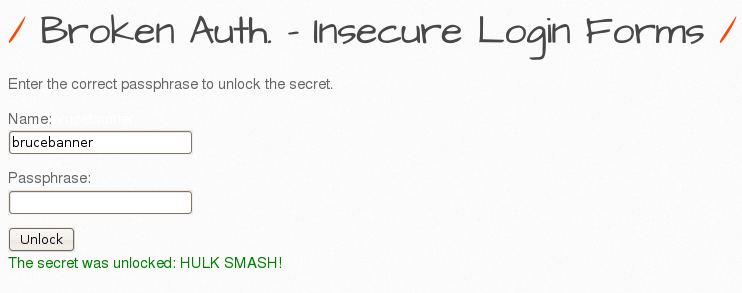
Vậy kí tự g sẽ tương ứng với = var g = bWAPP.charAt(1) = “a”

Vậy kí tự h sẽ tương ứng với = var h = bWAPP.charAt(2) = “s”

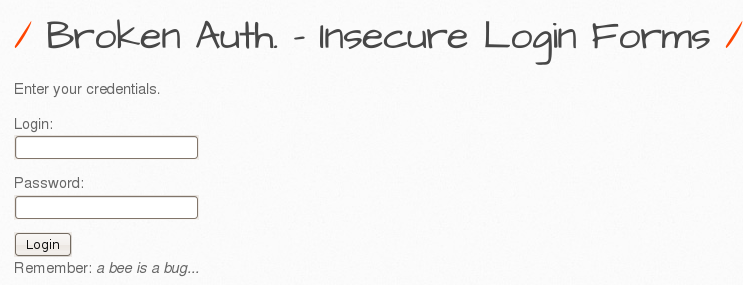
Vậy kí tự d sẽ tương ứng với = var d = bWAPP.charAt(3) = “h”

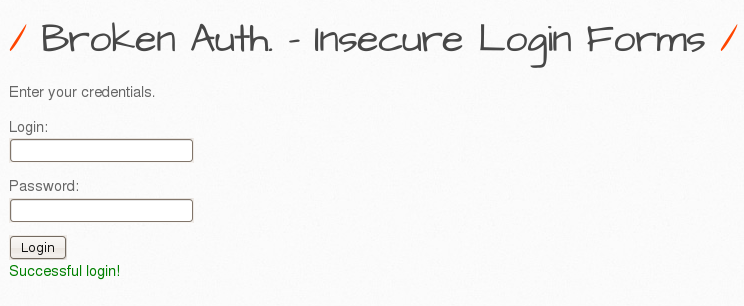
Vậy kí tự p sẽ tương ứng với = var p = bWAPP.charAt(28) = “!”

Vậy passphase ta tìm được là “hulk smash!” và chúng ta sẽ cùng thử xem đã đúng với đề bài chưa



Vậy là chúng ta đã thành công đi qua level Medium của trang web này tiếp theo sẽ là level High

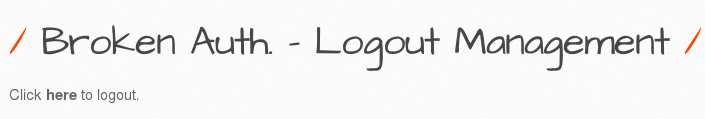


Ở level này những lỗ hổng như ở 2 level trên đã không còn nữa nhưng chúng ta có thể thấy gợi ý của trang web đó là “Remember: a bee is a bug…” và ta cũng biết bee là default user của trang web còn bug là default password của trang web nên chúng ta sẽ đăng nhập thử bằng 2 cái này luôn  


Và chúng ta đã thành công đi hết toàn bộ level của Broken Auth.- Insecure Login Forms

**Broken Auth. - Logout Management**

Đây là giao diện của trang web

****

Đối với bài tập này ban đầu mình cũng khá khó hiểu là chúng ta sẽ khai thác lỗ hổng nào với bài như vậy và sau nhiều lần nghịch thì mình đặt ra giả thiết cho bài tập này như sau

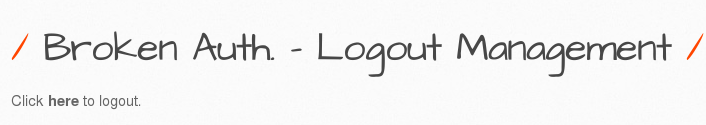
Khi một máy tính có người dùng trong công ty ấn logout và bắt đầu đi ra ngoài và quên khóa máy



Và sau đó một người có ý đồ xấu đến gần máy tính này và nhìn thấy được trên thanh công cụ có nút để back lại trang cũ



Và người này ấn vào

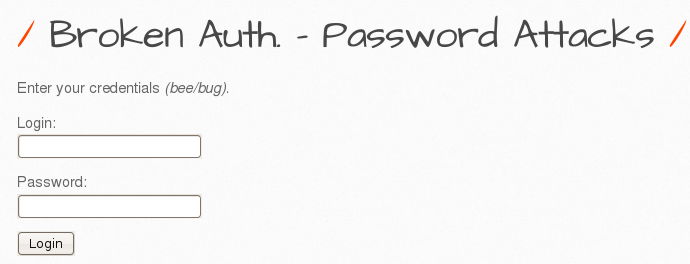


Trang web lại quay lại với tài khoản của người đã quên khóa máy. Và tôi nghĩ đấy là kịch bản để chúng ta khai thác lỗ hổng trong bài toán Logout Management này. Và điều này được xác thực hơn khi ta lên level Medium thì nút back lại sẽ không thể sử dụng được nữa

Và ở level Medium trang web đã sử dụng hàm session\_destroy() để xóa đi toàn bộ những phiên đăng nhập cũ

**Broken Auth. - Password Attacks**

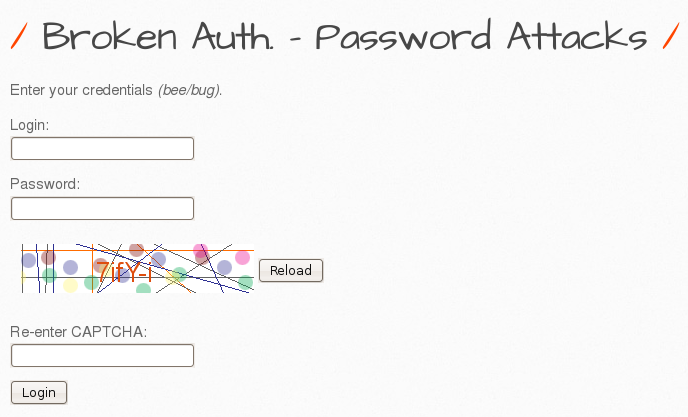
Đây là giao diện của trang web



Ở bài tập này đơn giản chỉ là chúng ta dùng word list để buforce dictionary trang web nhưng chúng ta đã biết trước việc login của trang web là bee và password là bug rồi nên gần như không có gì là khó khăn để đi qua level Low và Medium cả



Nhưng với mức độ High thì sẽ có chút khác biệt



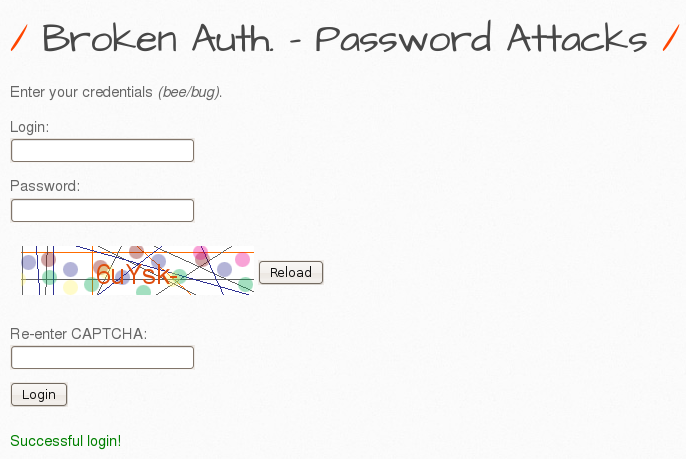
Chúng ta sẽ có thêm mã captra và tôi sẽ hướng dẫn các bạn các ta buforce dictionary khi mà có thêm captcha là chúng ta sẽ sử dụng Burp Suite để có thể tìm được tài khoản cũng như password của người dùng có trong wordlist



Việc chúng ta cần làm là điền đúng mã captcha và đưa login và password vào payload và dùng



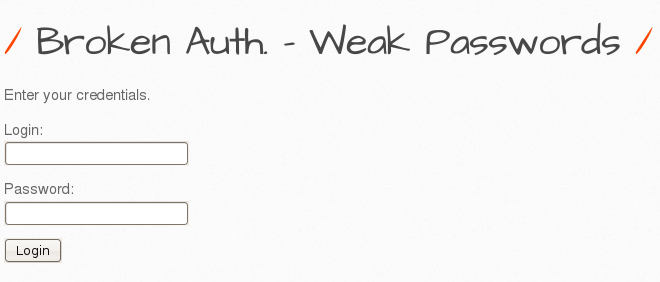
Và sử dụng chức năng cluster bomb để buforce dictionary đến khi hiện flag là successful là được và sau quá trình đó ta sẽ có được user và password là bee và bug



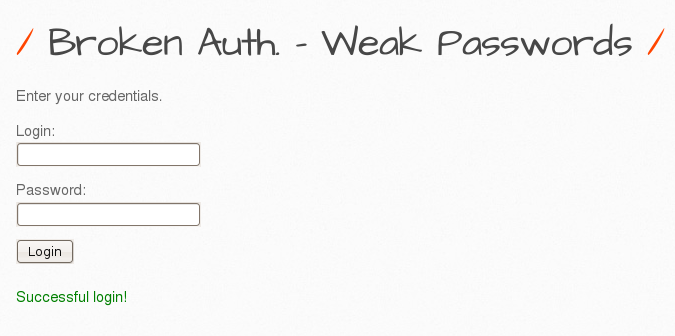
Vậy là ta đã thành công đi qua cả 3 level của Broken Auth. - Password Attacks

**Broken Auth. - Weak Passwords**

Đây là giao diện của trang web



Và bài tập này cũng là chúng ta sử dụng phương pháp buforce dictionary để tìm ra mật khẩu của trang web và với level low đấy là test, test và ta được kết quả và với những level sau cũng tương tự như vậy



Vậy là chúng ta đã thành công đi qua 6 lỗ hổng trong phần Broken Authetication của trang web bWAPP