

phpBB Forum Password Reset Hack

Contributed by Spider
Saturday, 31 March 2007

In this tutorial I will be demonstrating the simplicity in getting around password reset systems that are based on random numbers. Specifically, we will be looking at the very popular opensource forum software phpBB. I won't be providing fully functional applications, to avoid it getting into the wrong hands, but I will illustrate enough for anyone with any coding skills to draft up their own version.

{mosgoogle_right}

Before we start the hack there's a few things we need to get out of the way. The first is to get the servers time. To do this, we can use a number of techniques but I won't be going into them. I'll simply assume that you already know how to do this. The second step is make a password reset request for the account which we want to take over. Note that the email must be sent from the system that is hosting the phpBB forum. Make note of when you make the request, this will become crucial. When the email is sent it will contain a link that will perform the actual password reset. It is this URL that we are going to try and generate with our application. An important thing to note is that the system generated reset URL is only valid for 48 hours. This means that the hack has to be carried out during that time period. Luckily that's more than enough for us to successfully pull it off.

The Technique:

By this stage you should have already successfully sent a password reset request and made note of the time it was made. So let's move onto how this hack is actually going to work. Essentially we're going to generate the exact same URL that was sent in the email.

In order to do this we will need to employ the same algorithm that phpBB has used to generate the address. The way phpBB does it is by using a random number within the URL. Of course, anyone that has done any amount of coding in their life can tell you that random numbers are never truly random. All you need to produce the same random value is the seed that was used within the random number algorithm.

Most systems will use the server clock because the value is always changing and wouldn't you believe it, that's exactly what phpBB does. So it's with this little bit of information that we can generate the same seed that was used when the reset password email was created. By now you've probably started wondering how we're going to know what that seed is. Well the short and sweet of it is that we don't. We're going to be messy and brute force the seed. This is why we need to note the time the email was sent with only a small amount of certainty.

Implementing the Technique:

Since we're going to be brute forcing things here we might as well be efficient and give ourselves a five minute buffer on either side of our recorded reset request time. This should give us a ten minute window from when the reset was mad, which ought to be plenty. The next step is to generate every possible URL that could have been generated during that time period with the intervals acting as the seed. We'll store the URLs in memory with perhaps a linked list or an array. Ultimately the choice is up to you so long as you can access the values later on. The final step is to run through each of the generated URLs to find a successfully validated reset request.

Conclusion

While the process may seem long and tedious, through proper automation and analysis of the process there's a number of ways that one could reduce the amount of generated results. As for those non-coders out there, this would definitely be a good start in understanding simple concepts like loops, conditionals, efficiency and regular expression.