Install CoovaChilli on Ubuntu 20.04

Introduction

- RADIUSdesk can function perfectly without CoovaChilli.
- CoovaChilli however is the best open source captive portal software around.
- If you want to install CoovaChilli on a machine; make sure that there are at least two functional network cards present.
- One network card will be used by CoovaChilli as the Internet connection (WAN).
- The second network card will be used to run a captive portal on (LAN).
- This captive portal is created by the CoovaChilli program who in turn receives it's instructions from FreeRADIUS (RADIUSdesk)

Enable Packet forwarding for IPv4

- The machine running Coova Chilli will act as a router and thus needs to be configured as such.
- This means that the IP packets needs to be **forwarded** from one interface to the other.
- Edit the **/etc/sysctl.conf**
- Find and uncomment **net.ipv4.ip_forward=1** line.

Installing CoovaChilli

- The version of CoovaChilli is 1.6 as of this writing.
- We will download and build the .deb package from source.
- First ensure the required packages to build the .deb package are installed.

```
sudo apt-get install build-essential libssl-dev libjson-c-dev gengetopt
sudo apt install devscripts debhelper
```

- Download the **release 1.6**'s .tar.gz file of the source here: https://github.com/coova/coova-chilli/releases
- Before we can build the package, we have to remove a dependency (**hasrl**) specified in the Debian control file.
- This dependency is not required and including it causes trouble when you want to **install** the package.

```
# If you downloaded with wget
tar -xzvf 1.6.tar.gz
# If you downloaded with the browser
tar -xzvf coova-chilli-1.6.tar.gz
cd coova-chilli-1.6/

vi debian/control
#Look for this part
#------
#Depends:
# ${shlibs:Depends},
# iptables,
# haserl,
# adduser,
#------
#------ CHANGE TO THIS (remove haserl as a dependency)
#------
#Depends:
# ${shlibs:Depends},
# iptables,
# adduser,
#-----

debuild -i -us -uc -b
cd ..
sudo dpkg --install coova-chilli_1.6_amd64.deb
```

- From the output of the dpkg command you will see that CoovaChilli is by default disabled. In the next section we will configure it to become a working entity.

```
Selecting previously unselected package coova-chilli.
(Reading database ... 125842 files and directories currently installed.)
Preparing to unpack coova-chilli_1.6_amd64.deb ...
Unpacking coova-chilli (1.6) ...
Setting up coova-chilli (1.6) ...
Chilli default off. Look at /etc/default/chilli
Processing triggers for libc-bin (2.31-0ubuntu9.2) ...
Processing triggers for systemd (245.4-4ubuntu3.6) ...
Processing triggers for man-db (2.9.1-1) ...
```

Configuring CoovaChilli

Enable CoovaChilli

- Edit the following file

```
    sudo vi /etc/default/chilli
```

- Change it to look like this

```
    START_CHILLI=1
    CONFFILE="/etc/chilli.conf"
    HS_USER="chilli"
```

- Save the file and start CoovaChilli

```
   #First enable the service so it will start up during boot
   sudo systemctl enable chilli
   # Feedback code should be...
   # chilli.service is not a native service, redirecting to systemd-sysv-install.
   # Executing: /lib/systemd/systemd-sysv-install enable chilli
   sudo systemctl start chilli
```

- Check the feedback on our startup attempt.

```
sudo systemctl status chilli
```

- Because we have not yet configured anything; we do get some errors. This will be fixed and Coova should run fine once we configured the program.

```
● chilli.service - LSB: Start CoovaChilli daemon at boot time
     Loaded: loaded (/etc/init.d/chilli; generated)
     Active: active (exited) since Mon 2021-01-18 10:03:07 UTC; 7s ago
       Docs: man:systemd-sysv-generator(8)
    Process: 98449 ExecStart=/etc/init.d/chilli start (code=exited,
status=0/SUCCESS)

Jan 18 10:03:07 osboxes chilli[98540]: TX queue length set to 100
Jan 18 10:03:07 osboxes coova-chilli[98544]: PID 98544 loading binary
options file /var/run/chilli.98540.cfg.bin
Jan 18 10:03:07 osboxes coova-chilli[98544]: Loading modules
Jan 18 10:03:07 osboxes chilli[98540]: No such device: ioctl(SIOCSIFFLAGS)
failed on eth1
Jan 18 10:03:07 osboxes coova-chilli[98544]: USER root(0/0), GROUP root(0/0)
CHILLI[UID 113, GID 118]
Jan 18 10:03:07 osboxes coova-chilli[98544]: Running /etc/chilli/up.sh (0/0)
Jan 18 10:03:07 osboxes chilli[98540]: No such device: ioctl(SIOCSIFFLAGS)
failed
Jan 18 10:03:07 osboxes chilli[98540]: No such device: ioctl(SIOCSIFADDR)
failed
Jan 18 10:03:07 osboxes chilli[98540]: No such device: ioctl(d=2,
request=35111) failed
Jan 18 10:03:07 osboxes chilli[98540]: Failed to create dhcp listener on
eth1
```

Modify the configuration file

- CoovaChilli is configured by editing or creating certain files under the **/etc/chilli** directory.
- Before we proceed with the config we have to touch the bit about our interfaces.
- For CoovaChilli to work correct; there needs to be at least two network interfaces on the machine. (Excluding the loopback interface **lo**)
- One interface will be directly connected to the Internet. We will refer to this interface as the WAN interface
- The other interface will be running the captive portal on top of it. We will refer to this interface as the LAN interface.
- With previous versions of Ubuntu, the network interfaces would have fairly predictable names like **eth0** and **eth1**.
- With the current versions of Ubuntu, the naming convention changed.
- We would advice you to issue the **ip addr** command to see which it present.

```
 ip addr
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
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state
UP group default qlen 1000
    link/ether 08:00:27:fe:57:09 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.111/24 brd 192.168.1.255 scope global dynamic enp0s3
       valid_lft 256221sec preferred_lft 256221sec
    inet6 fe80::a00:27ff:fefe:5709/64 scope link
       valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state
UP group default qlen 1000
    link/ether 08:00:27:8c:d3:32 brd ff:ff:ff:ff:ff:ff
    inet6 fe80::a00:27ff:fe8c:d332/64 scope link
```

- Using this, on our system, the WAN will be **enp0s3** and the LAN will be **enp0s8**.
- Next lets look at how to configure CoovaChilli under **/etc/chilli**.

File	Comment
config	start as a copy of **defaults** and is edited to override specific variables defined in **defaults**
defaults	To avoid the splash screen we have to remove one line from this file
ipup.sh	Custom firewall rules for start-up
ipdown.sh	Custom firewall rule clean-up during shut-down.

- Use the following /etc/chilli/config file as a guideline to configure CoovaChilli

```
HS_WANIF=enp0s3              # WE ALSO Need to specify this
HS_LANIF=enp0s8              # Subscriber Interface for client devices
HS_NETWORK=10.1.0.0        # HotSpot Network (must include HS_UAMLISTEN)
HS_NETMASK=255.255.0.0     # HotSpot Network Netmask
HS_UAMLISTEN=10.1.0.1      # HotSpot IP Address (on subscriber network)
HS_UAMPORT=3990            # HotSpot UAM Port (on subscriber network)
HS_UAMUIPORT=4990          # HotSpot UAM "UI" Port (on subscriber network,
for embedded portal)
HS_NASID=localhost
HS_RADIUS=localhost
HS_RADIUS2=localhost
HS_RADSECRET=testing123    # Set to be your RADIUS shared secret
HS_UAMSECRET=greatsecret     # Set to be your UAM secret
HS_UAMALIASNAME=chilli
HS_SSID="Struisbaai"
HS_NASIP=127.0.0.1    # To explicitly set NAS-IP-Address
HS_UAMSERVER=$HS_UAMLISTEN
HS_UAMFORMAT=http://\$HS_UAMLISTEN/cake4/rd_cake/dynamic-details/chilli-
browser-detect/
HS_MACAUTH=on              # To turn on MAC Authentication
HS_TCP_PORTS="80 23 8000"
HS_MODE=hotspot
HS_TYPE=chillispot
HS_WWWDIR=/etc/chilli/www
HS_WWWBIN=/etc/chilli/wwwsh
```

```
HS_PROVIDER=Coova
HS_PROVIDER_LINK=http://www.coova.org/
HS_LOC_NAME="My HotSpot"           # WISPr Location Name and used in portal
HS_COAPORT=3799
#Please specify the DNS servers of your choice here
#Here we specified out own DSL router and as a fallback one of the Google
servers
HS_DNS1=192.168.1.1
HS_DNS2=8.8.8.8
```

- Comment the following line out of */etc/chilli/defaults*.

```
#   Same principal goes for HS_UAMHOMEPAGE.
#HS_UAMHOMEPAGE=http://\$HS_UAMLISTEN:\$HS_UAMPORT/www/coova.html
```

- Use the following */etc/chilli/ipup.sh* file as a guideline

```
#UAM server specified as 10.1.0.1
iptables -I INPUT -i tun0 -p tcp -m tcp --dport 80 --dst 10.1.0.1 -j ACCEPT
iptables -I INPUT -i tun0 -p tcp -m tcp --dport 443 --dst 10.1.0.1 -j ACCEPT
iptables -I INPUT -i tun0 -p tcp -m tcp --dport 22 --dst 10.1.0.1 -j ACCEPT
iptables -I INPUT -i tun0 -p tcp -m tcp --dport 8000 --dst 10.1.0.1 -j
ACCEPT
# force-add the final rule necessary to fix routing tables (Enabling NAT)
iptables -F POSTROUTING -t nat
iptables -I POSTROUTING -t nat -o $HS_WANIF -j MASQUERADE
```

- Use the following */etc/chilli/ipdown.sh* file as a guideline

```
#UAM server specified as 10.1.0.1
iptables -D INPUT -i tun0 -p tcp -m tcp --dport 80 --dst 10.1.0.1 -j ACCEPT
iptables -D INPUT -i tun0 -p tcp -m tcp --dport 443 --dst 10.1.0.1 -j ACCEPT
iptables -D INPUT -i tun0 -p tcp -m tcp --dport 22 --dst 10.1.0.1 -j ACCEPT
iptables -D INPUT -i tun0 -p tcp -m tcp --dport 8000 --dst 10.1.0.1 -j
ACCEPT
```

Test it out

- Restart CoovaChilli for the latest changes to be effected.

```
sudo systemctl stop chilli
sudo systemctl status chilli
sudo systemctl start chilli
```

- Confirm it started fine

```
sudo systemctl status chilli

.......
● chilli.service - LSB: Start CoovaChilli daemon at boot time
   Loaded: loaded (/etc/init.d/chilli; generated)
   Active: active (running) since Sat 2019-12-21 03:05:26 UTC; 2s ago
```

```
     Docs: man:systemd-sysv-generator(8)
  Process: 7619 ExecStart=/etc/init.d/chilli start (code=exited,
status=0/SUCCES
    Tasks: 1 (limit: 1108)
   CGroup: /system.slice/chilli.service
           └─7706 /usr/sbin/chilli -c /etc/chilli.conf

Dec 21 03:05:26 osboxes systemd[1]: Started LSB: Start CoovaChilli daemon at
boo
Dec 21 03:05:26 osboxes chilli[7706]: PID 7706 saving options to
/var/run/chilli
Dec 21 03:05:26 osboxes chilli[7706]: PID 7706 loading binary options file
/var/
Dec 21 03:05:26 osboxes chilli[7706]: Loading modules
Dec 21 03:05:26 osboxes chilli[7706]: CoovaChilli 1.4. Copyright 2002-2005
Mondr
Dec 21 03:05:26 osboxes chilli[7706]: TX queue length set to 100
Dec 21 03:05:26 osboxes coova-chilli[7713]: PID 7713 loading binary options
file
Dec 21 03:05:26 osboxes coova-chilli[7713]: Loading modules
Dec 21 03:05:26 osboxes coova-chilli[7713]: USER root(0/0), GROUP root(0/0)
CHIL
Dec 21 03:05:26 osboxes coova-chilli[7713]: Running /etc/chilli/up.sh (0/0)
.......
```

- Reboot the system and make sure CoovaChilli started up fine

Troubleshooting tips

- When things does not work is can be a bit tricky to figure out which part does not work.
- It might help if you see the captive portal as a mini router with a WAN side and a LAN side.
- So then there are a couple of important things to check.

Are you getting an IP address

- The captive portal also serve as a DHCP server and a device connecting to it with DHCP enabled should get an IP address from it.
- You can check both sides (server and client)
- To check on the server issue the following command.

```
sudo chilli_query list
08-00-27-54-A5-85 10.1.0.3 dnat 157706717100000002 0 08-00-27-54-A5-85 0/0
0/0 0/0 0/0 0 0 0/0 0/0 -
08-00-27-8C-D3-32 10.1.0.2 dnat 157706713900000001 0 08-00-27-8C-D3-32 0/0
0/0 0/0 0/0 0 0 0/0 0/0 -
```

- You can read more about the **chilli_query** command here: Chilli Query
- After consulting the documentation we can conclude that the captive portal has two clients connected but none has been authenticated.
- This means that they 'should' be redirected to a login page.
- If you are not redirected to a login page we can try the following procedure which basically

comes down to two things that is not working as intended.
  - The routing between the LAN and WAN is not working correct.
  - The DNS on the setup is not working correct.

Forcing the login page to display

- CoovaChilli has a special URL which will log you out and redirect you to the login page.
- The URL is http://1.0.0.0
- If you do get a login page you can next try to test the routing and the DNS.

Test the routing

- To test the routing you can use the chilli_query command and manually authorize the client.

```
#Show the current list
sudo chilli_query list
08-00-27-54-A5-85 10.1.0.3 dnat 157706717100000002 0 08-00-27-54-A5-85 0/0
0/0 0/0 0/0 0 0 0/0 0/0 http://detectportal.firefox.com/success.txt
08-00-27-8C-D3-32 10.1.0.2 dnat 157706713900000001 0 08-00-27-8C-D3-32 0/0
0/0 0/0 0/0 0 0 0/0 0/0 -
#Authorize the client we want to test
 sudo chilli_query authorize ip 10.1.0.3
#Note how 'dnat' now changed to 'pass' and the 5th field changed from 0 to 1
system@osboxes:~$ sudo chilli_query list
08-00-27-54-A5-85 10.1.0.3 pass 157706717100000002 1 08-00-27-54-A5-85 6/0
0/0 7073/0 3253/0 0 0 0%/0 0%/0 http://detectportal.firefox.com/success.txt
08-00-27-8C-D3-32 10.1.0.2 dnat 157706713900000001 0 08-00-27-8C-D3-32 0/0
0/0 0/0 0/0 0 0 0/0 0/0 -
```

- On the client you can now try to go to an IP Address that is reachable on the WAN side. I tried to get to the IP Address of my DSL router (192.168.1.1) and could reach it. This means that the traffic flow between the LAN of my captive portal to the LAN of the captive portal is fine.
- I can now again log this client off using the chilli_query command

```
sudo chilli_query logoff ip 10.1.0.3
#'pass' changed again back to 'dnat'
system@osboxes:~$ sudo chilli_query list
08-00-27-54-A5-85 10.1.0.3 dnat 157706939200000002 0 08-00-27-54-A5-85 0/0
0/0 460326/0 146821/0 0 0 0/0 0/0
http://detectportal.firefox.com/success.txt
08-00-27-8C-D3-32 10.1.0.2 dnat 157706713900000001 0 08-00-27-8C-D3-32 0/0
0/0 0/0 0/0 0 0 0/0 0/0 -
```

- Next we will test DNS

Test the DNS

- DNS traffic has to flow regardless of a client being authurised (pass) or not (dnat).
- In order for the client to magically pop up the login page, DNS has also to work correct.
- A common problem is that sometimes the client has their own DNS servers specified and then (depending on the configuration settings of CoovaChilli) it might not allow the DNS traffic to those server through. (Leaving things broken).
- After you confirmed that the client does not have any hard defined DNS servers you can try a

ping test.
- With a ping test you just want to test and confirm that the DNS is working correct on the client.
- You can try and ping any known FQDN and see if the system resolve that to an IP address.
- Here I try to ping www.radiusdesk.com. As you can see the name resolution worked correct, but since I am not yet authorized (still in dnat state) the pings are not going through which is fine.

```
ping www.radiusdesk.com
PING radiusdesk.com (164.160.91.12) 56(84) bytes of data.
^C
--- radiusdesk.com ping statistics ---
2 packets transmitted, 0 received, 100% packet loss, time 1025ms
```

Conclusion

- By using these check points on the captive portal setup, you can now point to a component which does not work as intended and try to resolve it.
  - Be it the login page.
  - The routing between WAN and LAN.
  - The DNS service.

From:
http://radiusdesk.com/wiki/ - **RADIUSdesk**

Permanent link:
http://radiusdesk.com/wiki/getting_started/20_install_ubuntu_coovachilli

Last update: **2022/08/31 22:01**