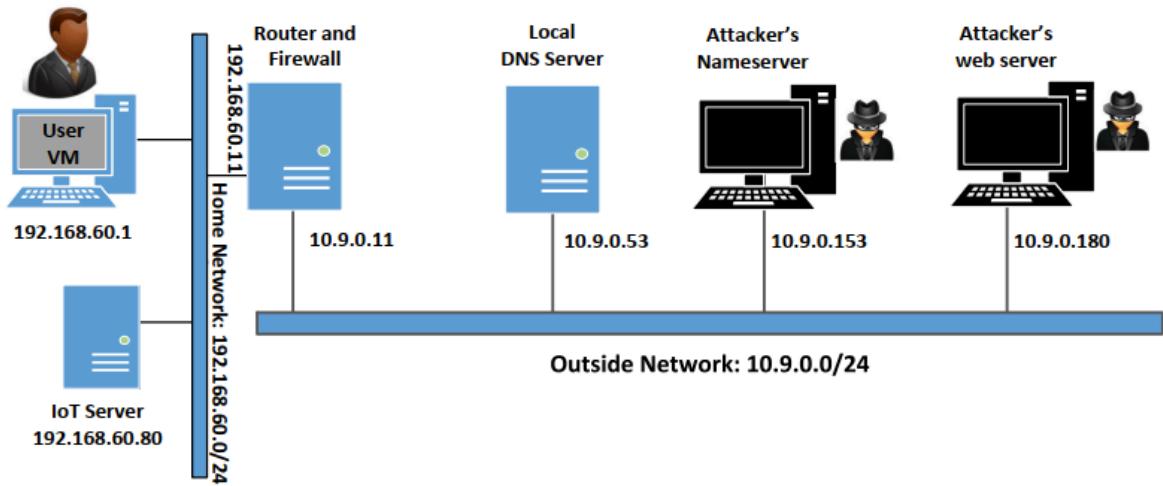


DNS_Rebinding Attack Lab

Before

老样子搭建环境



顺便记录一下ID

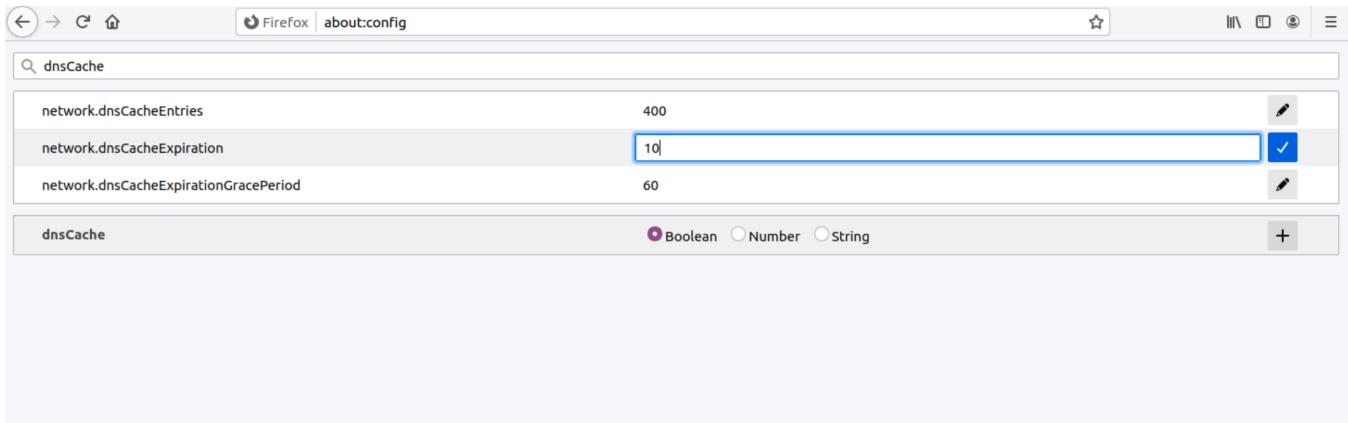
```
[08/06/25] seed@VM:~/.../volumes$ dockps
d43162db634d local-dns-server-10.9.0.53
59302f75ccaa attacker-ns-10.9.0.153
9be18ab94e3d router
524906a099f0 iot-192.168.60.80
6c4eb680e1de attacker-www-10.9.0.180
```

接下来还要进一步配置虚拟机

step0：禁用 Firefox 浏览器的 "DNS over HTTPS" 功能

版本较新的 Firefox 浏览器会默认启用 "DNS over HTTPS" 功能。在该模式下，域名系统（DNS）解析可能不会经过本地 DNS 服务器或/etc/hosts文件。这会给本实验带来问题，因此我们需要将其禁用。进入 Setting，点击"Privacy & Security"选项卡；找到 "DNS over HTTPS" 选项，然后选择 off。

step1：减少Firefox的DNS缓存时间



按手册要求操作，记得重启浏览器

step2：修改/etc/hosts

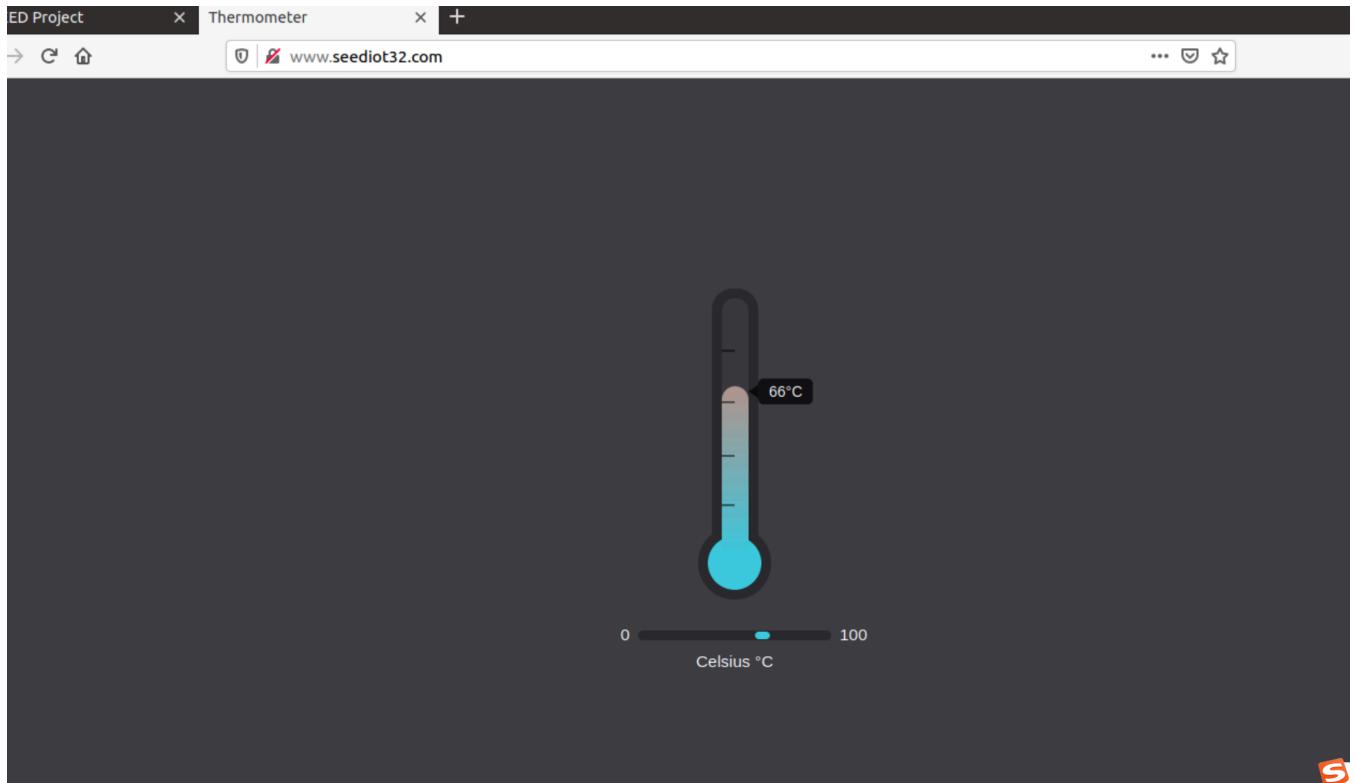
但是当我们检查一下这个文件时，会发现已经按Lab加入了条目

```
[08/06/25] seed@VM:~/.../volumes$ cat /etc/hosts
127.0.0.1      localhost
127.0.1.1      VM

# The following lines are desirable for IPv6 capable hosts
::1      ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters

# For DNS Rebinding Lab
192.168.60.80    www.seedIoT32.com
```

让我们用浏览器测试一下物联网服务器<http://www.seedIoT32.com>



嗯，可以看到物联网服务器运行良好

step3：本地DNS服务器

按要求修改文件

在 `/etc/resolvconf/resolv.conf.d/head` 文件中添加以下条目（在我们的设置中，10.9.0.53 是本地域名系统（DNS）服务器的 IP 地址），注意修改时要在前面输入 sudo 保证权限

```
nameserver 10.9.0.53
```

这个 head 文件的内容将被添加到动态生成的解析器配置文件的开头。这个文件本来只有注释（事实上不止一行注释），进行更改后，我们需要运行以下命令以使更改生效：

```
$ sudo resolvconf -u
```

```
[08/06/25] seed@VM:~/.../volumes$ 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
# 127.0.0.53 is the systemd-resolved stub resolver.
# run "systemd-resolve --status" to see details about the actual nameservers.
nameserver 10.9.0.53
nameserver 127.0.0.53
```

说明修改成功

Testing

按要求完成两次dig操作

```
[08/06/25]seed@VM:~/.../volumes$ dig www.attacker32.com

; <>> DiG 9.16.1-Ubuntu <>> www.attacker32.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 18879
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: ee80bbdbd5921d9e0100000068931ddab057f7132fe3946b (good)
;; QUESTION SECTION:
;www.attacker32.com.           IN      A

;; ANSWER SECTION:
www.attacker32.com.    259200  IN      A      10.9.0.180

;; Query time: 3 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Wed Aug 06 05:18:18 EDT 2025
;; MSG SIZE rcvd: 91
```

```
[08/06/25]seed@VM:~/.../volumes$ dig ns.attacker32.com

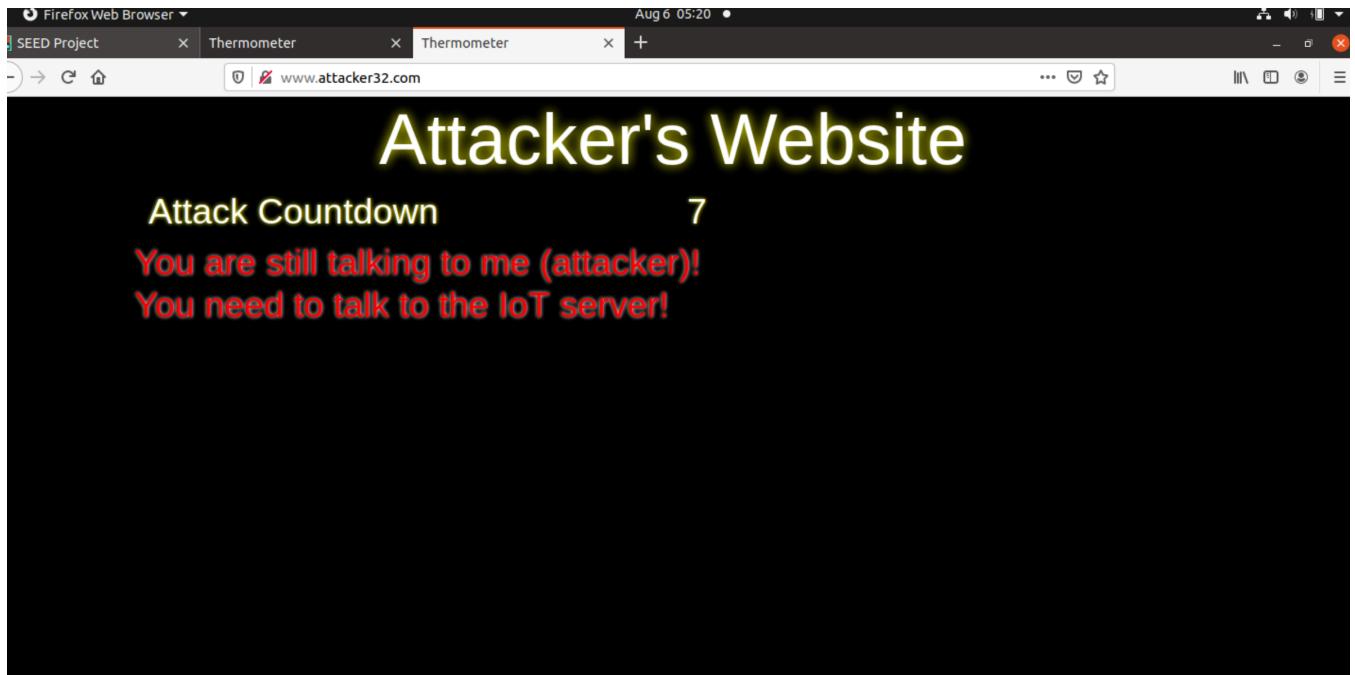
; <>> DiG 9.16.1-Ubuntu <>> ns.attacker32.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 59367
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: f251c832e20e4b780100000068931df23fe44c6df2515a2d (good)
;; QUESTION SECTION:
;ns.attacker32.com.           IN      A

;; ANSWER SECTION:
ns.attacker32.com.    259200  IN      A      10.9.0.153

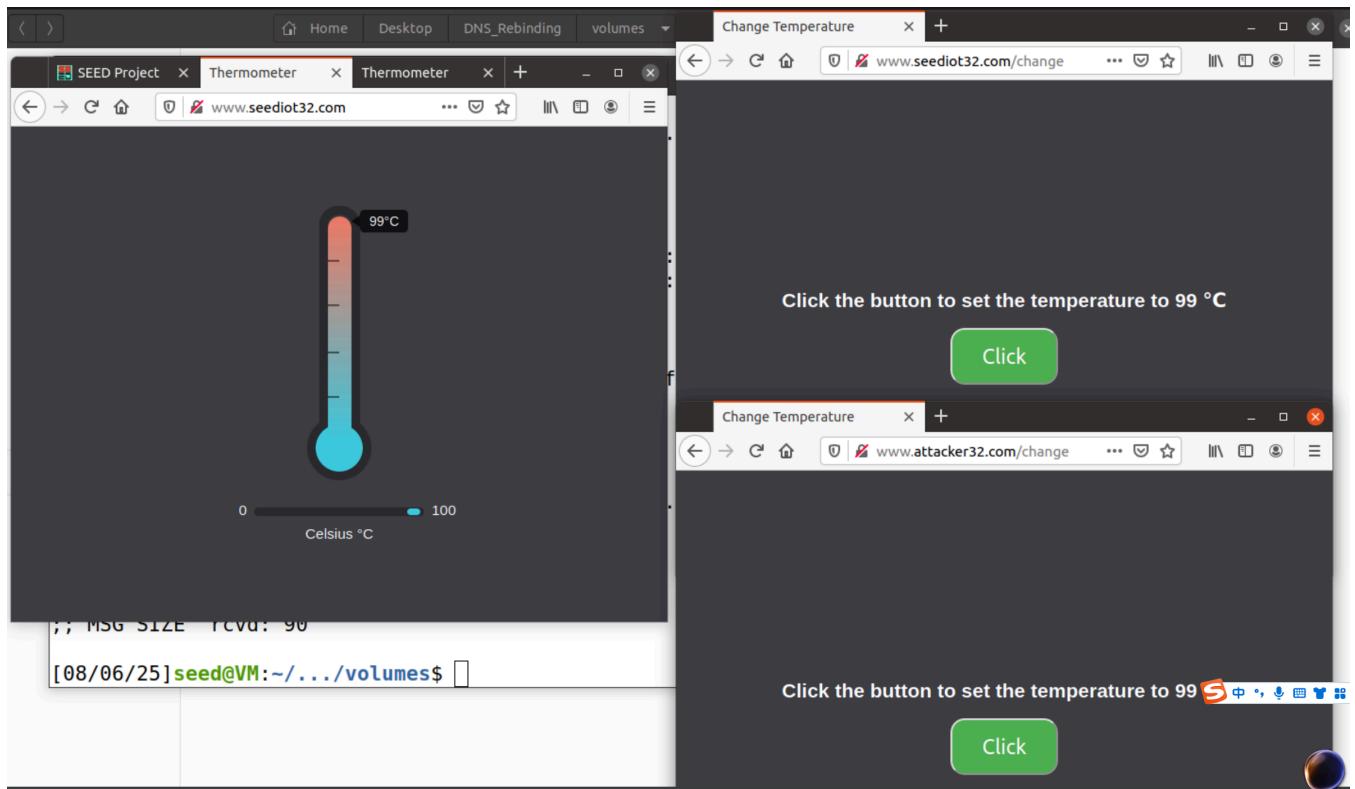
;; Query time: 0 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Wed Aug 06 05:18:42 EDT 2025
;; MSG SIZE rcvd: 90
```

再测试一下攻击者的网站



显然我们已经完成了所有准备工作

Task1:理解同源策略防护



按要求打开三个网页，测试后发现只有<http://www.seediot32.com/change> 的按钮起作用

至于为什么另一个不能？

! Cross-Origin Request Blocked: The Same Origin Policy disallows reading the remote resource at <http://www.seediot32.com/password>. (Reason: CORS header 'Access-Control-Allow-Origin' missing). [\[Learn More\]](#).

这就是由于同源策略保护，简单来讲就是禁止不同域名或不同端口的网页互相访问，不过有一点就是这个策略是【默认拒绝，显示授权】的，属于Web安全的底层逻辑

Task2：攻破同源策略保护

实验手册上提到了一点关键信息——同源策略的执行基于主机名，而不是IP地址

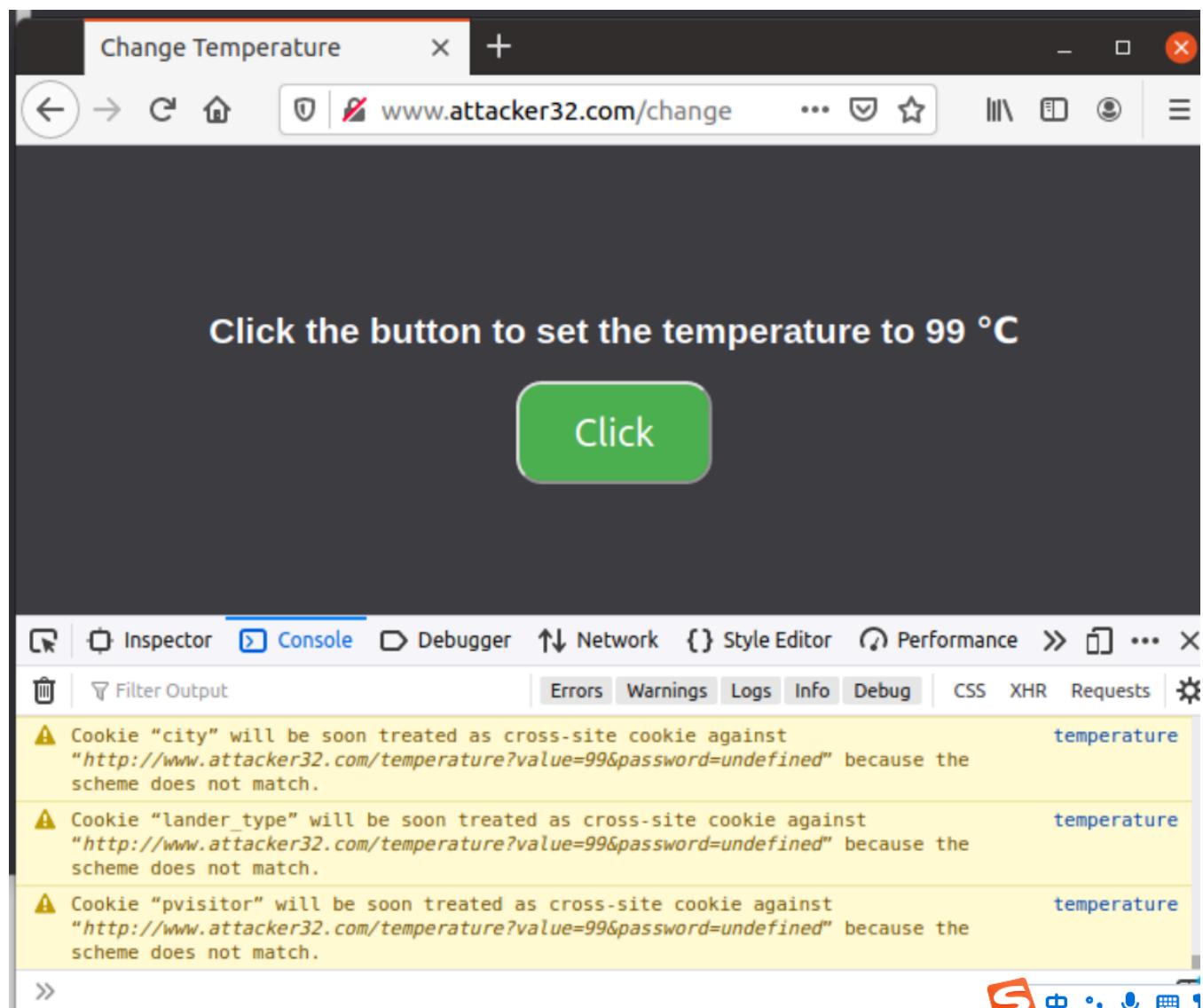
step1：修改JavaScript代码

进入attacker-www-10.9.0.180容器，使用nano命令修改/app/rebind_server/templates/js/change.js文件，将第一行修改为

```
let url_prefix = 'http://www.attacker32.com'
```

完成修改后重启该容器

刷新www.attacker32.com/change页面后重新点击按钮，可见错误不再显示



但此时物联网设备实际上未改变，这是由于此请求实际发往www.attacker32.com而不是物联网设备的网页

step2:进行DNS重绑定

首先我们需要修改DNS映射关系，按要求修改attacker-ns-10.9.0.15的/etc/bind/zone_attacker32.com文件中的内容

```
GNU nano 4.8                               zone_attacker32.com
$TTL 1000
@      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
```

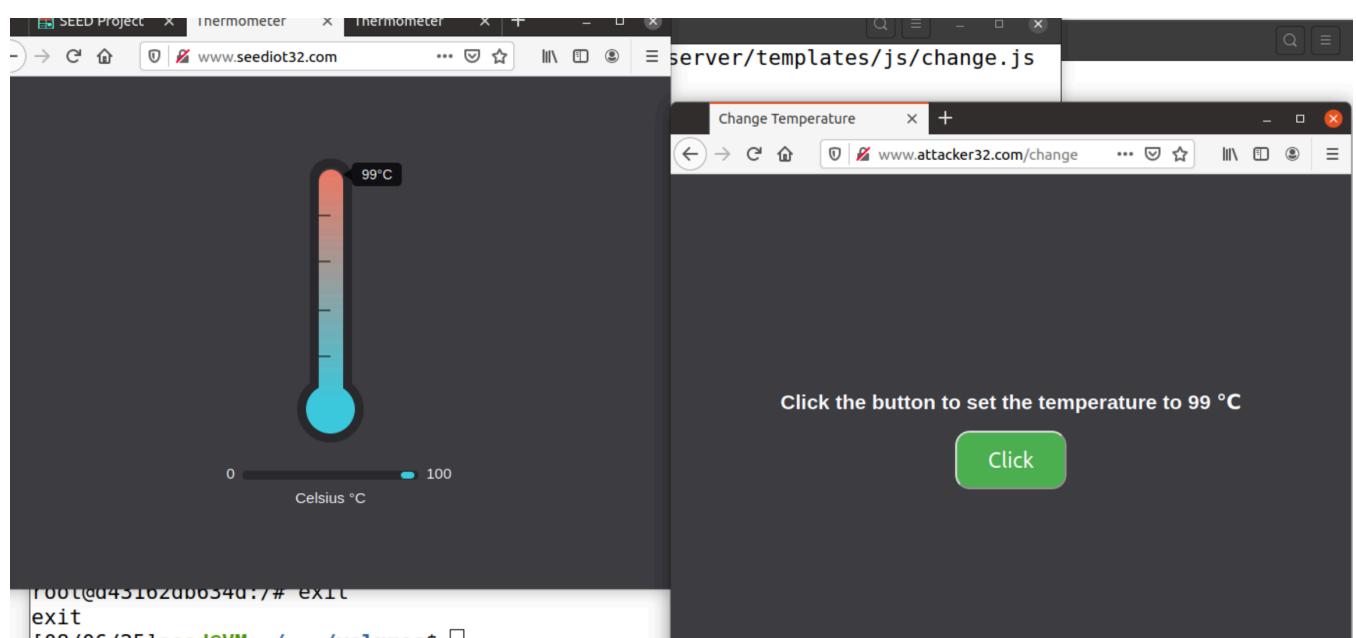
主要是ttl和www对应的ip

然后运行下面的命令让域名服务器重新加载修订后的配置

```
# rndc reload attacker32.com
```

最后别忘了清除本地DNS服务器的缓存

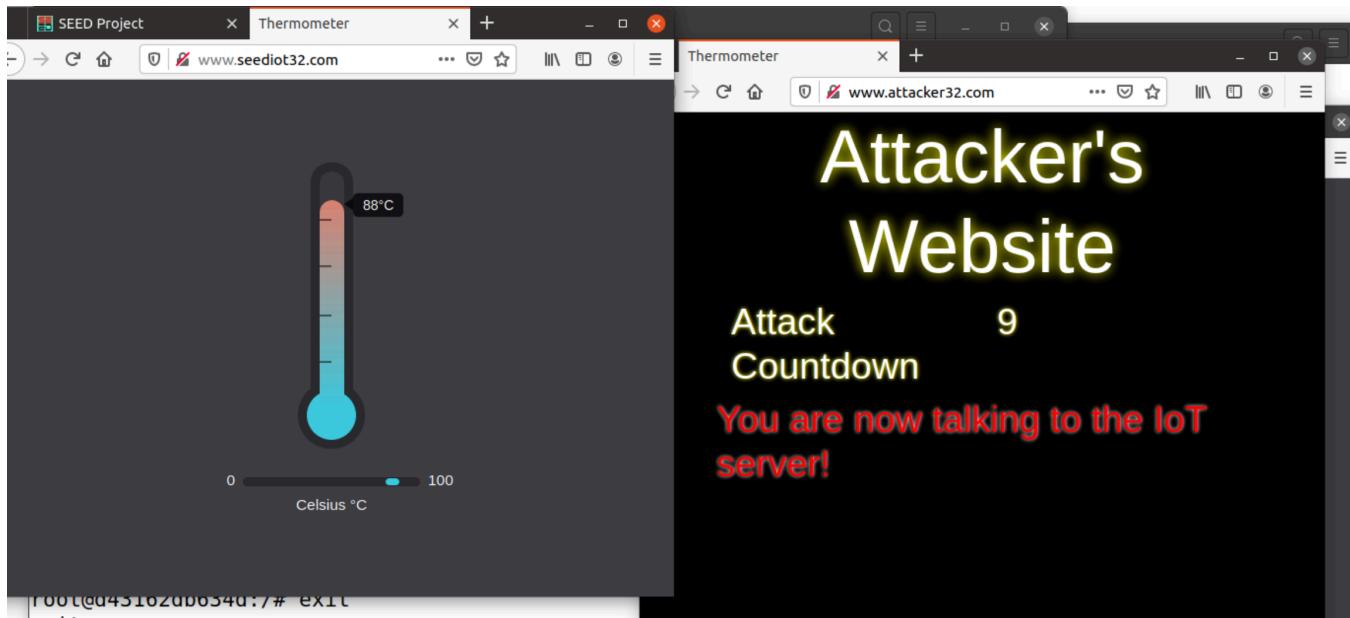
让我们从 www.attacker32.com 的页面点击 change 按钮



会发现恒温器的温度设置成功

Task3：实施攻击

登上攻击的网页查看，每10s，恒温器会自动刷新到88度



显然攻击成功

本次实验较为简单，工具基本现成，只需略微修改数据即可，不过对JavaScript不熟悉导致一开始不怎么想动手。实际上也不怎么用的到，呵呵，当然，对于同源策略保护的陌生和对web相关知识的不了解使得对于攻击的原理理解还不够深刻。

Over!