**Lab Report of DNS\_LOCAL**

Tips:

Machine A IP=10.0.2.5 Name: Ayase(User)

Machine B IP=10.0.2.15 Name: Kirino(Attacker)

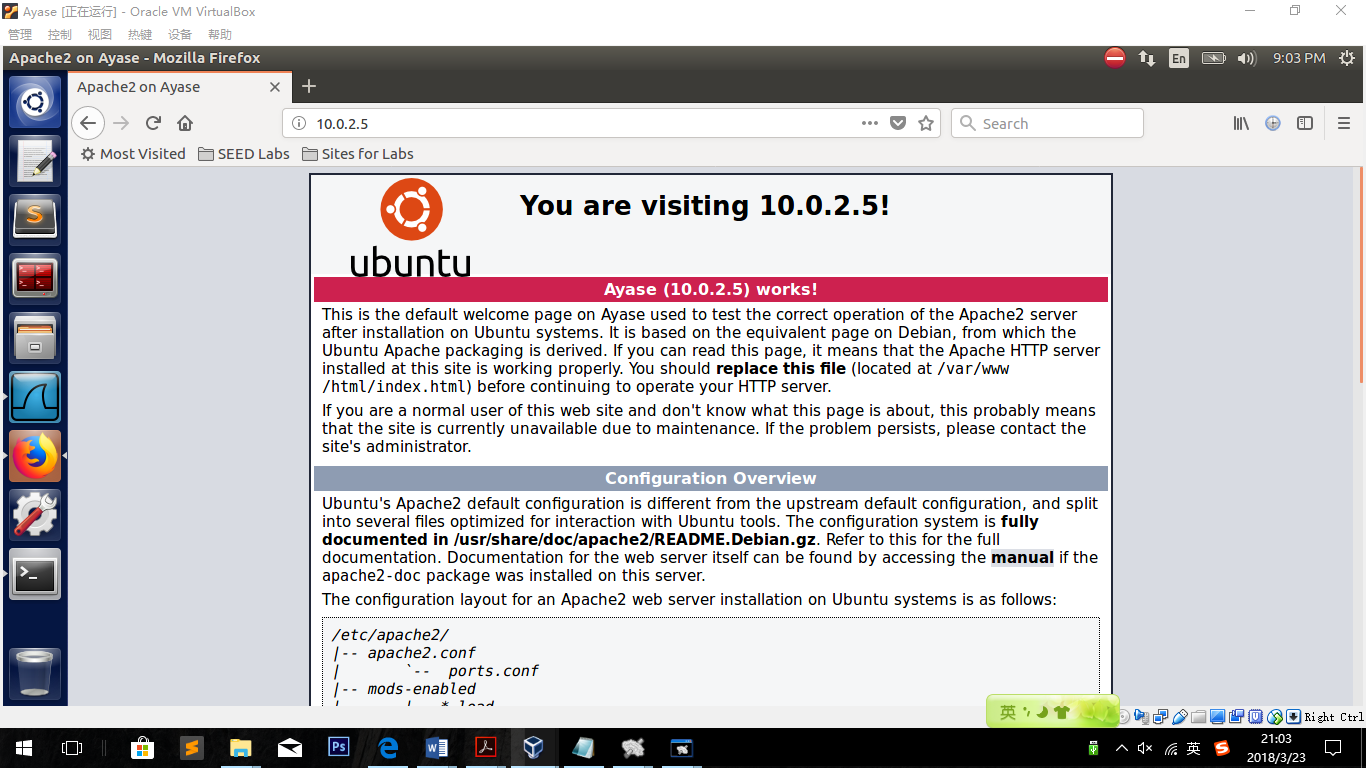
Machine C IP=10.0.2.6 Name: Kanako(Server)

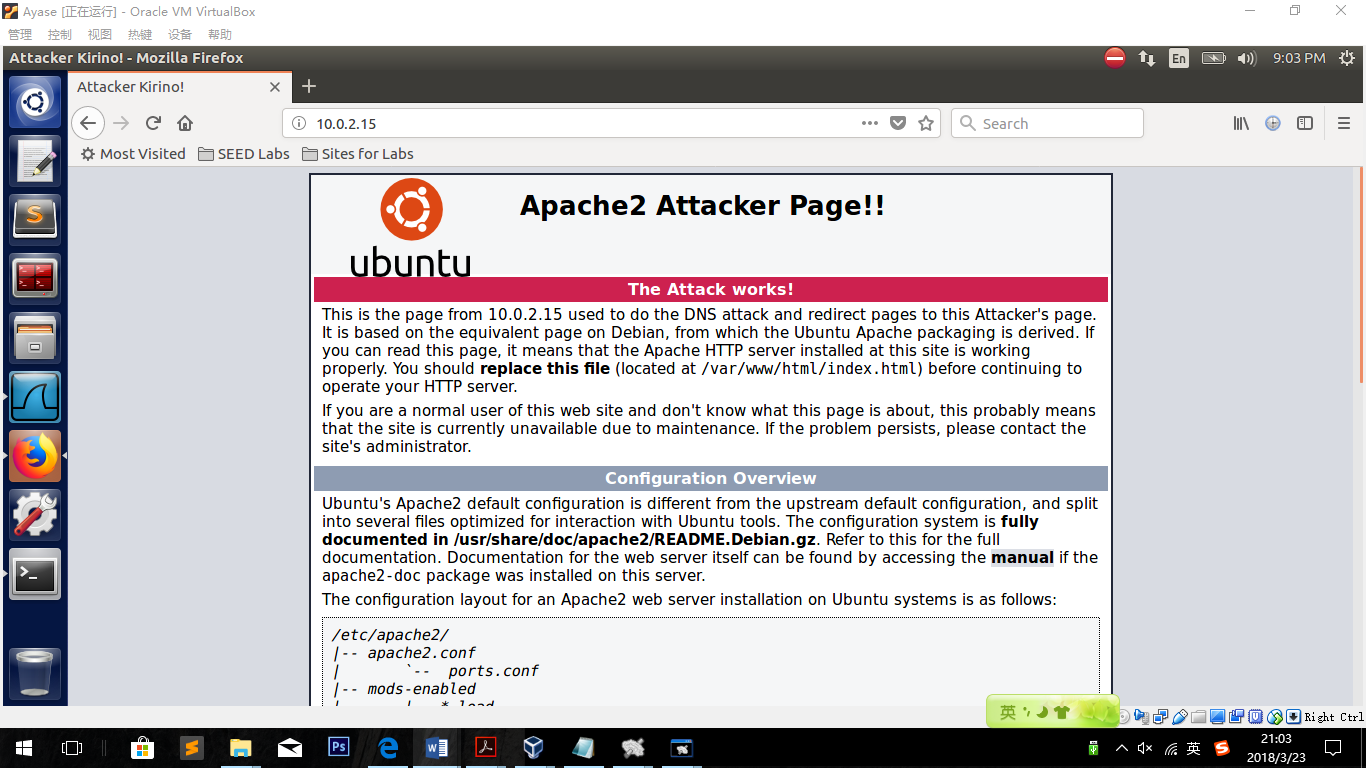
Machine A has another NIC only for SSH connection with host Windows PC, ip address is 192.168.56.101

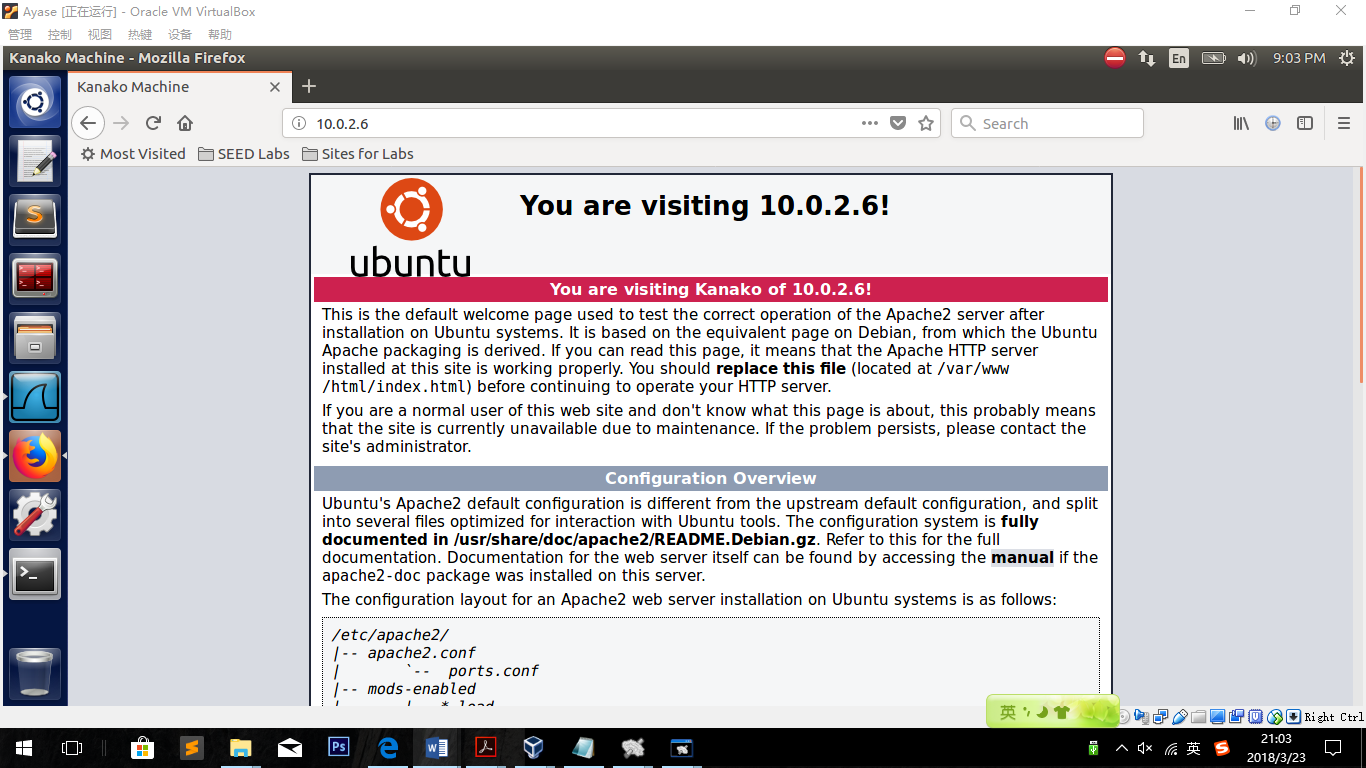
Machine B has another NIC only for SSH connection with host Windows PC, ip address is 192.168.56.102

Machine C has another NIC only for SSH connection with host Windows PC, ip address is 192.168.56.103

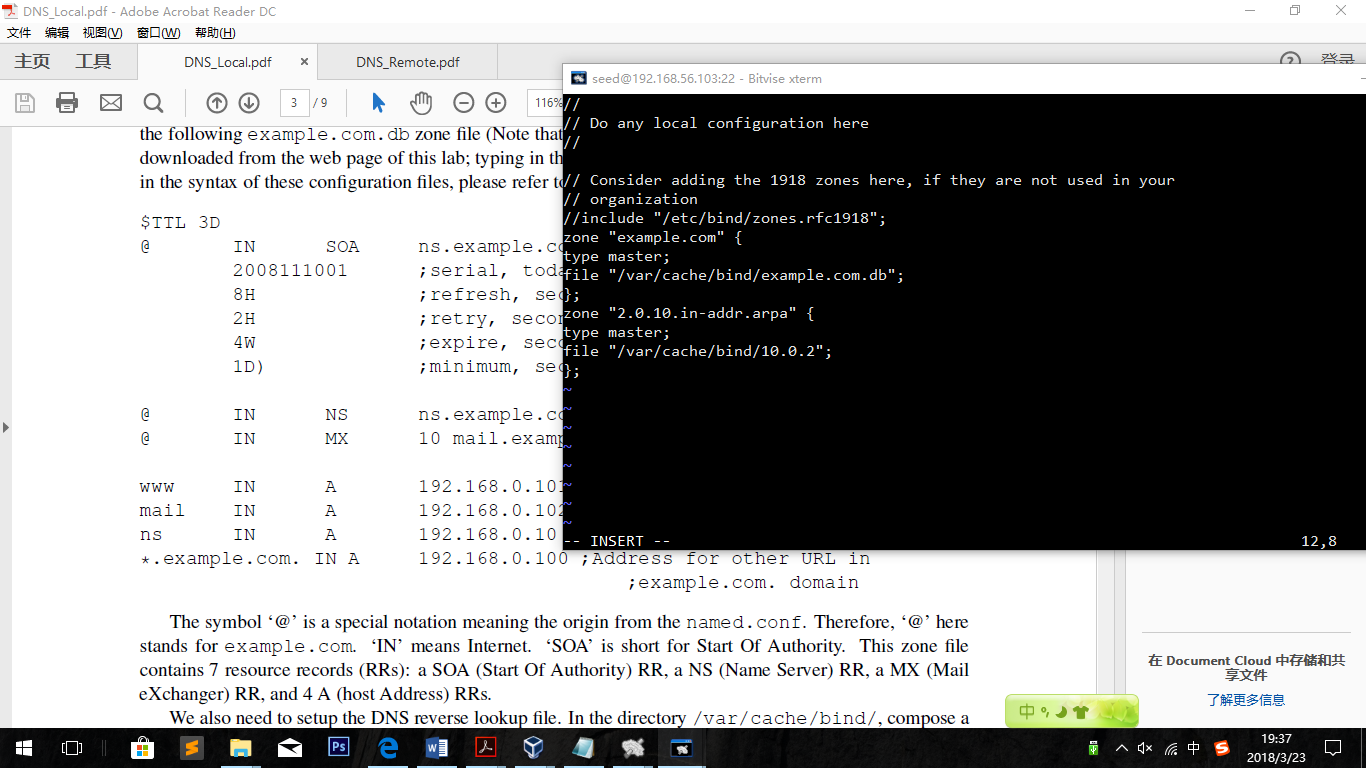
To see results clearly, I modified the home apache pages of the 3:

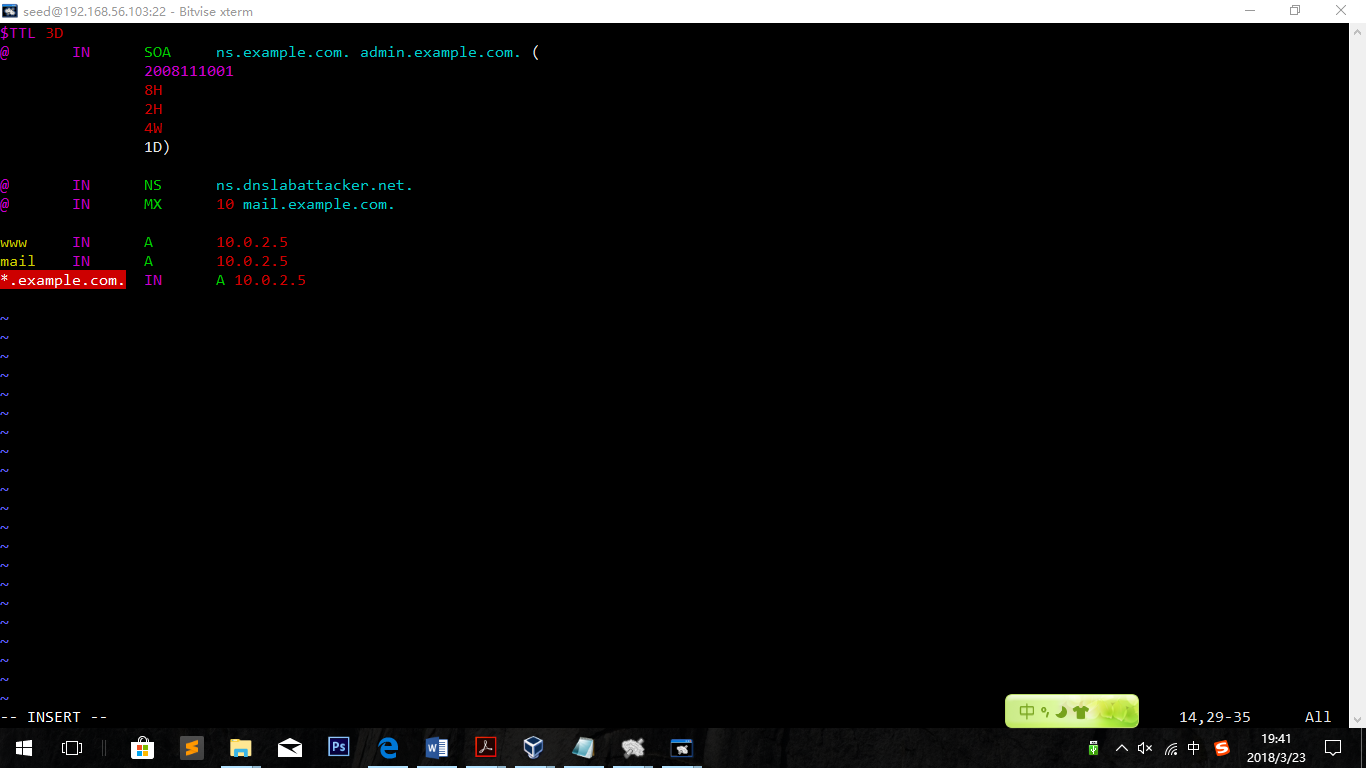


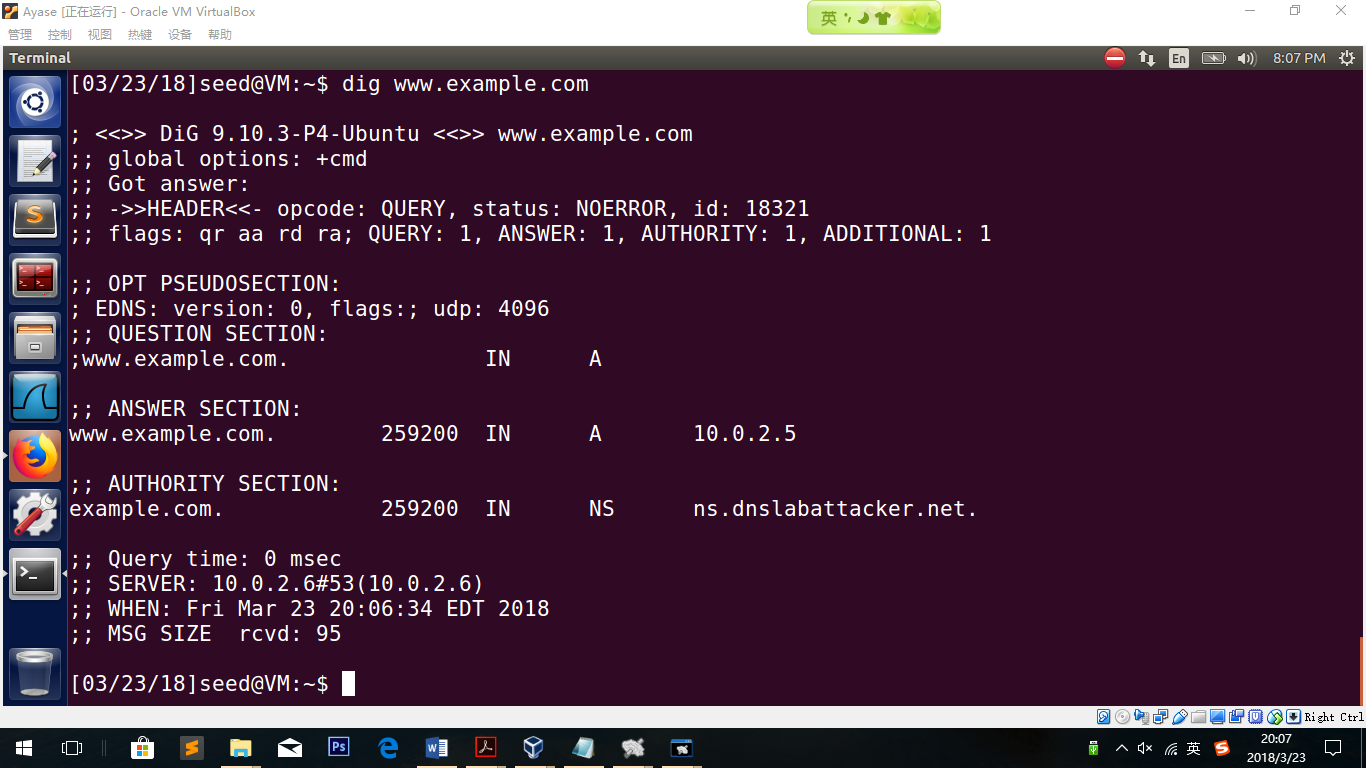




**Zone files:**





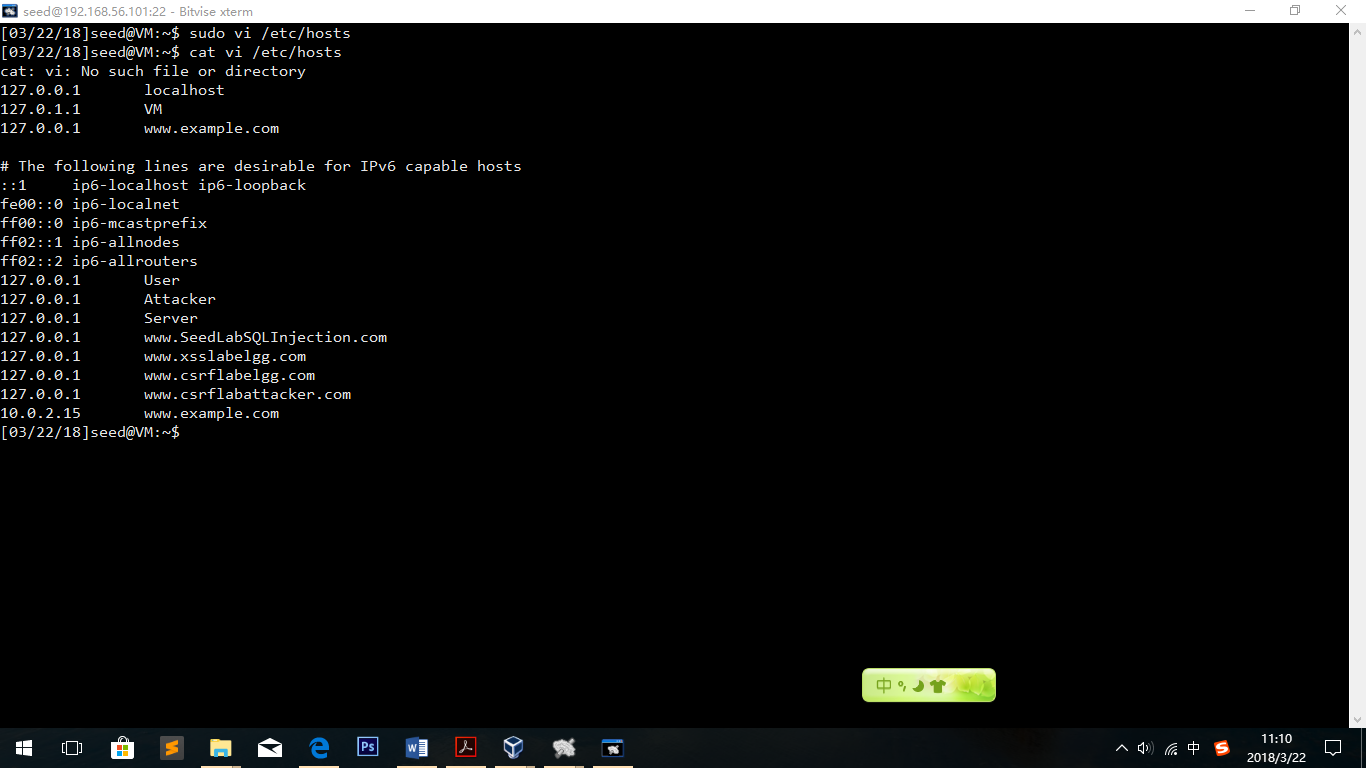


**3.1 Task 1: Attackers have already compromised the victim’s machine**

Tip:

I was doing this before those configuration, so at this time the DNS Server does not work and [www.example.com](http://www.example.com) should be the example page.

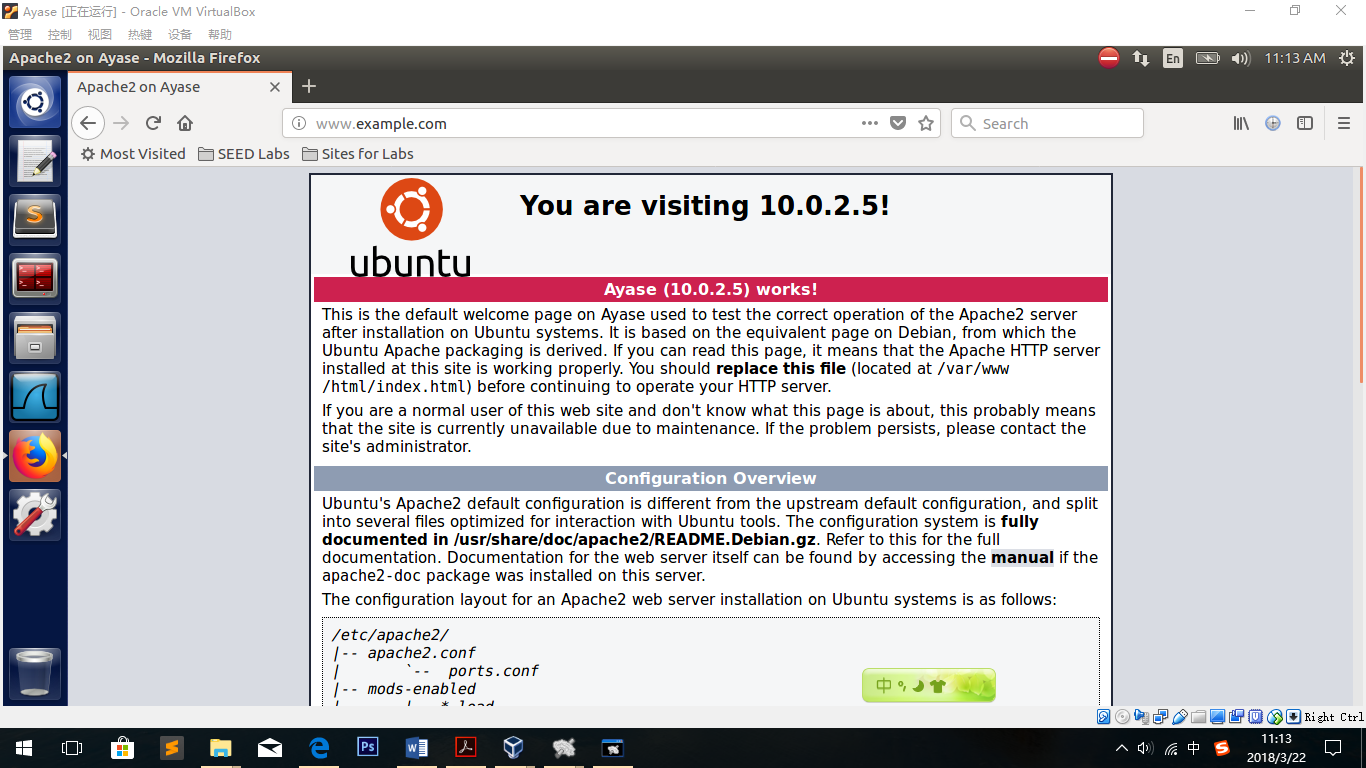
Modifying HOSTS file on Machine Ayase



Observation:

The website [www.example.com](http://www.example.com) has been redirected to 10.0.2.15(which is the attacker’s apache website)

Visit the website on Machine Ayase:



Observation:

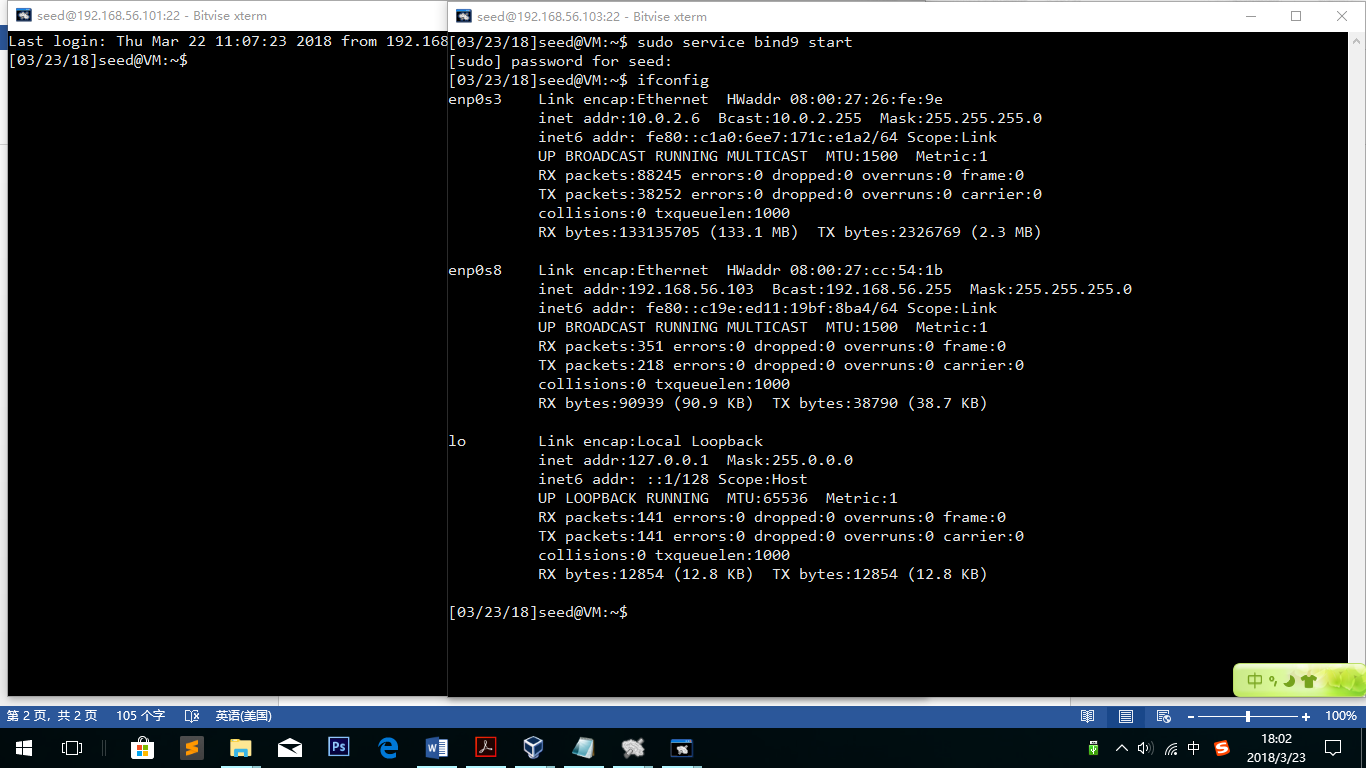
The website [www.example.com](http://www.example.com) has been redirected to Machine A’s homepage successfully.

Explanation:

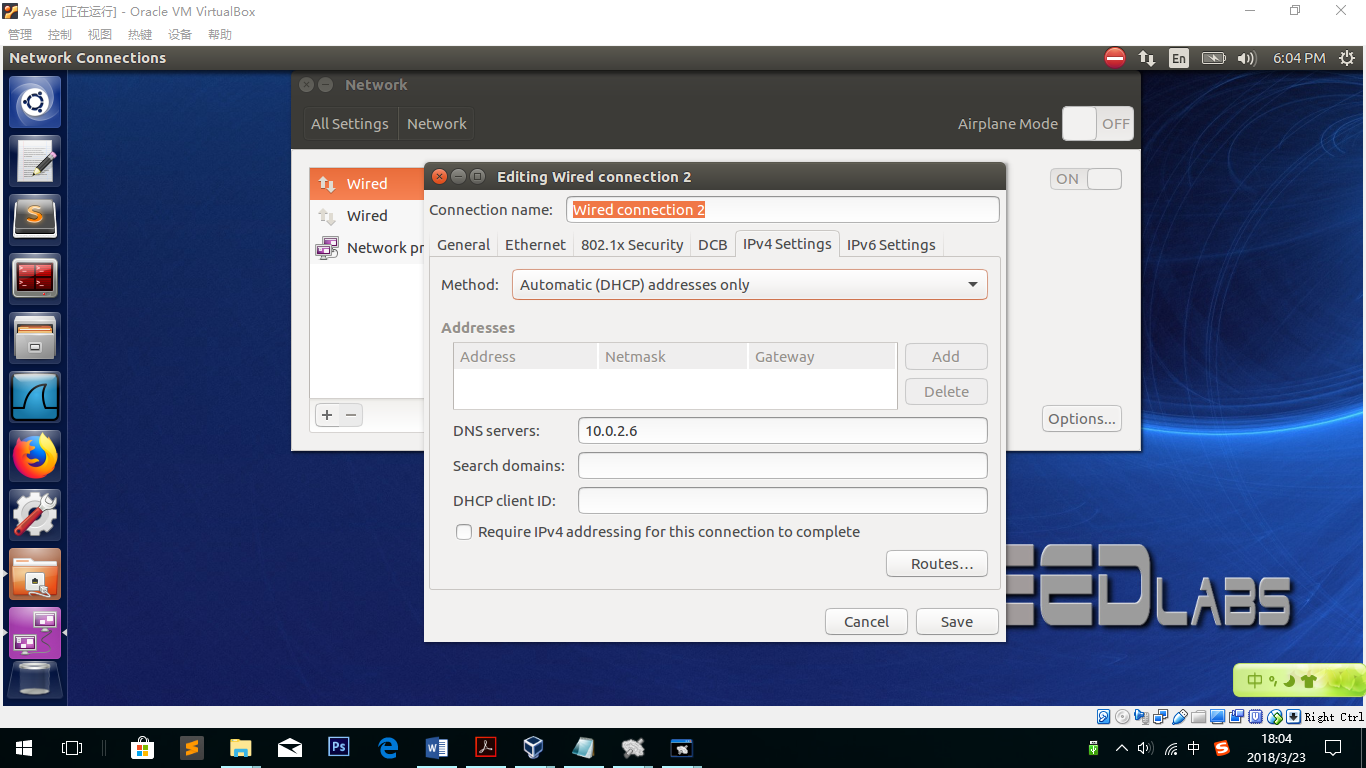
Scanning hosts file is the first step when the browser receive the name but not ip address. If the name is found in hosts file, then the browser will visit the IP address directly.

**3.2 Task 2: Directly Spoof Response to User**

Firstly, I set up Machine C(10.0.2.6) as my DNS server.

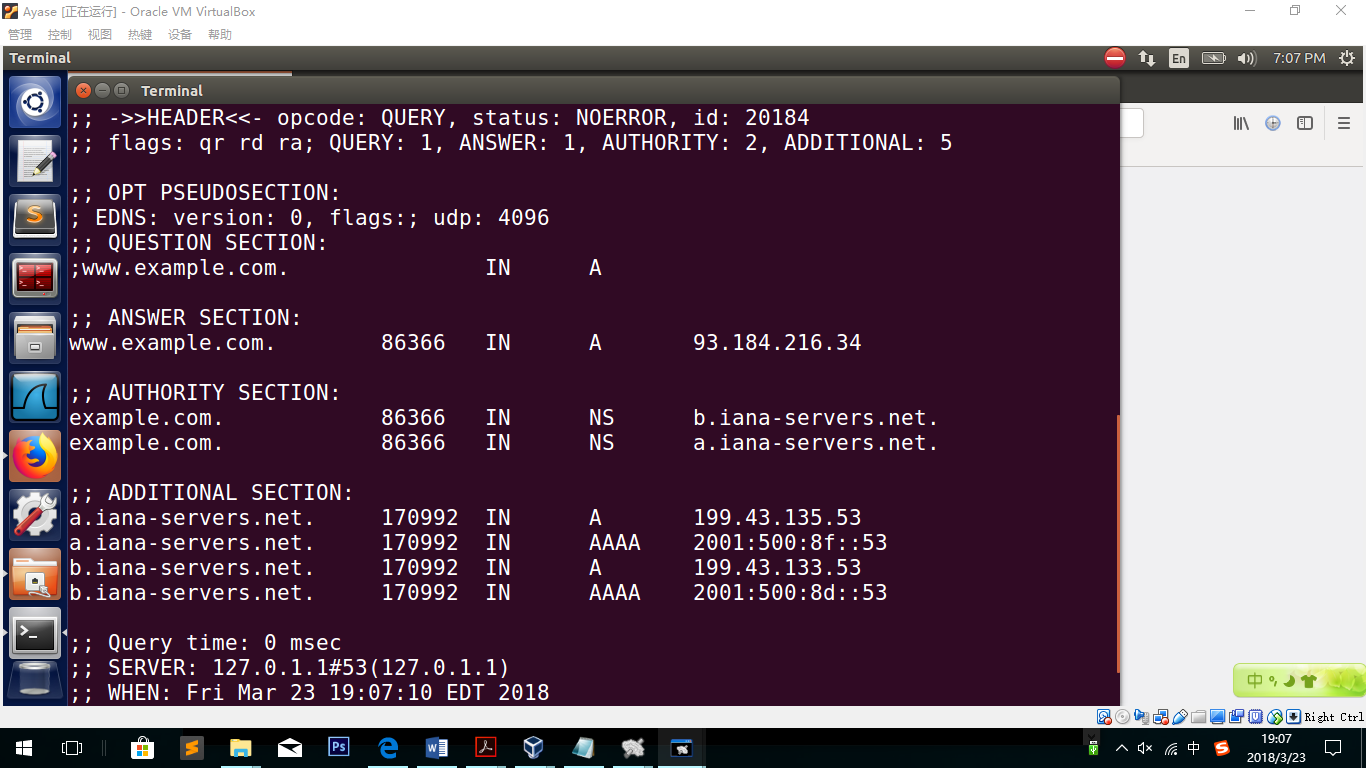


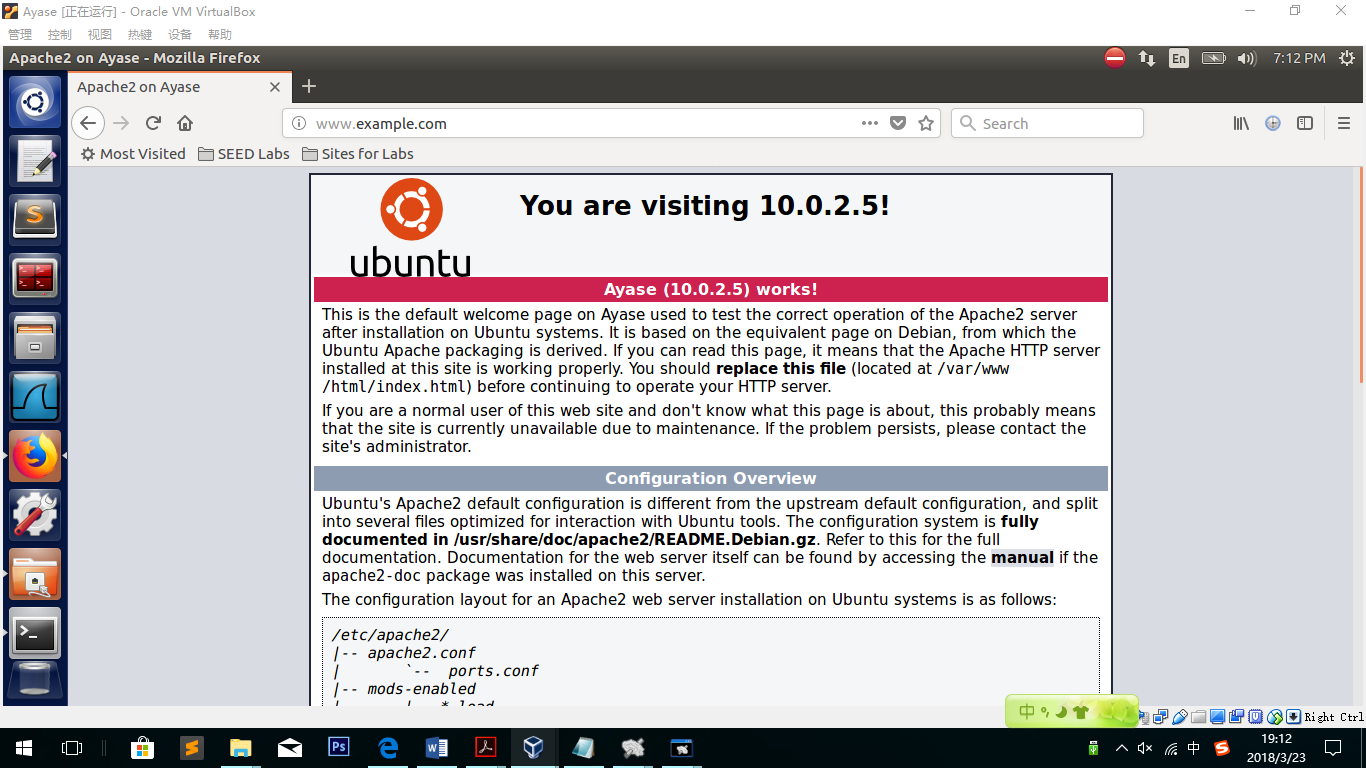
Settings in my User Machine:



By using this setting, 10.0.2.6 is set as the only one of Machine A’s DNS server.

Dig www.example.com:

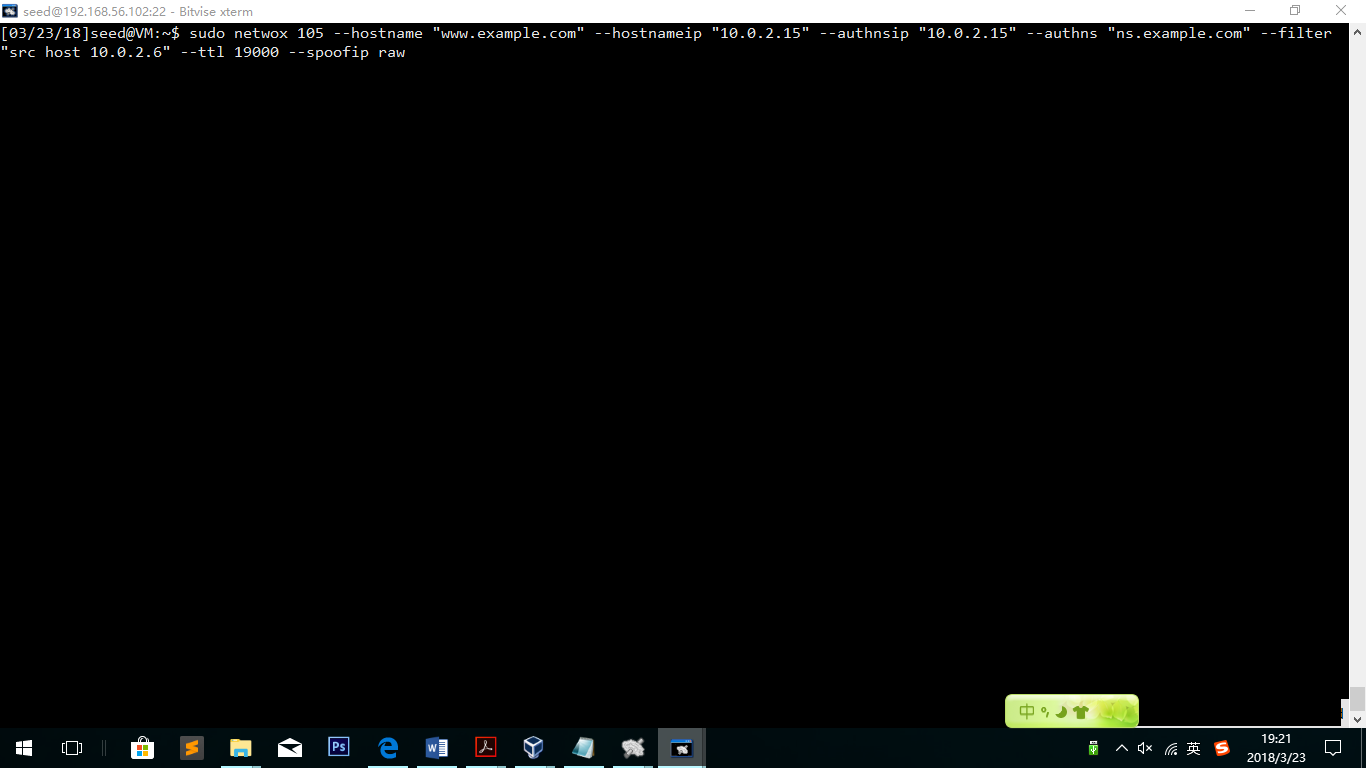




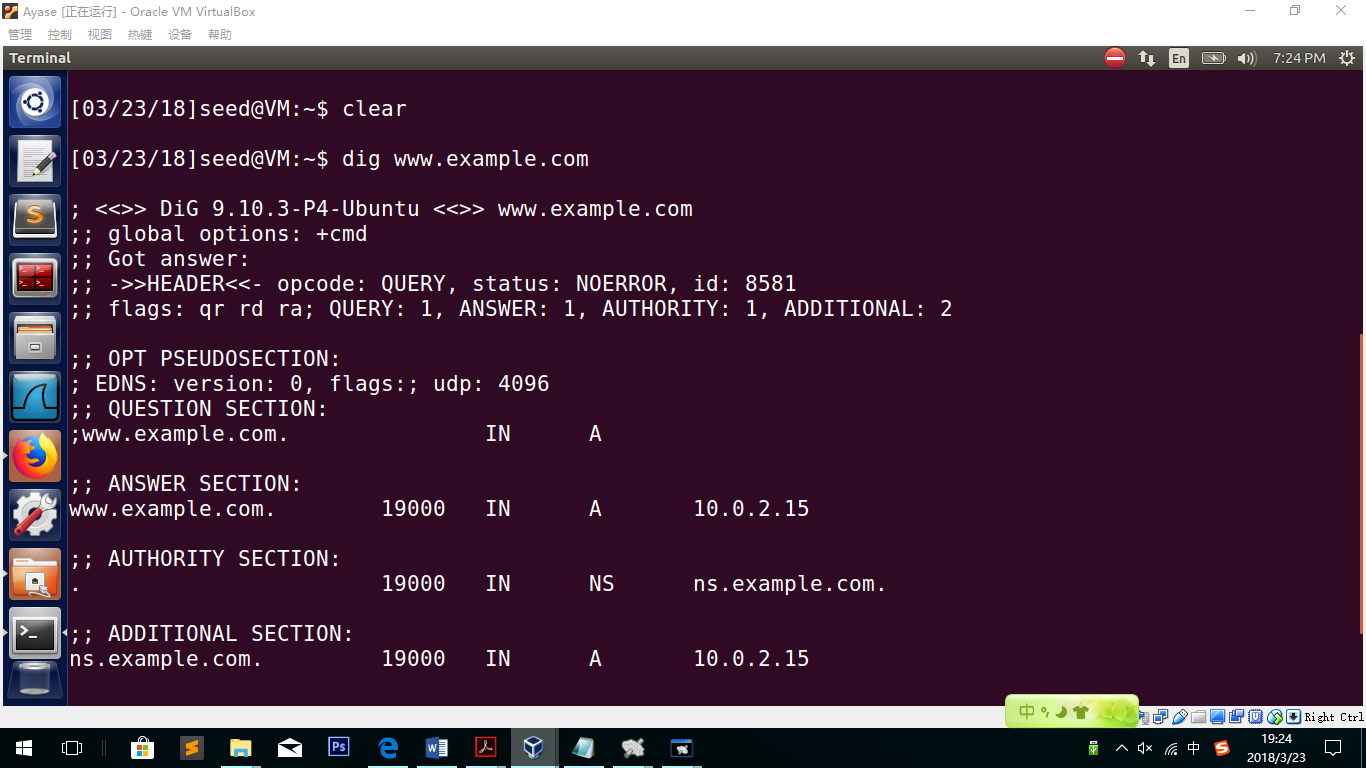
Explanation:

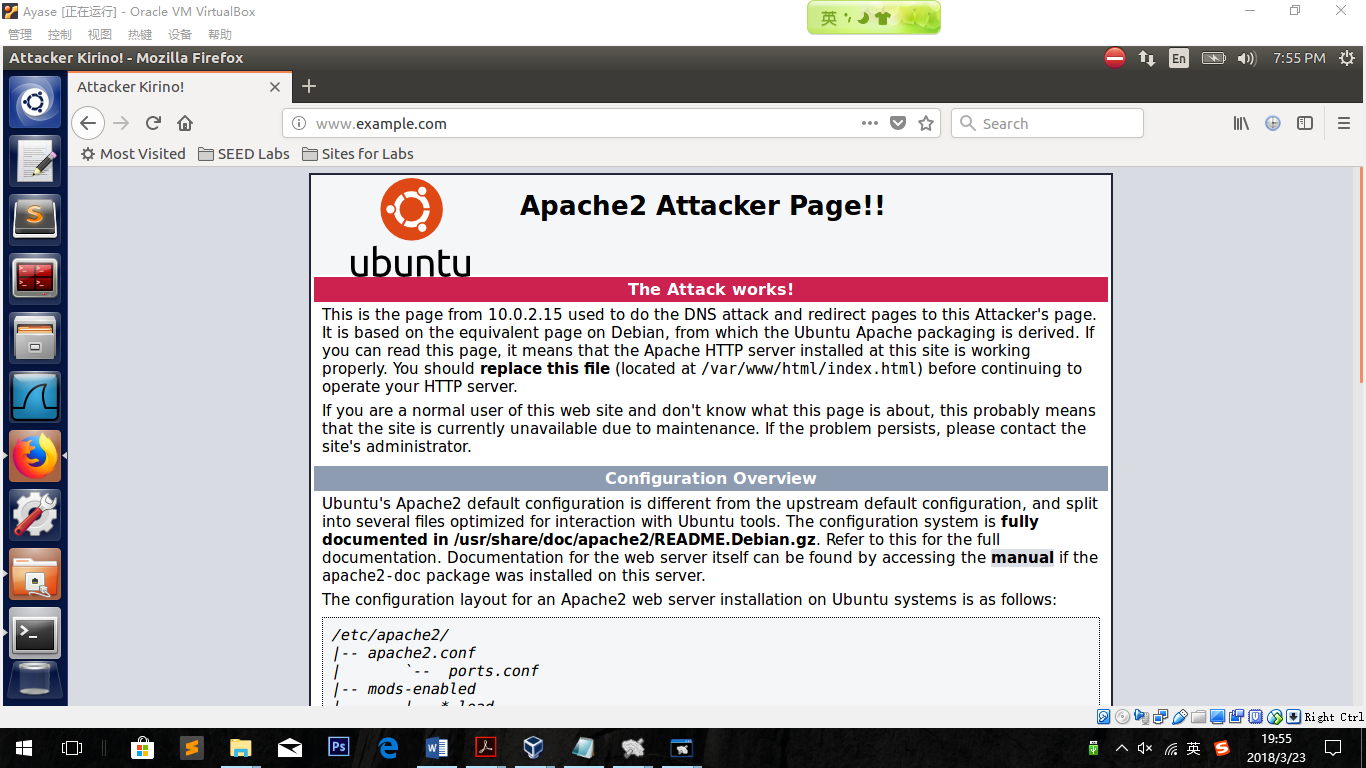
An example page is shown while visiting [www.example.com](http://www.example.com). This means that DNS server works as normal, and user can get normal response from DNS server. Visiting the website, the page has been redirected to its own address.

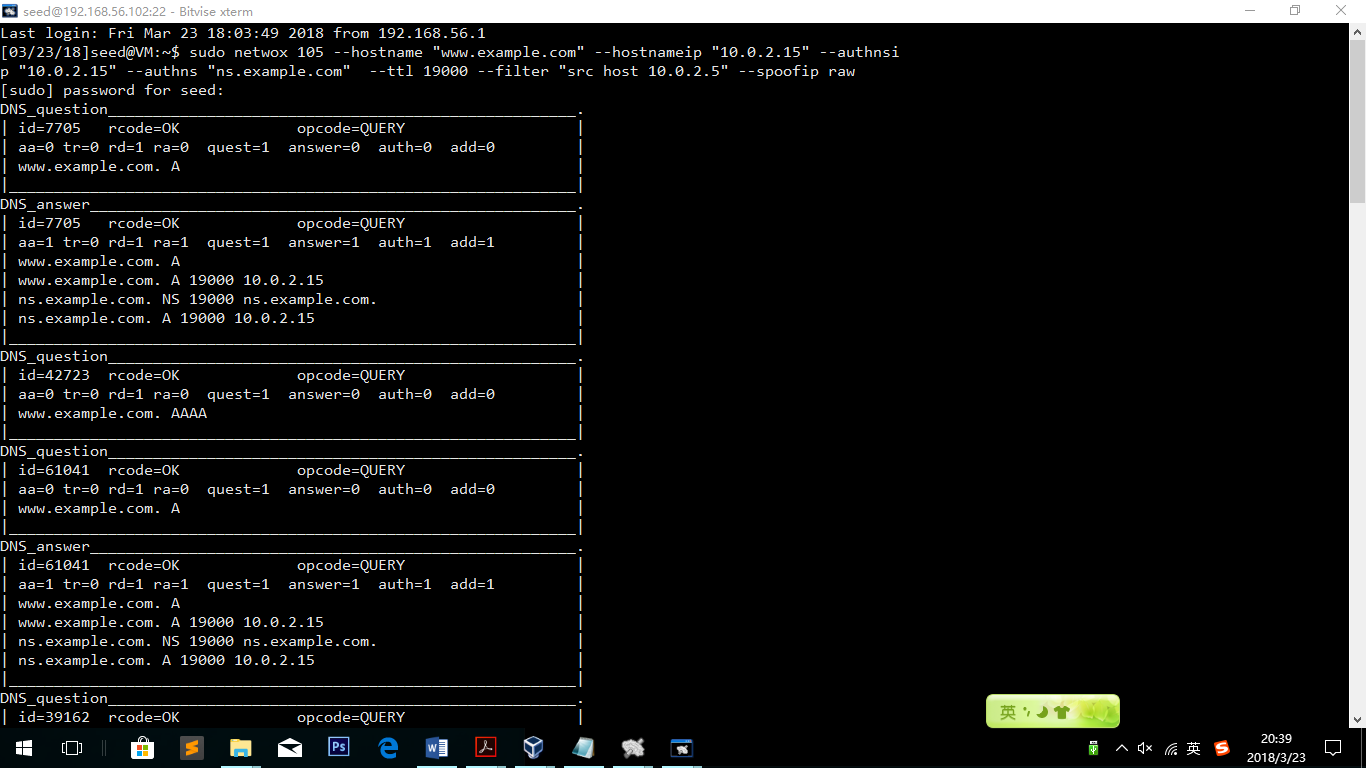
**Attack using netwox 105:**

*sudo netwox 105 --hostname "www.example.com" --hostnameip "10.0.2.15" --authnsip "10.0.2.15" --authns "ns.example.com" --filter "src host 10.0.2.5" --ttl 19000 --spoofip raw* 

Go back to Machine A and visit the website again:







Observation:

Obviously we succeed. Via my configuration, [www.example.com](http://www.example.com) should be directed to 10.0.2.5. However, it becomes the attacker’s homepage successfully.

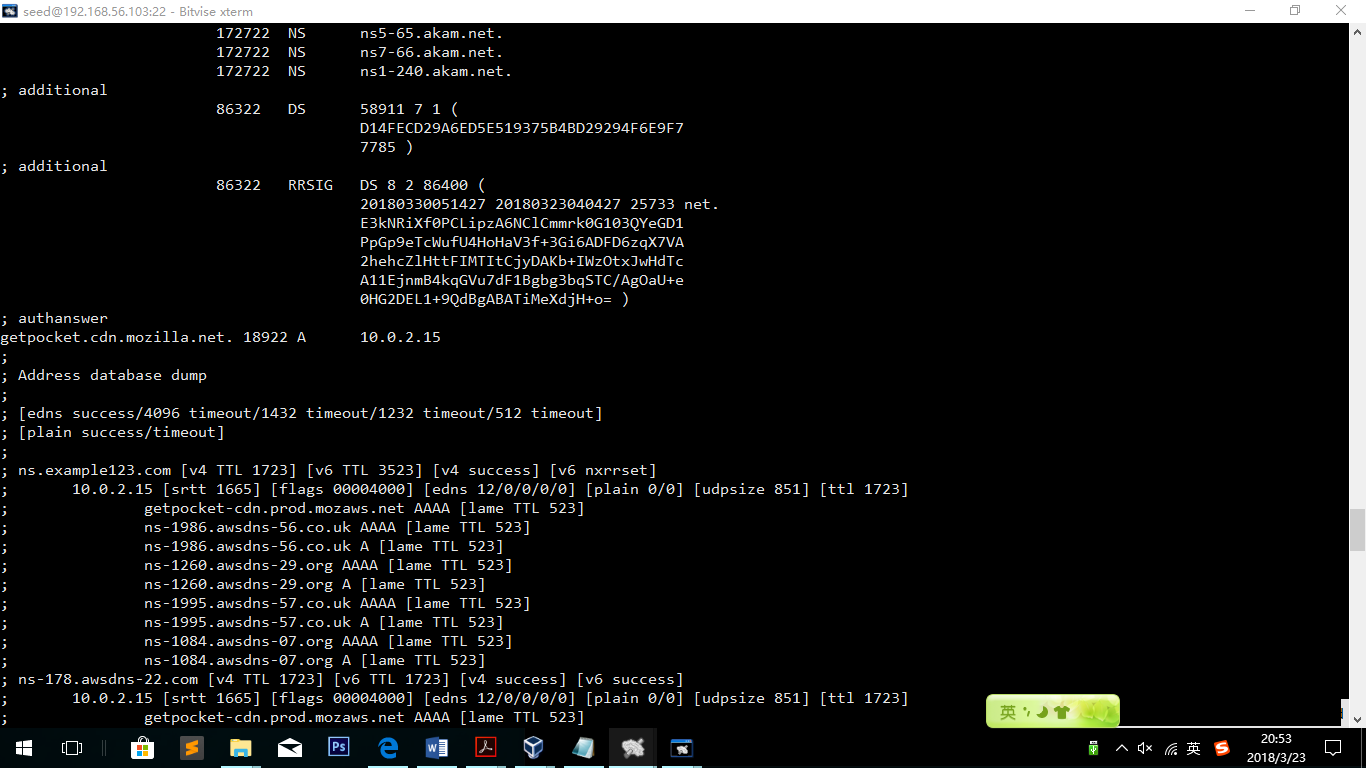
Explanation:

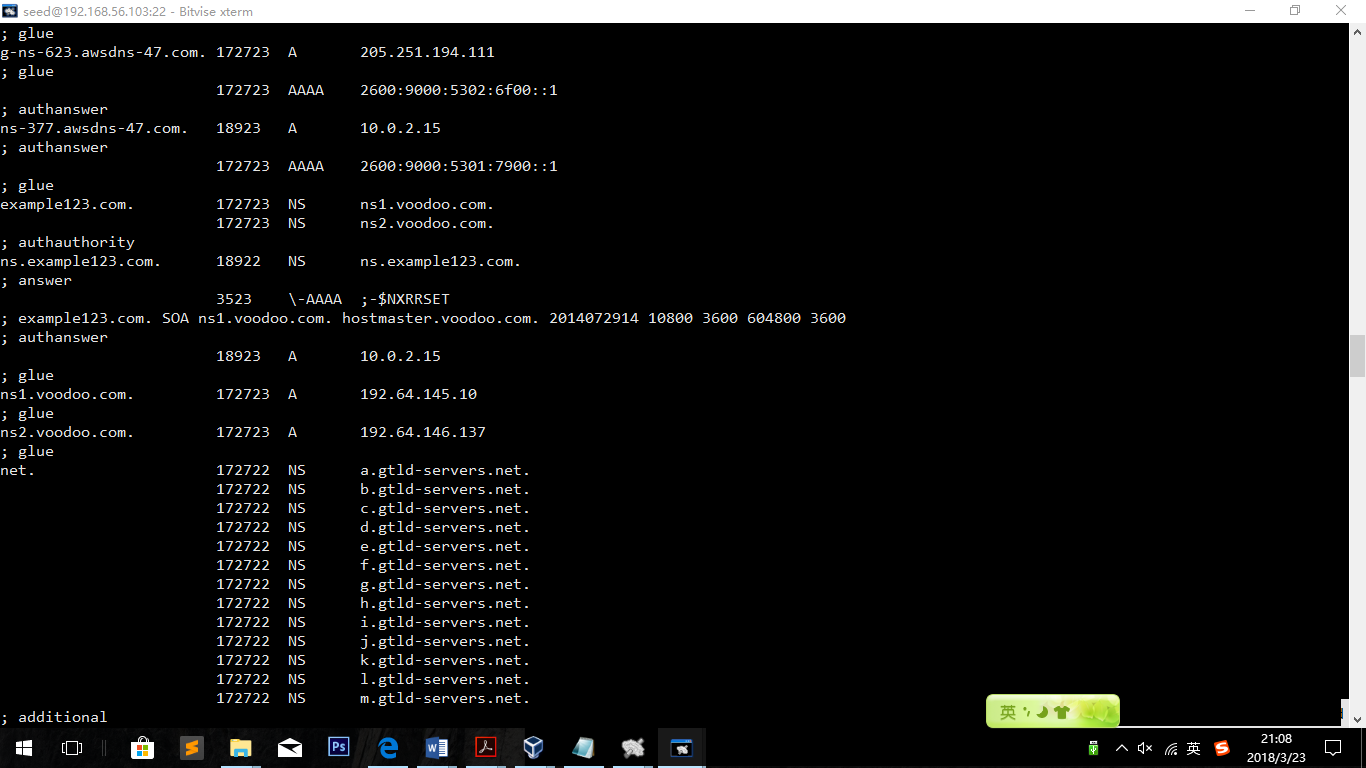
In this task, we did not attack the DNS server but spoofed a fake answer message to the user. After the user receive the fake message, the 8 conditions are finished by netwox 105, so the user believes that 10.0.2.15 is the target ip for [www.example.com](http://www.example.com) .

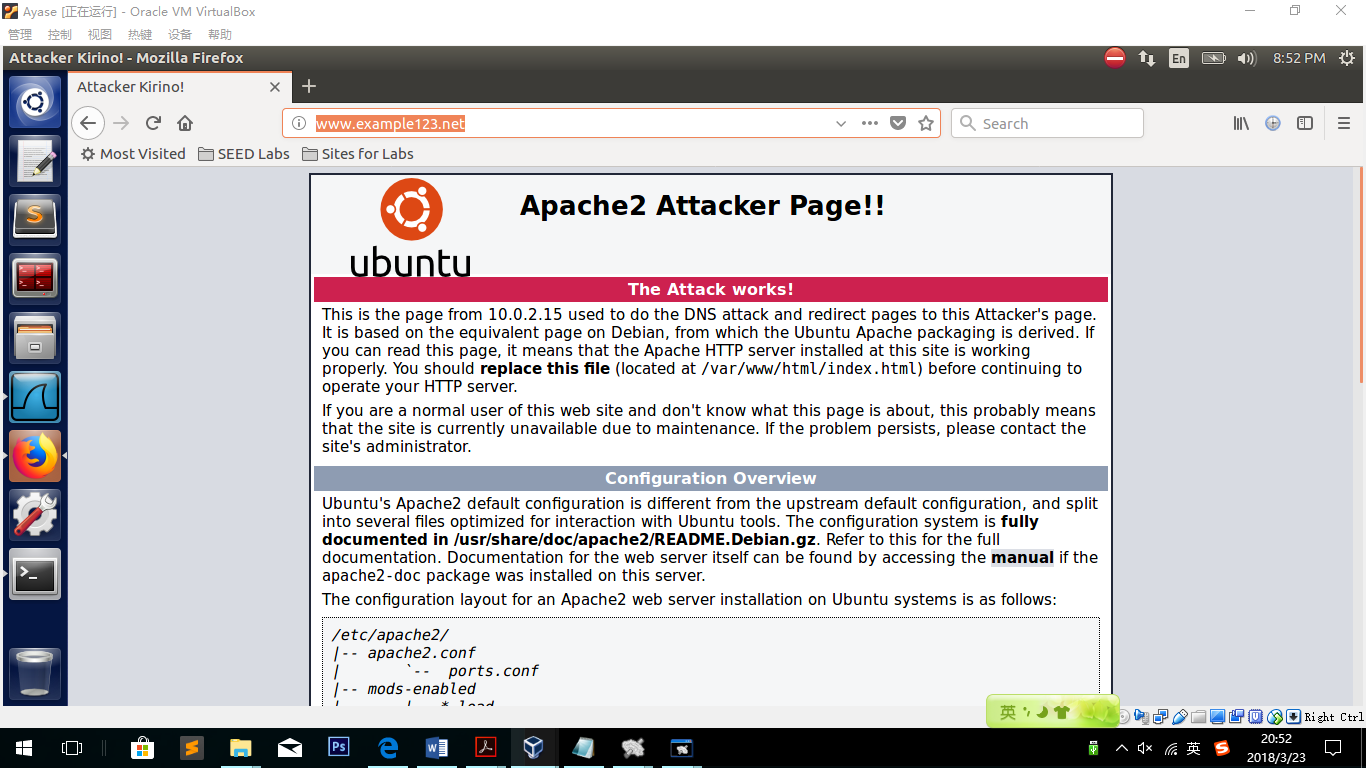
**3.3 Task 3: DNS Server Cache Poisoning**

*sudo netwox 105 --hostname "www.example123.net" --hostnameip "10.0.2.15" --authnsip "10.0.2.15" --authns "ns.example123.net" --filter "src host 10.0.2.6" --ttl 19000 --spoofip raw*

***Run on Attacker’s Machine***







**Observation:**

As you can see, the new address [www.example123.net](http://www.example123.net) is redirected to the attacker’s field. And in the server’s cache, we can see the example123.net is saved.

**Explanation:**

We spoofed a fake answer of [www.example123.net](http://www.example123.net). While the User started to ask the DNS server for answer, NETWOX 105 will not response because of its filter” src host 10.0.2.6”. However, as long as DNS server receive the question, it does not know the answer either, so it started to ask if anyone knows the address of [www.example123.net](http://www.example123.net) . That is when the netwox started to work. It sent fake message to DNS server and DNS server saved the fake information for future use. That is how the DNS server was poisoned.