Penetration Testing

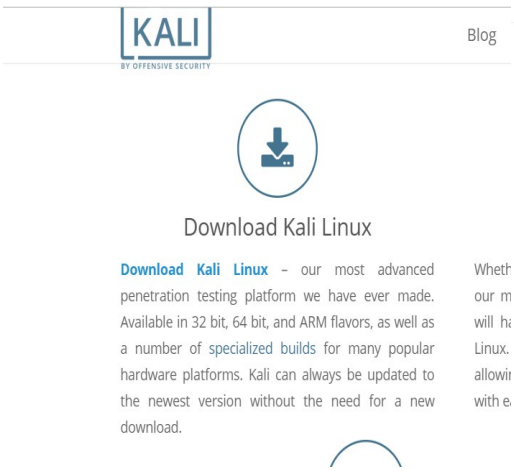
Team member name:
- Jin Zhang
- Jingyi Cui
- Yanjuan Li

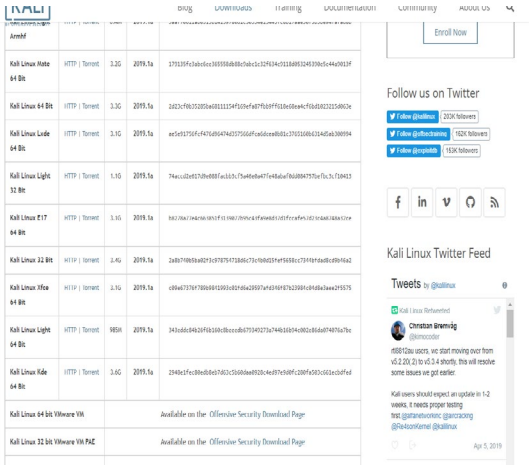
Goal:
- Identify and test your application against at least 3 attack vectors that do not exploit UI vulnerabilities. - You will document your findings in a PDF (Google doc exported as PDF) and commit it to your Github repository. - Your report should be as detailed as possible.
- You will document attacks on your own web application with and without the AWS WAF in place.

WorkFlow:
- Software Installation
- Overall scan
- SQL
- SSL
- DNS

So, I am using KALI. Pretty Advanced Penetration Testing Tools package. It integrates many very good penetration testing tools into one system for ease of use.

Download it at https://www.kali.org/





Kali Linux Kde 64 Bit	HTTP | Torrent	3.6G	**2019.1a**	2948e1fec80edb8eb7d63c5b60daa0928c4ed97e9d0fc280fa503c661ecbdfed
Kali Linux 64 bit VMware VM	Available on the Offensive Security Download Page			
Kali Linux 32 bit VMware VM PAE	Available on the Offensive Security Download Page			
Kali Linux 64 bit Vbox	Available on the Offensive Security Download Page			
Kali Linux 32 bit Vbox	Available on the Offensive Security Download Page			

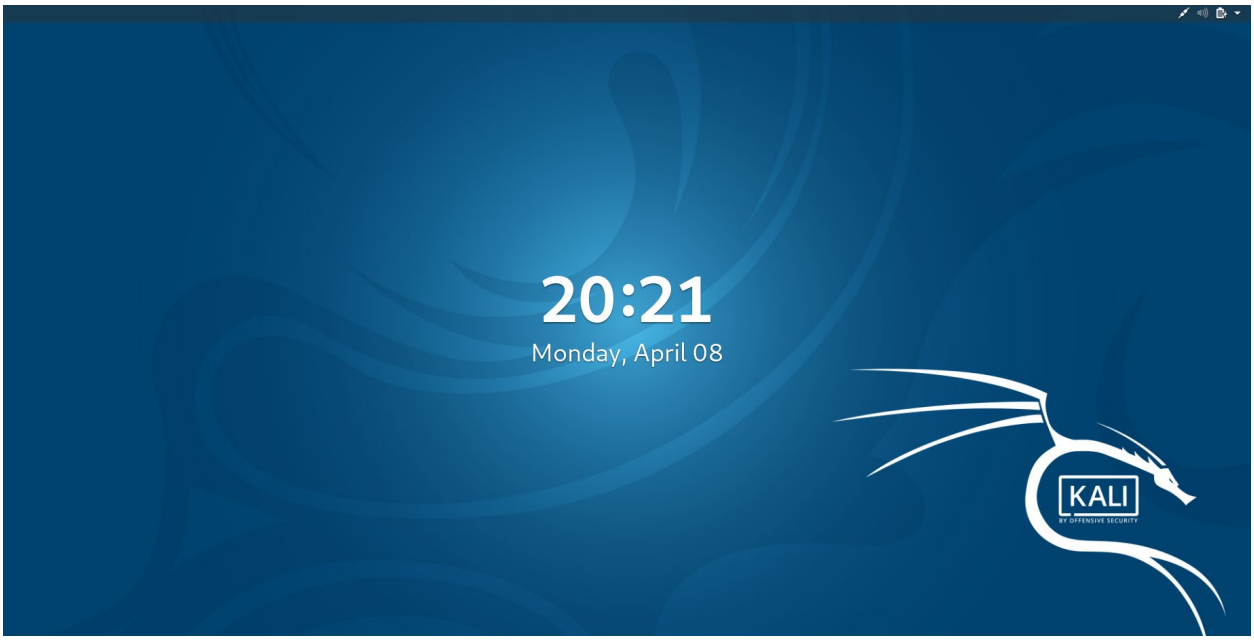
Kali Linux Weekly Builds

rate limited to 5 concurrent connections.

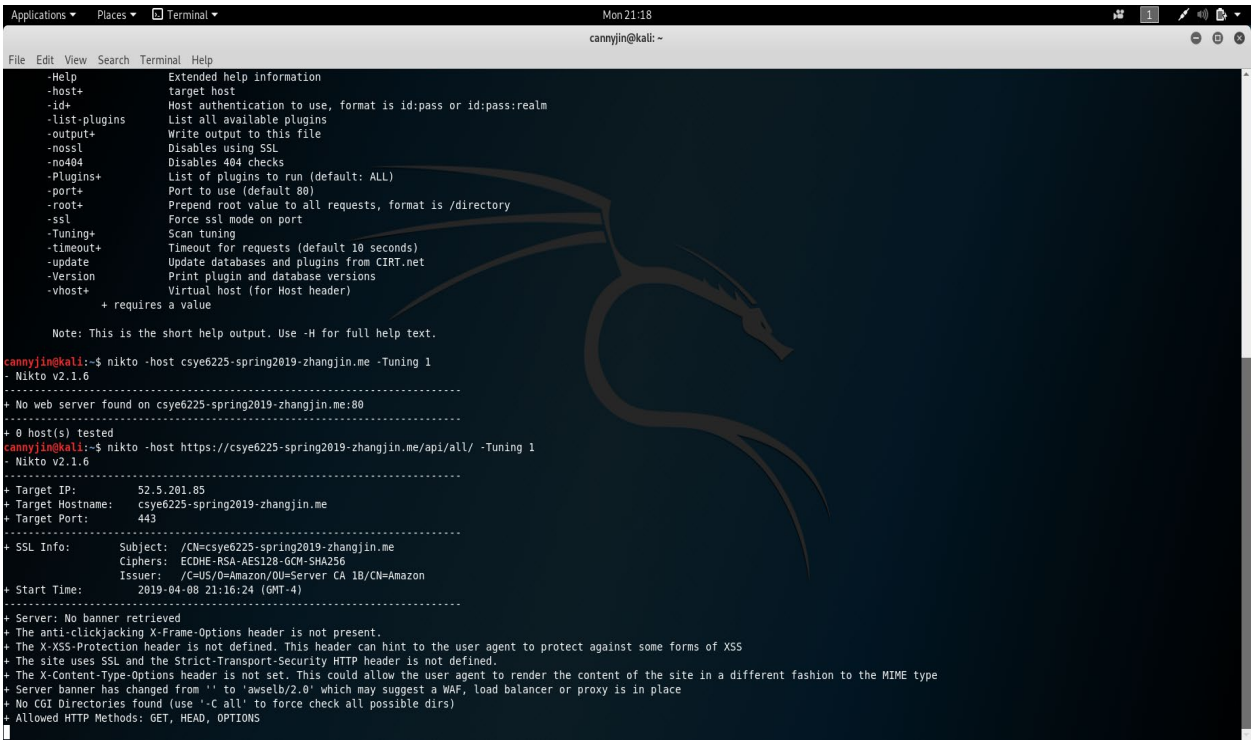
Kali Linux VMware Images | Kali Linux VirtualBox Images

Image Name	Torrent	Size	Version	SHA256Sum
Kali Linux Vm 64 Bit 7z	Torrent	2.5G	2019.1	e4c6999edccf27f97d4d014cdc66950b8b4148948abe8bb3a2c30bbc0915e95a
Kali Linux Vm 32 Bit 7z	Torrent	2.6G	2019.1	9d2c51b99da583c18fcdb3a47d001dbe1a9ed3013105d0557ed2594d033ce614

First for the very most, to install KALI Image. There are many editions of this platform. I choose Kali Linux 64 bit Vbox. This is like a VM with Linux OS.



First, I want to use Nikto: Nikto is an Open Source (GPL) web server scanner which performs comprehensive tests against web servers for multiple items. This is a multifunction powerful tool.



I want to use this tool to scan my app's vulnerable spot. To make a big picture, not to specify choose a vector.

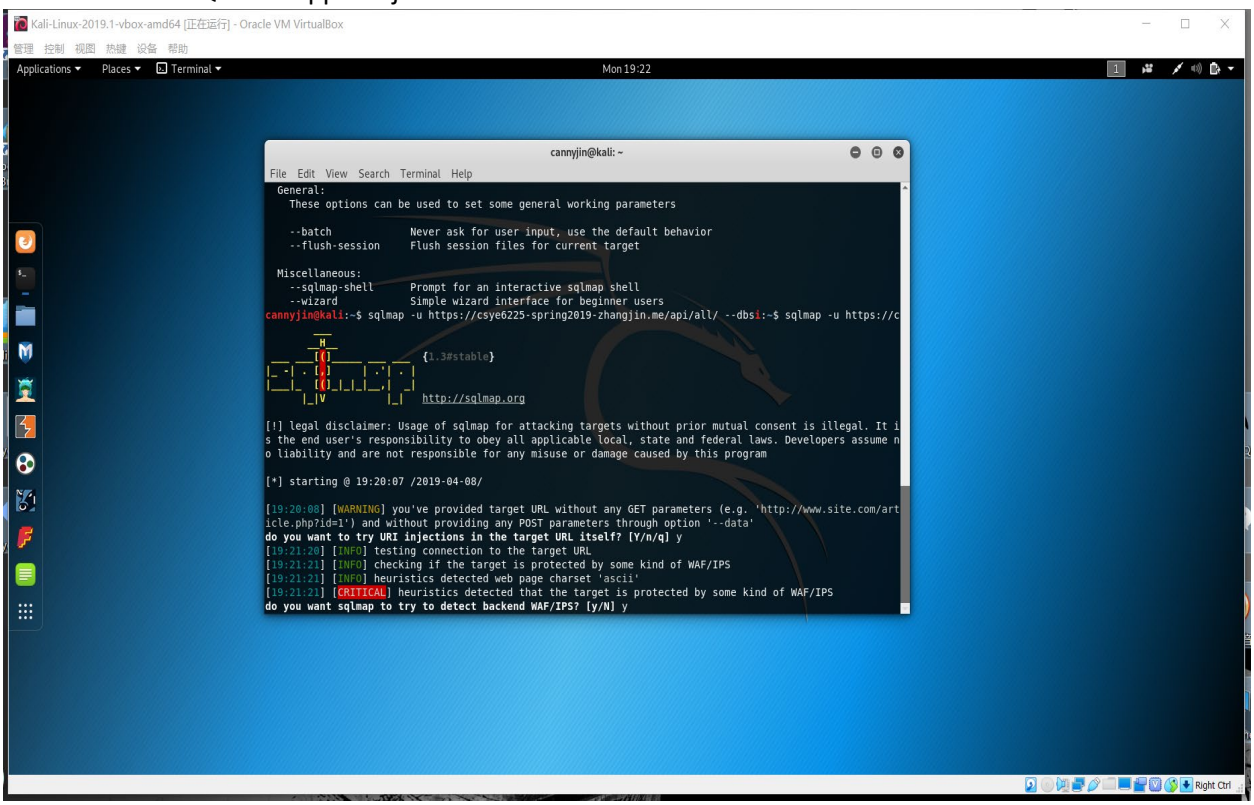
Because in my perspective the most critical step of the penetration test is to scan the vulnerable spot. If you don't know that. How can you know where to attack. How do I choose a specific vector?

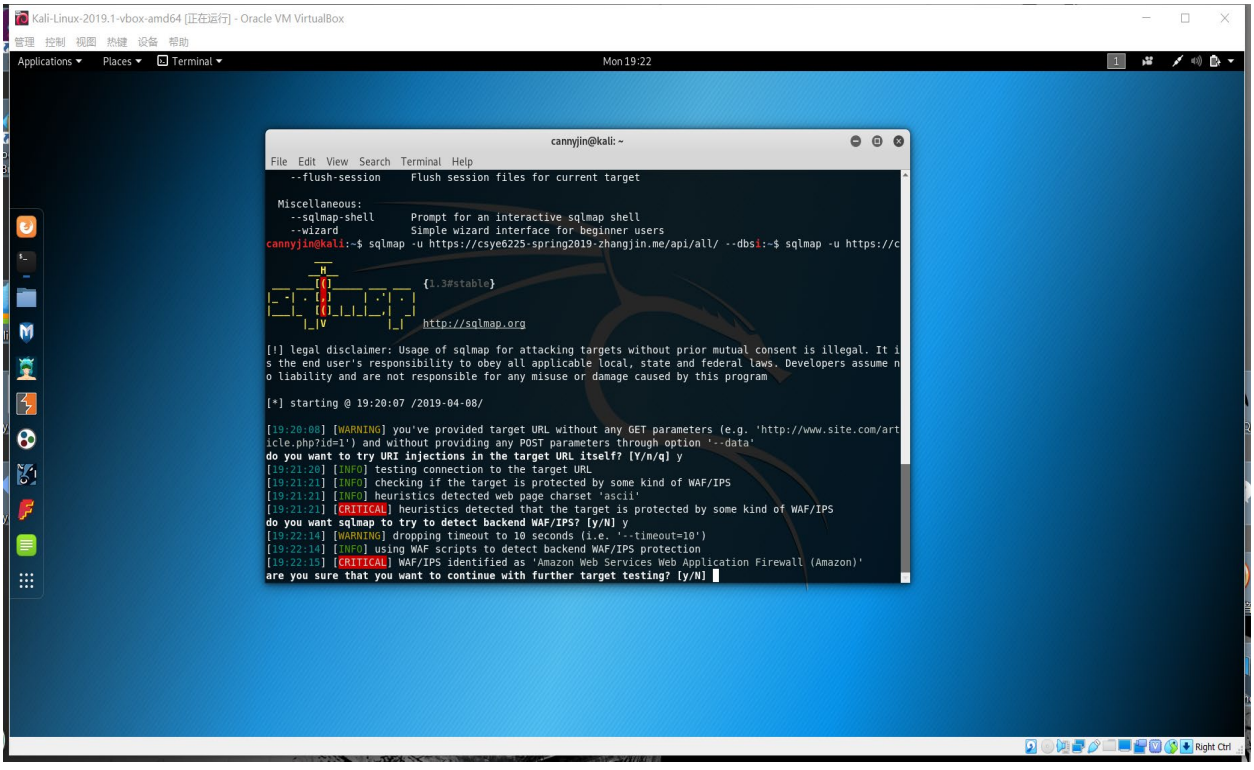
The result literally gives me the answer.

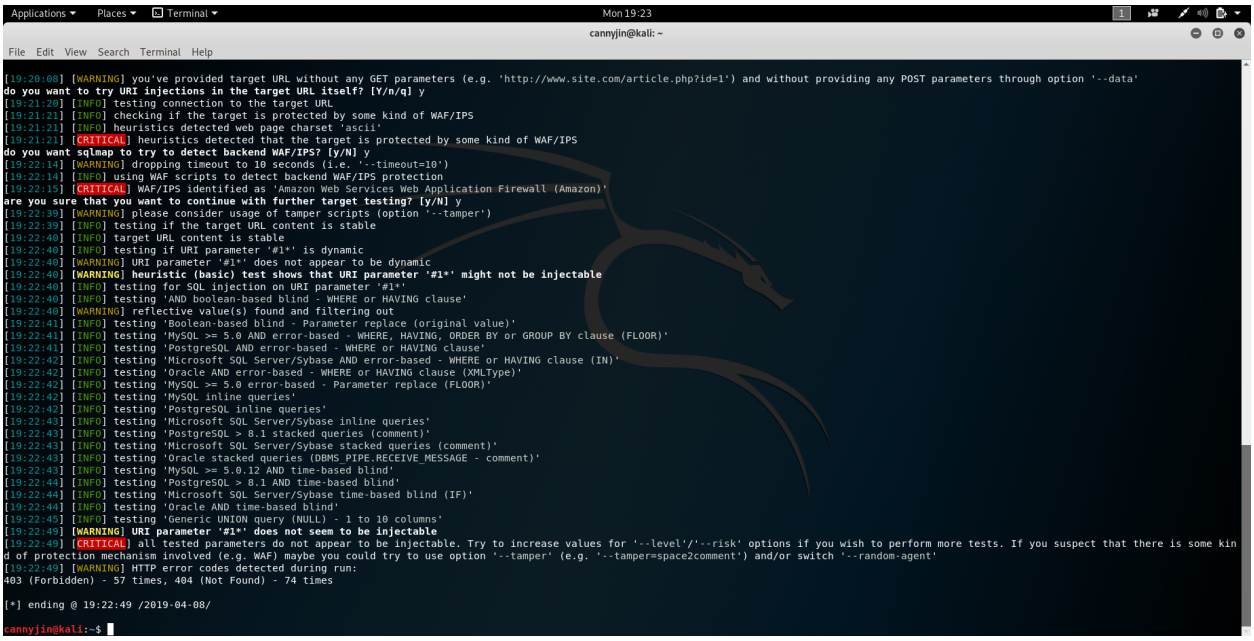
I would like to scan SQL and SSL and DNS.

SQLMAP to test the vulnerability of the database of our web application.

It can scan the SQL. See App is injectable or not.







I try to URI injections in the target URL which is my domain. https://csye6225-spring2019-zhangjin.me/api/all/

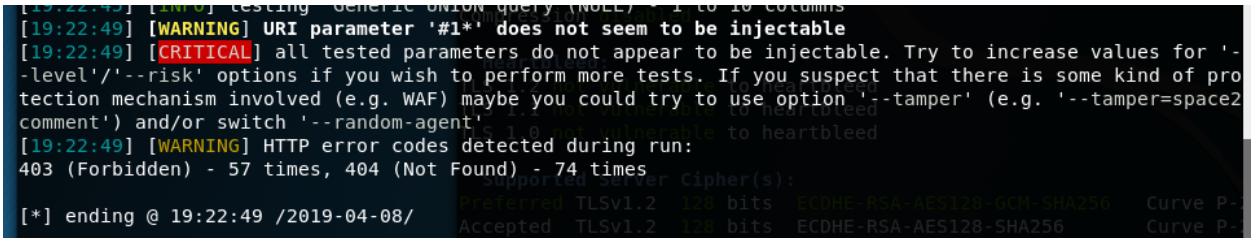
And is said [CRITICAL] heuristics detected that the target is protected by some kind of WAF/IPS

So, I try to detect backend WAF/IPS.

So, It identified as 'Amazon Web Services Web Application Firewall (Amazon)'

So, I go further target testing?

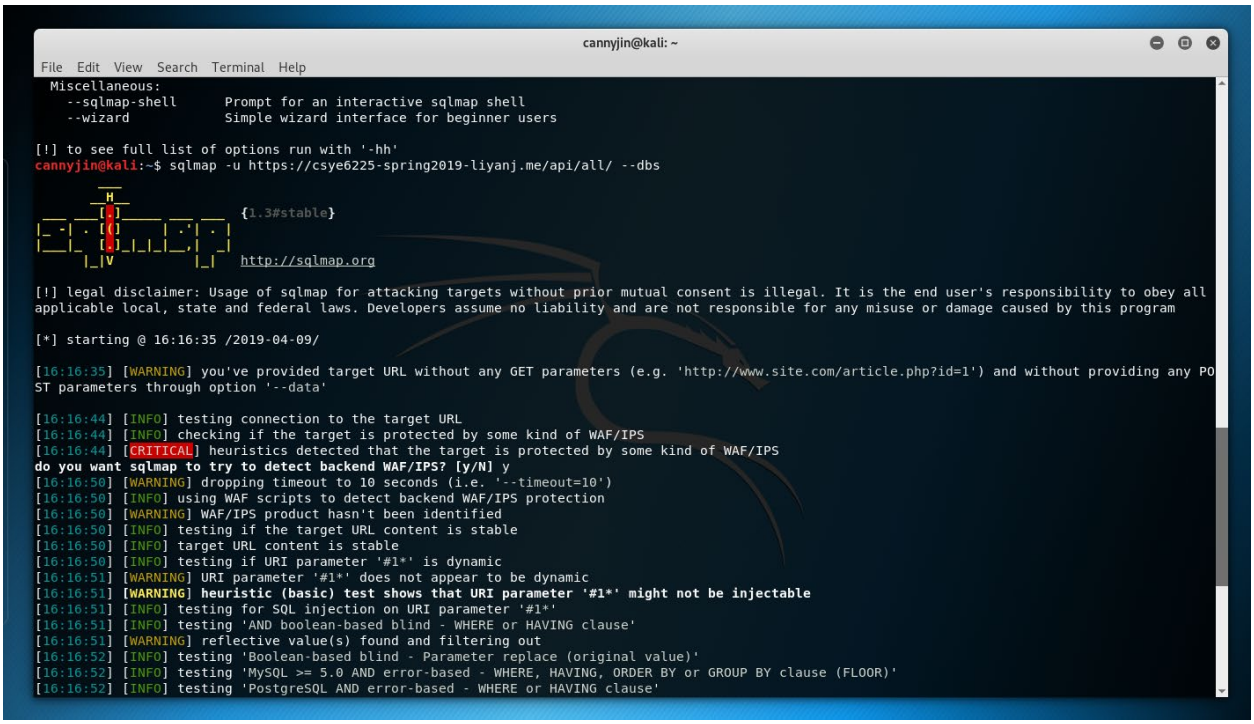
You can see the result in the screen shot.

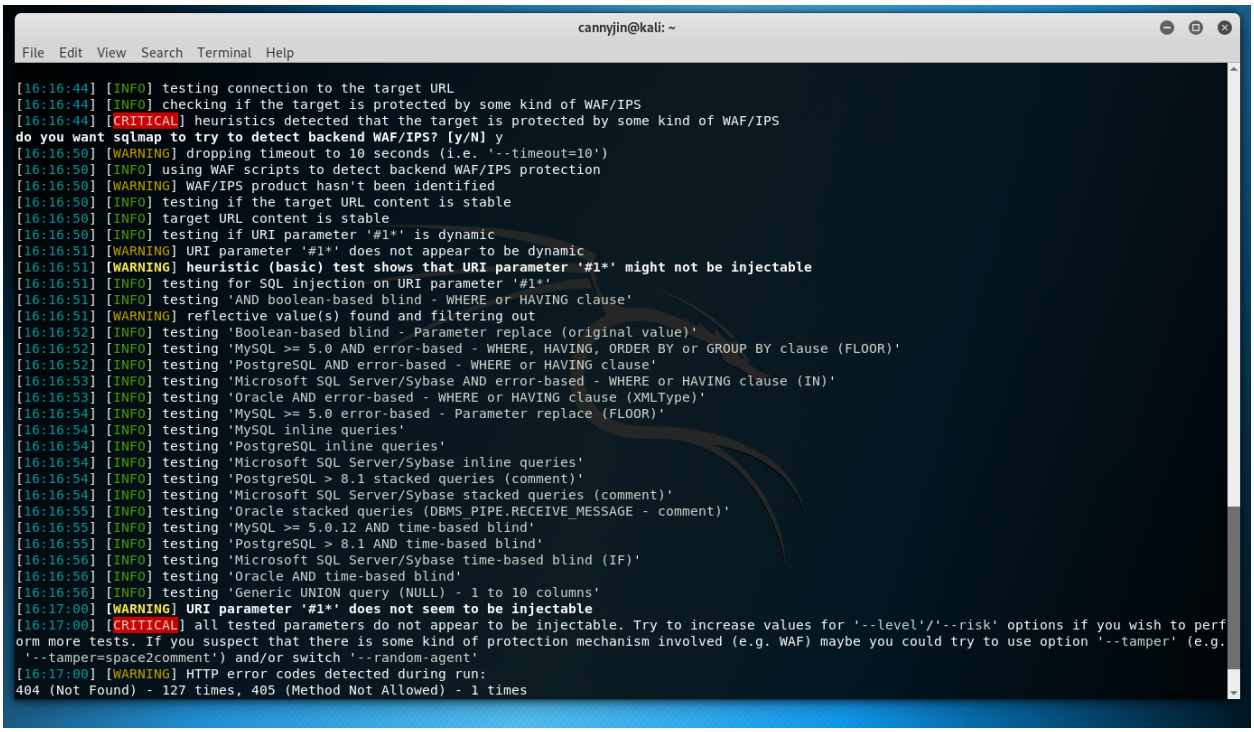


Look like our firewall work well.

Next, I will try my teammates Application that without firewall.

Now， try my teammamber’s domain that without firewall.



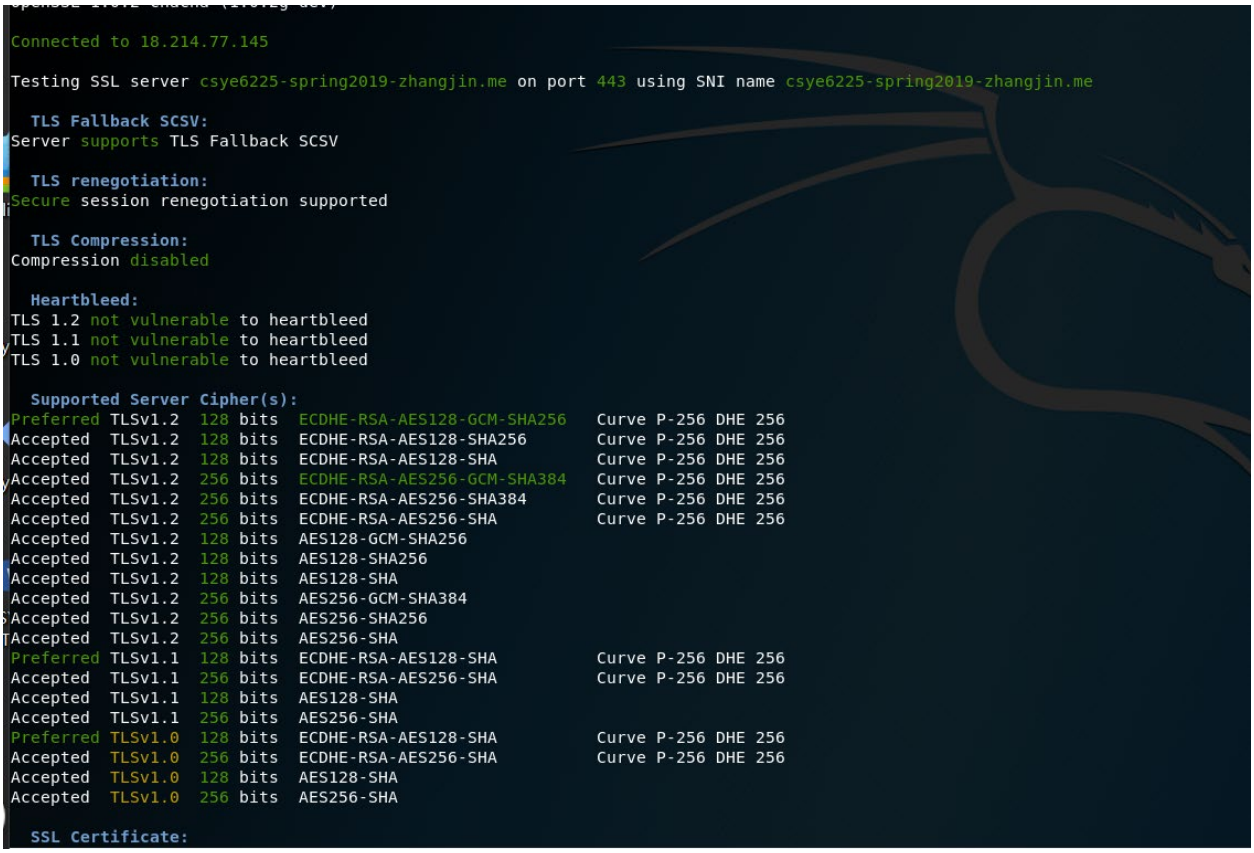


And the result seems to be the same.

SSLScan This queries SSL/TLS services, such as HTTPS, in order to determine the ciphers that are supported.

I want to see our cipher is secure enough or not.

```
                _
 ___ ___| |___ ___ ___ _ _ _
|_ -|_ -| |_ -|  _|  _| | | |
|___|___|_|___|___|___|_  |
                      |___|

                1.11.12-static
                OpenSSL 1.0.2-chacha (1.0.2g-dev)
Command:
  sslscan [Options] [host:port | host]

Options:
  --targets=<file>      A file containing a list of hosts to check.
                        Hosts can  be supplied  with ports (host:port)
  --sni-name=<name>     Hostname for SNI
  --ipv4, -4            Only use IPv4
  --ipv6, -6            Only use IPv6
  --show-certificate    Show full certificate information
  --no-check-certificate  Don't warn about weak certificate algorithm or keys
  --show-client-cas     Show trusted CAs for TLS client auth
  --show-ciphers        Show supported client ciphers
  --show-cipher-ids     Show cipher ids
  --show-times          Show handhake times in milliseconds
  --ssl2                Only check SSLv2 ciphers
  --ssl3                Only check SSLv3 ciphers
  --tls10               Only check TLSv1.0 ciphers
  --tls11               Only check TLSv1.1 ciphers
  --tls12               Only check TLSv1.2 ciphers
  --tlsall              Only check TLS ciphers (all versions)
  --ocsp                Request OCSP response from server
  --pk=<file>           A file containing the private key or a PKCS#12 file
                        containing a private key/certificate pair
  --pkpass=<password>   The password for the private  key or PKCS#12 file
  --certs=<file>        A file containing PEM/ASN1 formatted client certificates
  --no-ciphersuites     Do not check for supported ciphersuites
  --no-fallback         Do not check for TLS Fallback SCSV
  --no-renegotiation    Do not check for TLS renegotiation
  --no-compression      Do not check for TLS compression (CRIME)
  --no-heartbleed       Do not check for OpenSSL Heartbleed (CVE-2014-0160)
  --starttls-ftp        STARTTLS setup for FTP
  --starttls-imap       STARTTLS setup for IMAP
  --starttls-irc        STARTTLS setup for IRC
  --starttls-ldap       STARTTLS setup for LDAP
  --starttls-pop3       STARTTLS setup for POP3
  --starttls-smtp       STARTTLS setup for SMTP
```

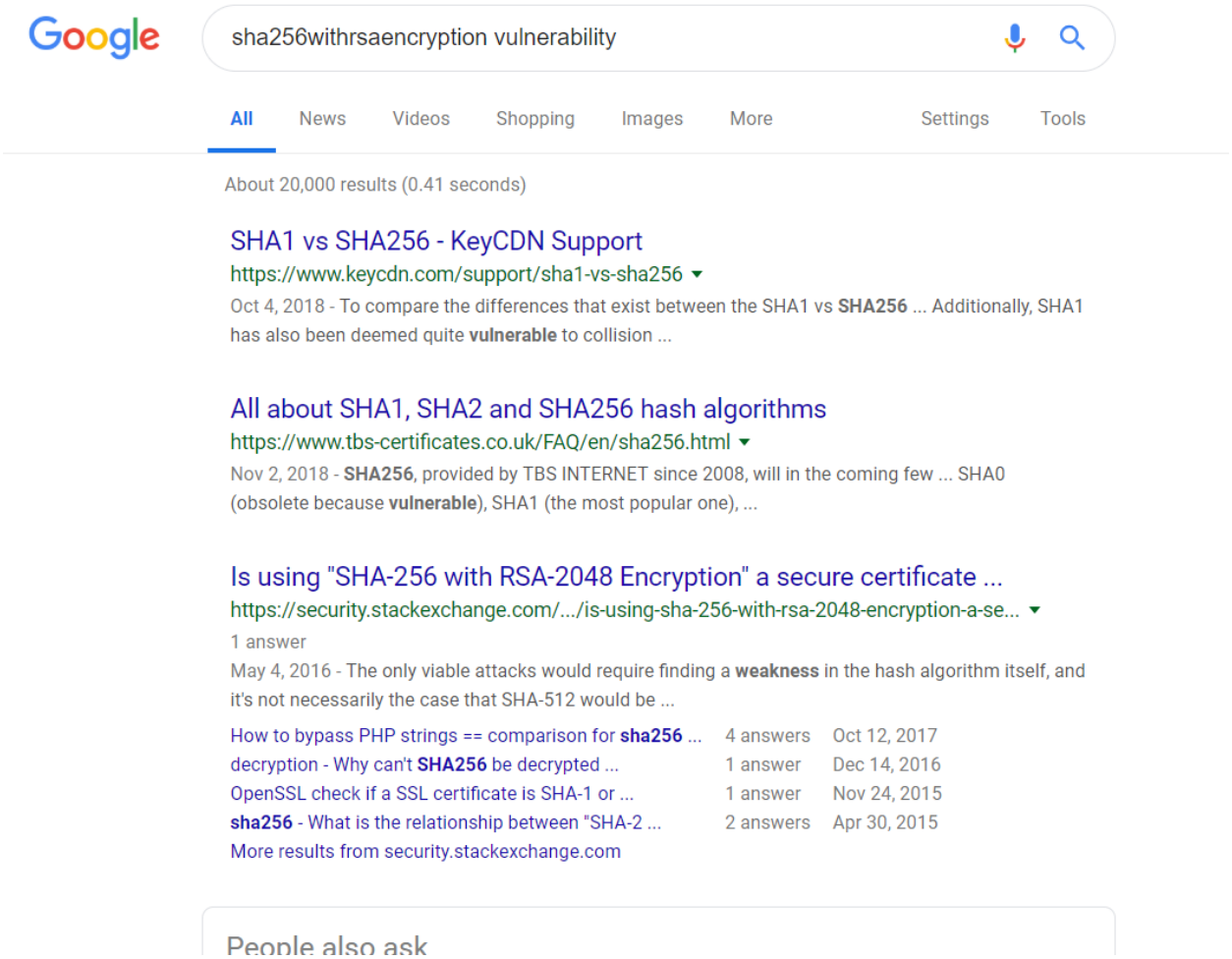


```
Connected to 18.214.77.145

Testing SSL server csye6225-spring2019-zhangjin.me on port 443 using SNI name csye6225-spring2019-zhangjin.me

  TLS Fallback SCSV:
Server supports TLS Fallback SCSV

  TLS renegotiation:
Secure session renegotiation supported

  TLS Compression:
Compression disabled

  Heartbleed:
TLS 1.2 not vulnerable to heartbleed
TLS 1.1 not vulnerable to heartbleed
TLS 1.0 not vulnerable to heartbleed

  Supported Server Cipher(s):
Preferred TLSv1.2  128 bits  ECDHE-RSA-AES128-GCM-SHA256   Curve P-256 DHE 256
Accepted  TLSv1.2  128 bits  ECDHE-RSA-AES128-SHA256       Curve P-256 DHE 256
Accepted  TLSv1.2  128 bits  ECDHE-RSA-AES128-SHA          Curve P-256 DHE 256
Accepted  TLSv1.2  256 bits  ECDHE-RSA-AES256-GCM-SHA384   Curve P-256 DHE 256
Accepted  TLSv1.2  256 bits  ECDHE-RSA-AES256-SHA384       Curve P-256 DHE 256
Accepted  TLSv1.2  256 bits  ECDHE-RSA-AES256-SHA          Curve P-256 DHE 256
Accepted  TLSv1.2  128 bits  AES128-GCM-SHA256
Accepted  TLSv1.2  128 bits  AES128-SHA256
Accepted  TLSv1.2  128 bits  AES128-SHA
Accepted  TLSv1.2  256 bits  AES256-GCM-SHA384
Accepted  TLSv1.2  256 bits  AES256-SHA256
Accepted  TLSv1.2  256 bits  AES256-SHA
Preferred TLSv1.1  128 bits  ECDHE-RSA-AES128-SHA          Curve P-256 DHE 256
Accepted  TLSv1.1  256 bits  ECDHE-RSA-AES256-SHA          Curve P-256 DHE 256
Accepted  TLSv1.1  128 bits  AES128-SHA
Accepted  TLSv1.1  256 bits  AES256-SHA
Preferred TLSv1.0  128 bits  ECDHE-RSA-AES128-SHA          Curve P-256 DHE 256
Accepted  TLSv1.0  256 bits  ECDHE-RSA-AES256-SHA          Curve P-256 DHE 256
Accepted  TLSv1.0  128 bits  AES128-SHA
Accepted  TLSv1.0  256 bits  AES256-SHA

  SSL Certificate:
```

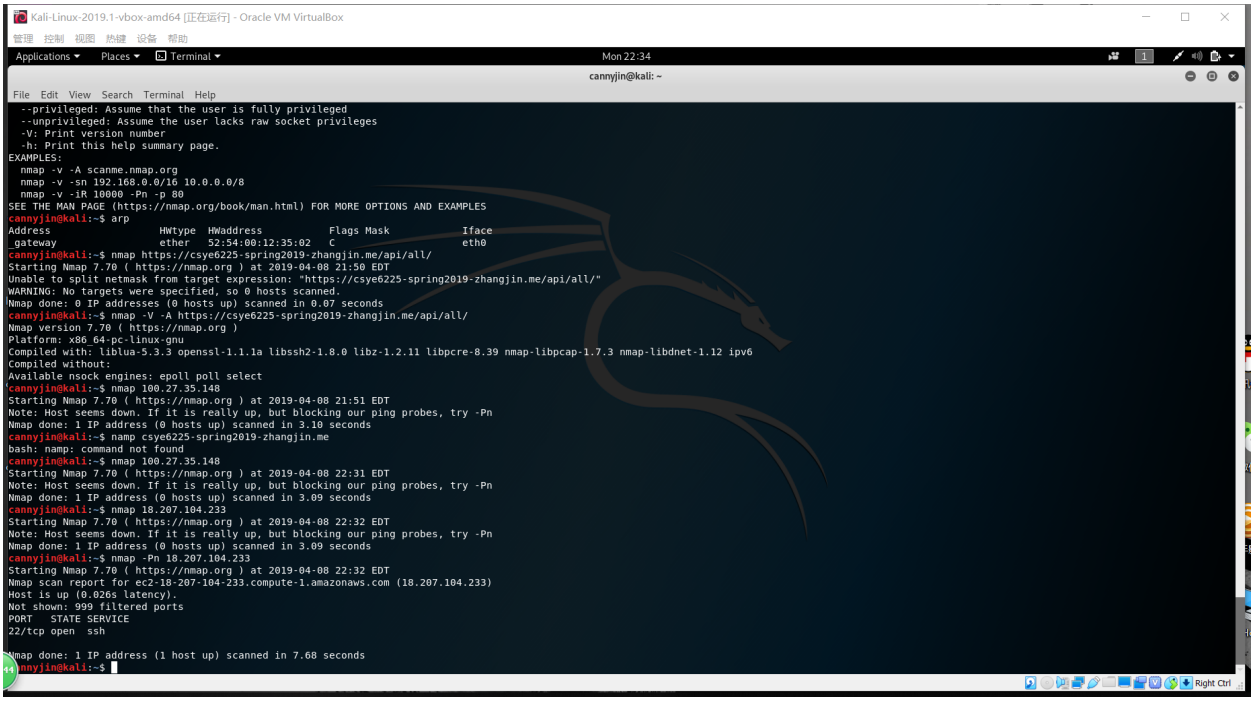
```
  SSL Certificate:
Signature Algorithm: sha256WithRSAEncryption
RSA Key Strength:    2048

Subject:  csye6225-spring2019-zhangjin.me
Altnames: DNS:csye6225-spring2019-zhangjin.me
Issuer:   Amazon

Not valid before: Apr  6 00:00:00 2019 GMT
Not valid after:  May  6 12:00:00 2020 GMT
```

As you can see I got a lot of results.

I got the cipher Algorithm, you can google it to check this sha256WithRSAEncryption Algorithm is secure or not.



Nmap I use it to discovery the network for our Web to see there is any security issue.



I was trying to get the network of my application.

I got nothing, as the screenshot show. First, I was confused. Maybe is the VPC, private cloud that makes me cannot be got the info of network.

But when I got this:

```
cannyjin@kali:~$ nmap 18.207.104.233
Starting Nmap 7.70 ( https://nmap.org ) at 2019-04-08 22:32 EDT
Note: Host seems down. If it is really up, but blocking our ping probes, try -Pn
Nmap done: 1 IP address (0 hosts up) scanned in 3.09 seconds
cannyjin@kali:~$ nmap -Pn 18.207.104.233
Starting Nmap 7.70 ( https://nmap.org ) at 2019-04-08 22:32 EDT
Nmap scan report for ec2-18-207-104-233.compute-1.amazonaws.com (18.207.104.233)
Host is up (0.026s latency).
Not shown: 999 filtered ports
PORT   STATE SERVICE
22/tcp open  ssh

Nmap done: 1 IP address (1 host up) scanned in 7.68 seconds
```

I realize is the SSH did this. So, looks like the network is pretty security I think.

I think SSL has nothing to do with firewall. I skip the section about testing the app without firewall.