Lab 3

By
Sakshi Kulkarni (smk8939@nyu.edu)
Chandana Srinivasa Yatisha (cs7074@nyu.edu)
Amrutha Patil (ap7982@nyu.edu)

Instructions:

1. Make a copy of this document.
2. Replace all instances of |____| with your actual responses. Both text and screenshots are fine.
3. Save your report as a PDF file.
4. Submit your report via GradeScope.

If you're unsure of any of the steps, check out the recording on Nov 2. I basically walked through the lab below.

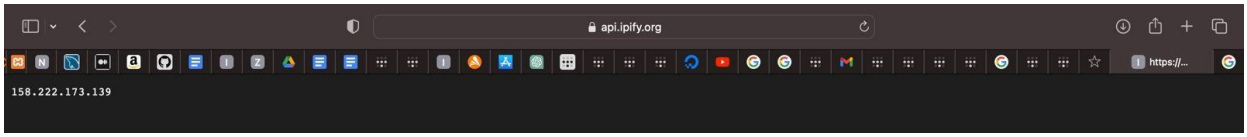
Set up Tailscale

Checking the IP address

On your computer's browser, go to https://api.ipify.org.

What is the output?

Answer:
158.222.173.139
It is the public IP address assigned to our router or modem by the internet service provider (ISP).



Is this IP address the same as the local IP address of your computer? Why or why not?

Answer:
No, this IP address is not same as the local IP address of our computer. The local IP address of our computer is 192.168.1.55.

The reason for the difference is that the public IP address (158.222.173.139) is assigned by the ISP and is the address through which our network communicates with the internet. Whereas, the local IP address (192.168.1.55) is assigned by DHCP and is used for internal communication within our local network.

Initial setup

On your computer, set up TailScale (https://tailscale.com/).

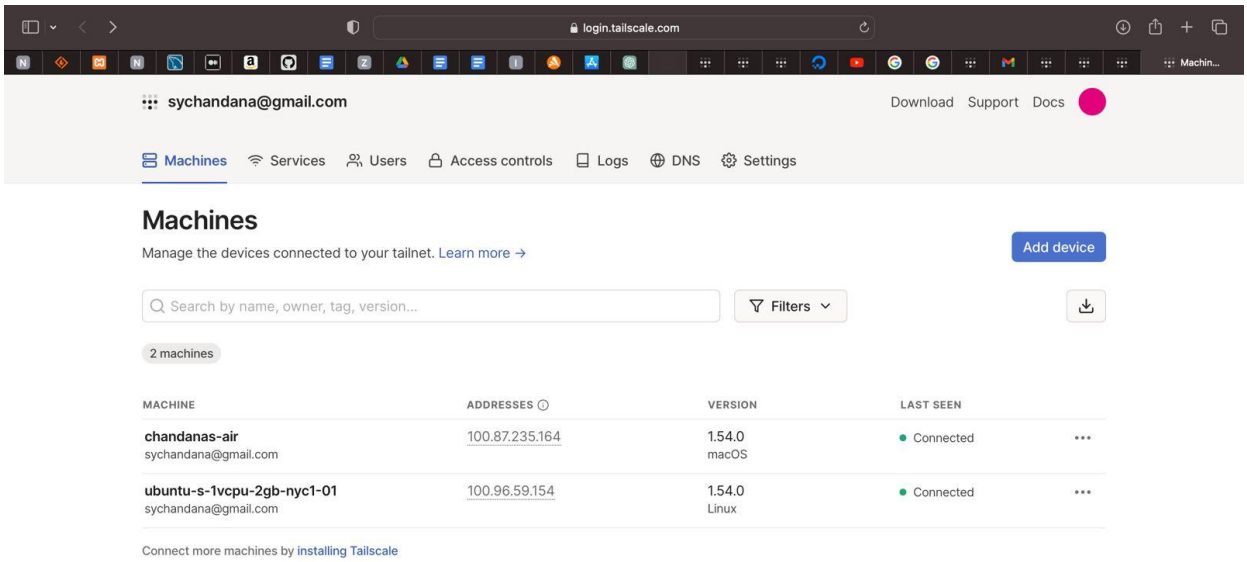
Create a new Linux-based virtual machine (VM) on the cloud; I recommend Digital Ocean. Make sure that the VM has an external-facing IP address.

Set up TailScale on the VM. Make sure that your computer and the VM are on the same TailScale network.

Once you complete the steps above, take a screenshot of your TailScale's Admin Console: https://login.tailscale.com/admin/machines Make sure that it shows at least two devices and their TailScale IP addresses: your computer and the VM.

Answer:
In the below screenshot, we see 2 devices. Machine ‘chandanas-air’ is our computer and Machine ‘ubuntu-s-1vcpu-2gb-nyc-01’ is the Digital Ocean VM.

Screenshot:

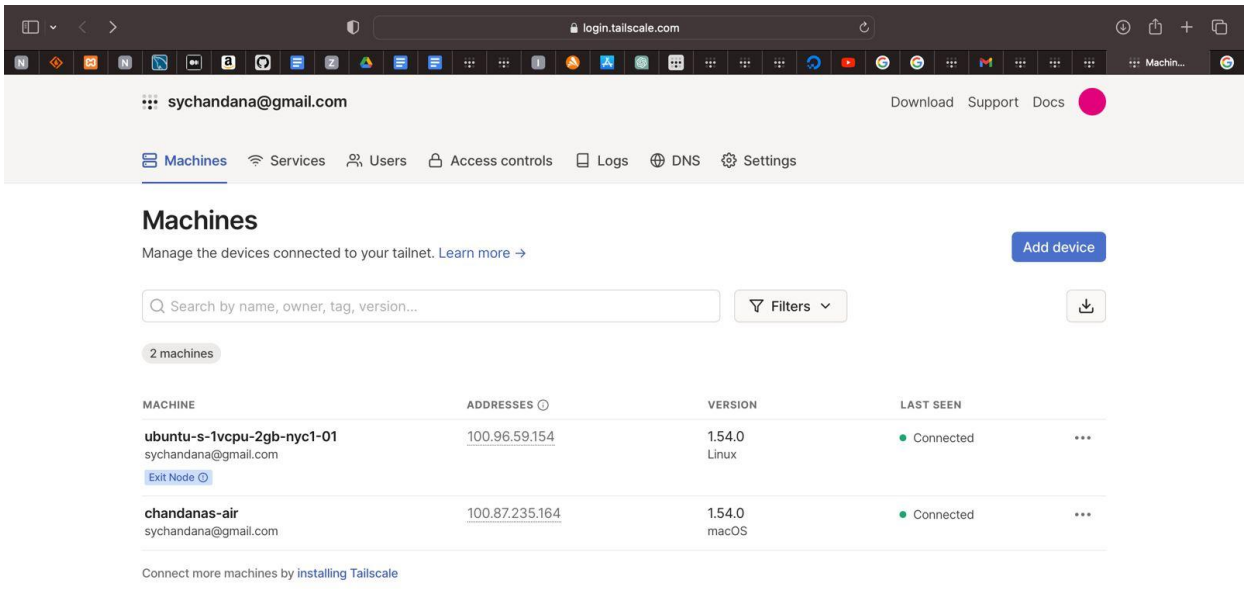


Set up a TailSacle exit node

Turn your VM into a TailScale exit node: https://tailscale.com/kb/1103/exit-nodes/

Once you're done, take a screenshot of your TailScale's Admin Console (https://login.tailscale.com/admin/machines) to show that the VM is indeed an exit node.

Answer:
Screenshot:



What is the external-facing IP address of the VM?

Answer:
143.198.164.203
It's the IP address of Digital Ocean VM (Available on the Digital Ocean dashboard or can be obtained through ‘curl ifconfig.me’ command)

What is the IP address on the TailScale interface?

Answer:
100.96.59.154
It's the IP address of Tailscale (Available on the Tailscale dashboard or can be obtained through ‘tailscale ip’ command)

To support your responses above, take a screenshot of the `ip a` command's output on the VM.

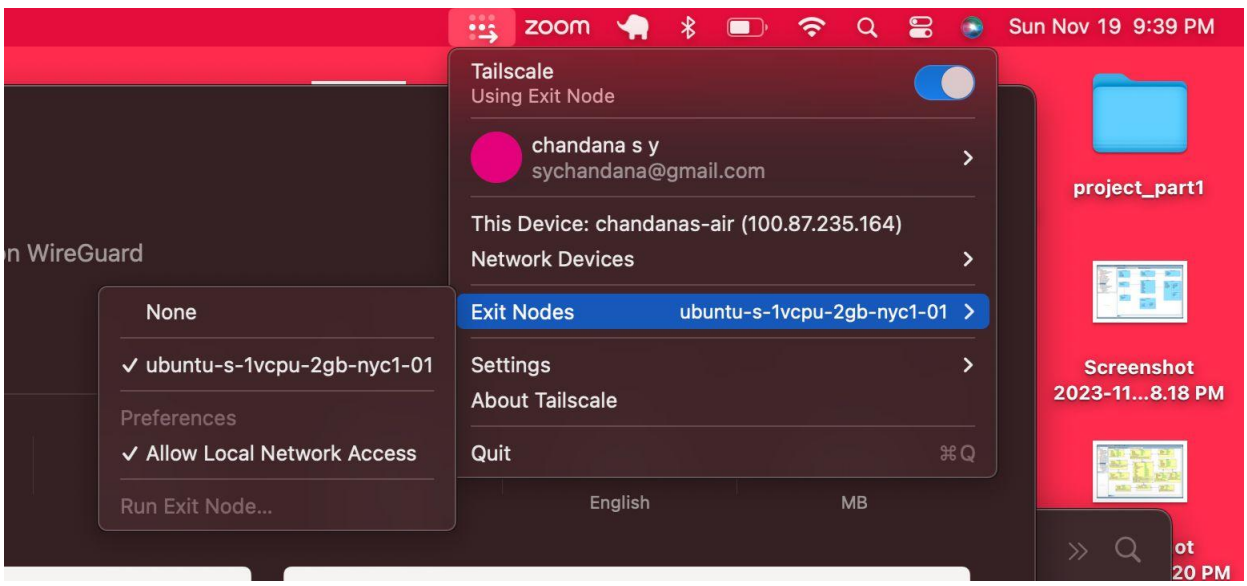
Answer:
Screenshot:

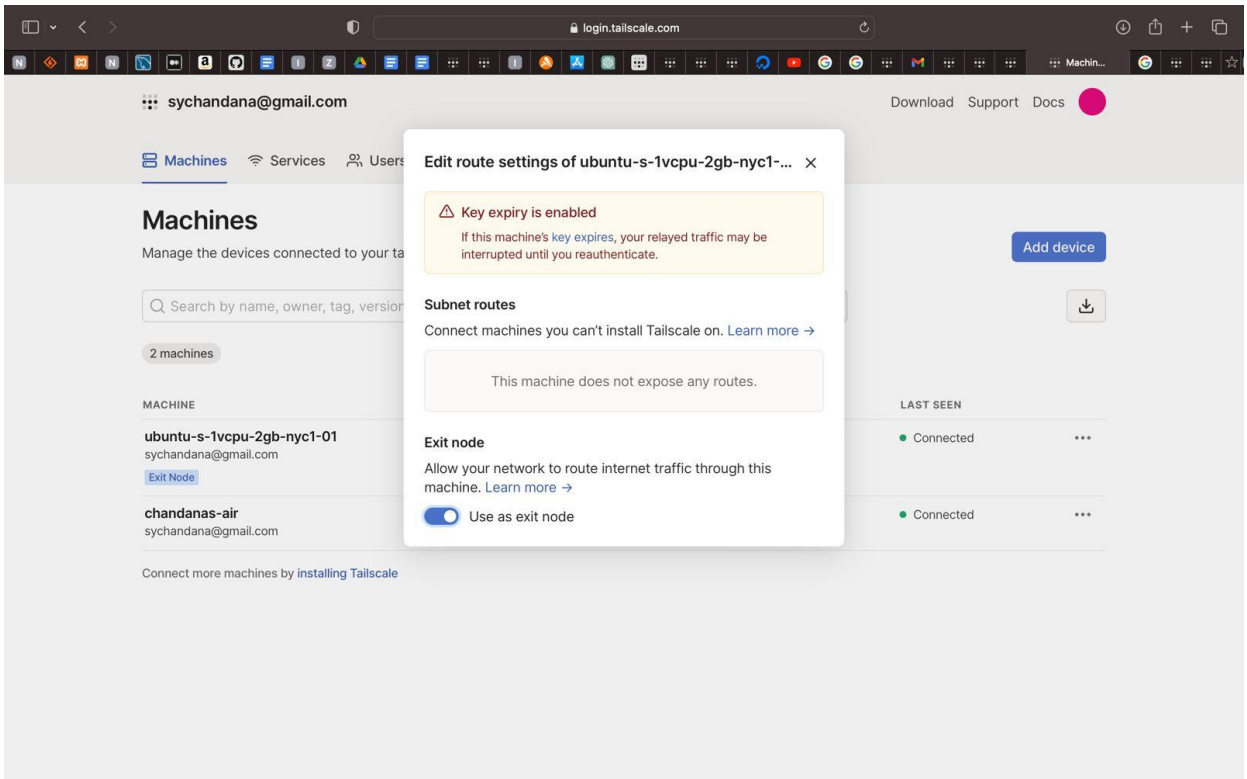
```
root@ubuntu-s-1vcpu-2gb-nyc1-01:~# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 8a:7d:d7:6b:0a:7c brd ff:ff:ff:ff:ff:ff
    altname enp0s3
    altname ens3
    inet 143.198.164.203/20 brd 143.198.175.255 scope global eth0
       valid_lft forever preferred_lft forever
    inet 10.10.0.5/16 brd 10.10.255.255 scope global eth0
       valid_lft forever preferred_lft forever
    inet6 fe80::887d:d7ff:fe6b:a7c/64 scope link
       valid_lft forever preferred_lft forever
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether fa:3b:48:64:bd:4c brd ff:ff:ff:ff:ff:ff
    altname enp0s4
    altname ens4
    inet 10.116.0.2/20 brd 10.116.15.255 scope global eth1
       valid_lft forever preferred_lft forever
    inet6 fe80::f83b:48ff:fe64:bd4c/64 scope link
       valid_lft forever preferred_lft forever
4: tailscale0: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1280 qdisc pfifo_fast state UNKNOWN group default qlen 500
    link/none
    inet 100.96.59.154/32 scope global tailscale0
       valid_lft forever preferred_lft forever
    inet6 fd7a:115c:a1e0:ab12:4843:cd96:6260:3b9a/128 scope global
       valid_lft forever preferred_lft forever
    inet6 fe80::5af1:6caa:917c:4dfb/64 scope link stable-privacy
       valid_lft forever preferred_lft forever
```

Using your computer's TailScale client, connect your computer to the VM's exit node.

Take a screenshot of your computer's TailScale client to show that it is connected to the exit node.

Answer:
Screenshot:

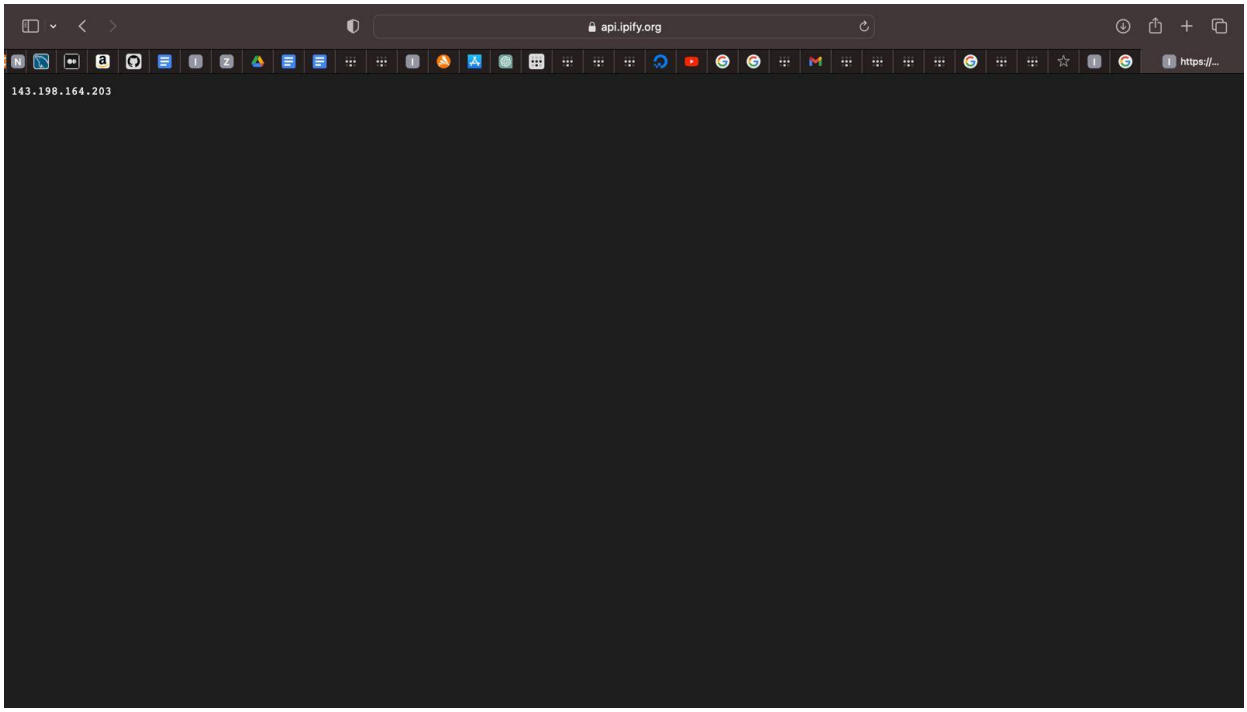




On your computer's browser, go to https://api.ipify.org.

What is the output? Why? Include relevant screenshots if needed.

Answer:
When the computer is connected to a Tailscale exit node configured on a DigitalOcean VM, the IP address obtained from https://api.ipify.org is same as the VM's external-facing IP (143.198.164.203) due to Tailscale's traffic routing. Tailscale channels the computer's internet-bound activities through the designated exit node, and as the VM serves as this exit point, the IP displayed by https://api.ipify.org represents the VM's external-facing address, showcasing it as the origin of your computer's internet traffic while utilizing Tailscale's network for secure and encrypted communication.



Once you're done with the above, disconnect your computer from the VM's exit node.

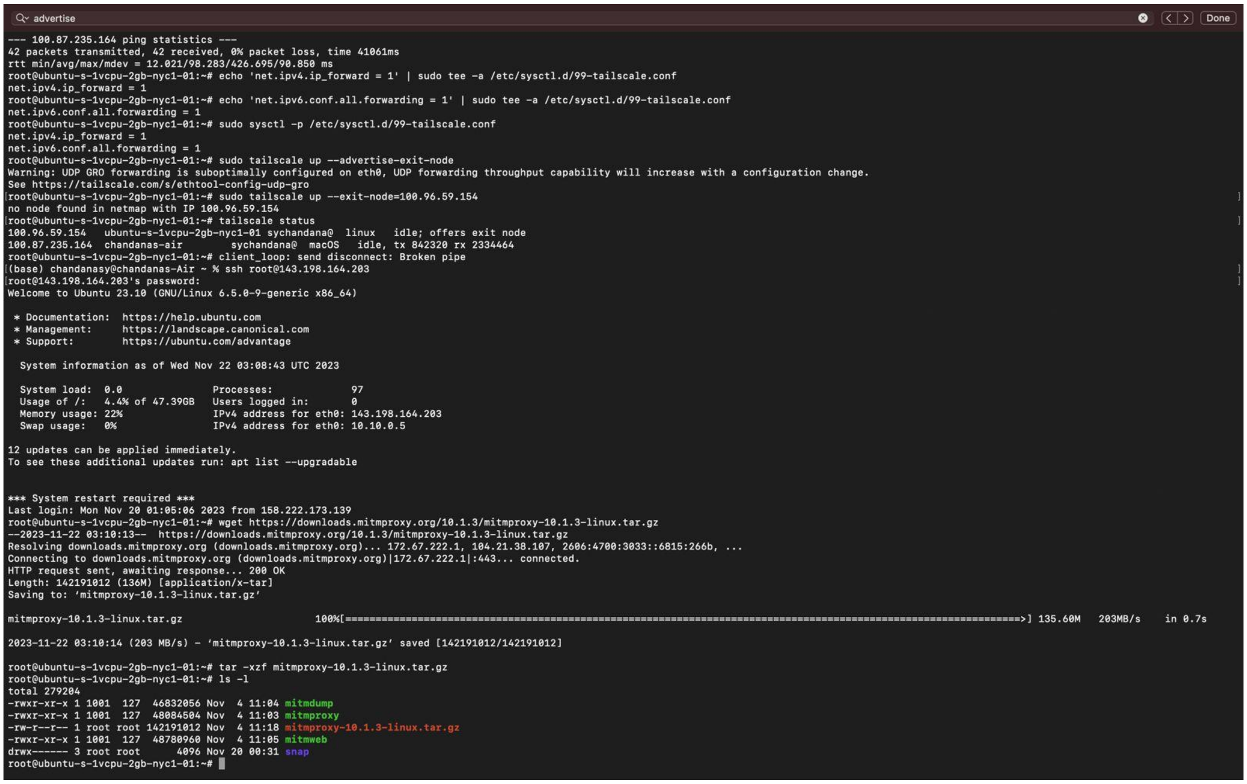
Set up mitmproxy

Initial setup

Download mitmproxy on your VM. I recommend this link:
https://downloads.mitmproxy.org/10.1.3/mitmproxy-10.1.3-linux.tar.gz

De-compress the file. Take a screenshot of the terminal to show that you have properly downloaded all three binaries of mitmproxy.

Answer:
Screenshot:



```
--- 100.87.235.164 ping statistics ---
42 packets transmitted, 42 received, 0% packet loss, time 41061ms
rtt min/avg/max/mdev = 12.021/98.283/426.695/90.850 ms
root@ubuntu-s-1vcpu-2gb-nyc1-01:~# echo 'net.ipv4.ip_forward = 1' | sudo tee -a /etc/sysctl.d/99-tailscale.conf
net.ipv4.ip_forward = 1
root@ubuntu-s-1vcpu-2gb-nyc1-01:~# echo 'net.ipv6.conf.all.forwarding = 1' | sudo tee -a /etc/sysctl.d/99-tailscale.conf
net.ipv6.conf.all.forwarding = 1
root@ubuntu-s-1vcpu-2gb-nyc1-01:~# sudo sysctl -p /etc/sysctl.d/99-tailscale.conf
net.ipv4.ip_forward = 1
net.ipv6.conf.all.forwarding = 1
root@ubuntu-s-1vcpu-2gb-nyc1-01:~# sudo tailscale up --advertise-exit-node
Warning: UDP GRO forwarding is suboptimally configured on eth0, UDP forwarding throughput capability will increase with a configuration change.
See https://tailscale.com/s/ethtool-config-udp-gro
root@ubuntu-s-1vcpu-2gb-nyc1-01:~# sudo tailscale up --exit-node=100.96.59.154
no node found in netmap with IP 100.96.59.154
root@ubuntu-s-1vcpu-2gb-nyc1-01:~# tailscale status
100.96.59.154   ubuntu-s-1vcpu-2gb-nyc1-01 sychandana@  linux   idle; offers exit node
100.87.235.164  chandanas-air        sychandana@  macOS   idle, tx 842320 rx 2334464
root@ubuntu-s-1vcpu-2gb-nyc1-01:~# client_loop: send disconnect: Broken pipe
(base) chandanasy@chandanas-Air ~ % ssh root@143.198.164.203
root@143.198.164.203's password:
Welcome to Ubuntu 23.10 (GNU/Linux 6.5.0-9-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Wed Nov 22 03:08:43 UTC 2023

  System load:  0.0               Processes:             97
  Usage of /:   4.4% of 47.39GB   Users logged in:       0
  Memory usage: 22%               IPv4 address for eth0: 143.198.164.203
  Swap usage:   0%                IPv4 address for eth0: 10.10.0.5

12 updates can be applied immediately.
To see these additional updates run: apt list --upgradable


*** System restart required ***
Last login: Mon Nov 20 01:05:06 2023 from 158.222.173.139
root@ubuntu-s-1vcpu-2gb-nyc1-01:~# wget https://downloads.mitmproxy.org/10.1.3/mitmproxy-10.1.3-linux.tar.gz
--2023-11-22 03:10:13--  https://downloads.mitmproxy.org/10.1.3/mitmproxy-10.1.3-linux.tar.gz
Resolving downloads.mitmproxy.org (downloads.mitmproxy.org)... 172.67.222.1, 104.21.38.107, 2606:4700:3033::6815:266b, ...
Connecting to downloads.mitmproxy.org (downloads.mitmproxy.org)|172.67.222.1|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 142191012 (136M) [application/x-tar]
Saving to: ‘mitmproxy-10.1.3-linux.tar.gz’

mitmproxy-10.1.3-linux.tar.gz                      100%[===============================================================================================================>] 135.60M   203MB/s    in 0.7s

2023-11-22 03:10:14 (203 MB/s) - ‘mitmproxy-10.1.3-linux.tar.gz’ saved [142191012/142191012]

root@ubuntu-s-1vcpu-2gb-nyc1-01:~# tar -xzf mitmproxy-10.1.3-linux.tar.gz
root@ubuntu-s-1vcpu-2gb-nyc1-01:~# ls -l
total 279204
-rwxr-xr-x 1 1001  127  46832056 Nov  4 11:04 mitmdump
-rwxr-xr-x 1 1001  127  48084504 Nov  4 11:03 mitmproxy
-rw-r--r-- 1 root root 142191012 Nov  4 11:18 mitmproxy-10.1.3-linux.tar.gz
-rwxr-xr-x 1 1001  127  48780960 Nov  4 11:05 mitmweb
drwx------ 3 root root      4096 Nov 20 00:31 snap
root@ubuntu-s-1vcpu-2gb-nyc1-01:~# 
```

Transparent proxying

On the VM, set up mitmproxy's transparent proxying by following Steps 1 through 5: https://docs.mitmproxy.org/stable/howto-transparent/ (For Step 3 make sure to replace `eth0` with `tailscale0`.)

On your computer, do not connect to the VM's exit node yet. Download and run Firefox. Open Firefox and visit http://mitm.it/. What do you see? Take a screenshot:

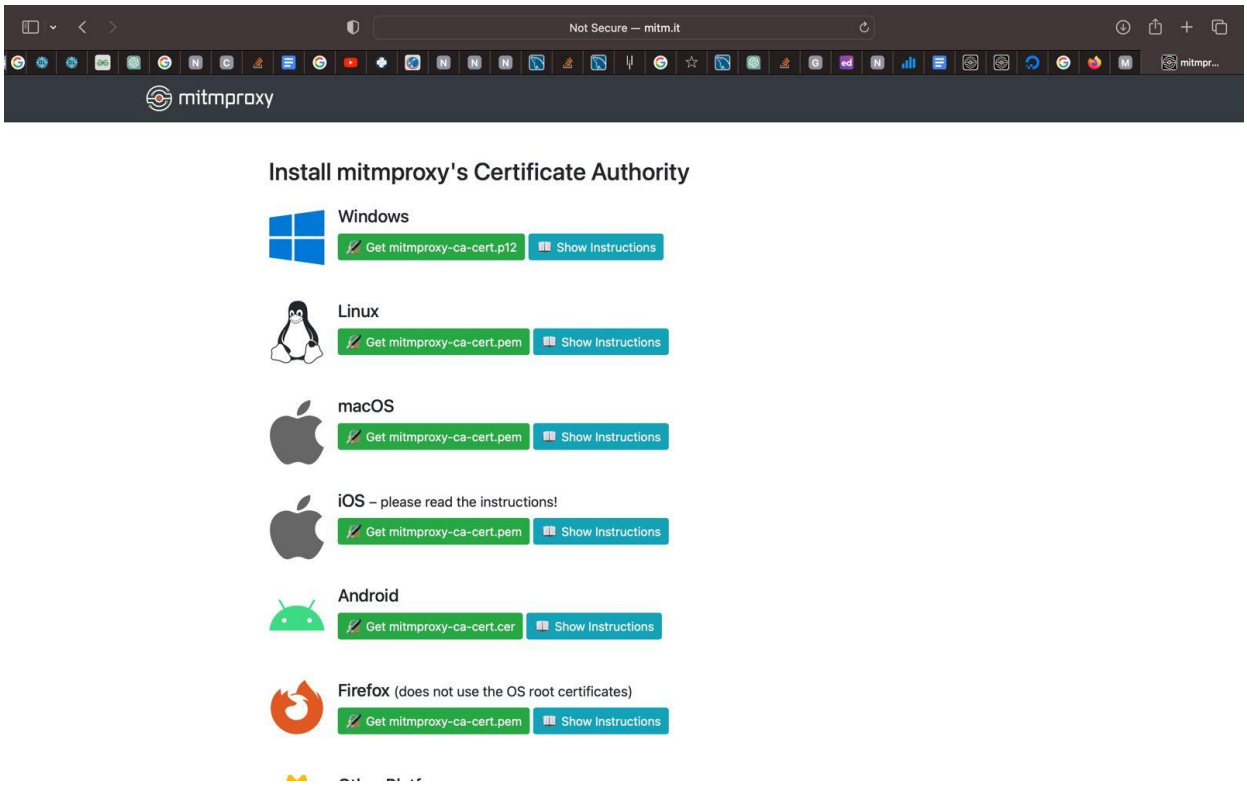
Answer:
We see the below message (screenshot) that suggests that the traffic is not currently passing through MITMproxy, indicating that the proxy interception is not active at the moment.

Screenshot:

Now, connect to the VM's exit node. Visit http://mitm.it/ again on Firefox. What do you see? Take a screenshot:

Answer:
We see the below (screenshot) which displays content served by MITMproxy, providing information related to Mitmproxy's certificate authority installation. This is because the traffic is now passing through the MITMproxy due to the connection to the VM's exit node.

Screenshot:



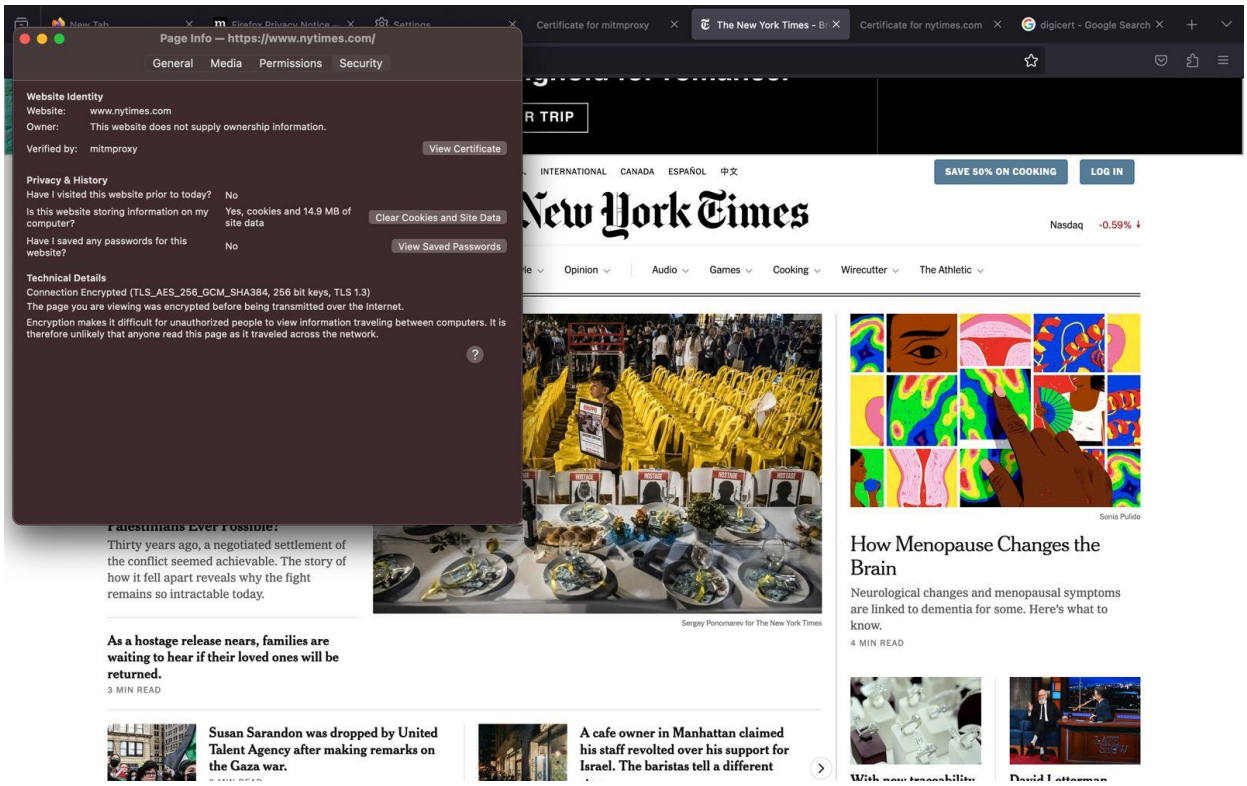
Explain how and why these two screenshots are the same/different.

Answer:
The two screenshots showcase differences based on the presence or absence of MITMproxy interception. The first screenshot, without the VM's exit node connection or MITMproxy interception, displays a standard browser message indicating that traffic isn't passing through MITMproxy. Conversely, the second screenshot, after connecting to the VM's exit node with MITMproxy interception properly set up, shows content served by MITMproxy, displaying instructions on Mitmproxy's certificate authority installation. These differences highlight the impact of MITMproxy's interception on network traffic, with one screen indicating the absence of interception and the other depicting intercepted traffic routed through MITMproxy.

On Firefox, follow the instructions of http://mitm.it/ and set up mitmproxy's CA in Firefox's trust store.

On Firefox, visit https://www.nytimes.com/. Take a screenshot of the certificate presented by NY Times to your Firefox. [1]

Answer:
Screenshot:

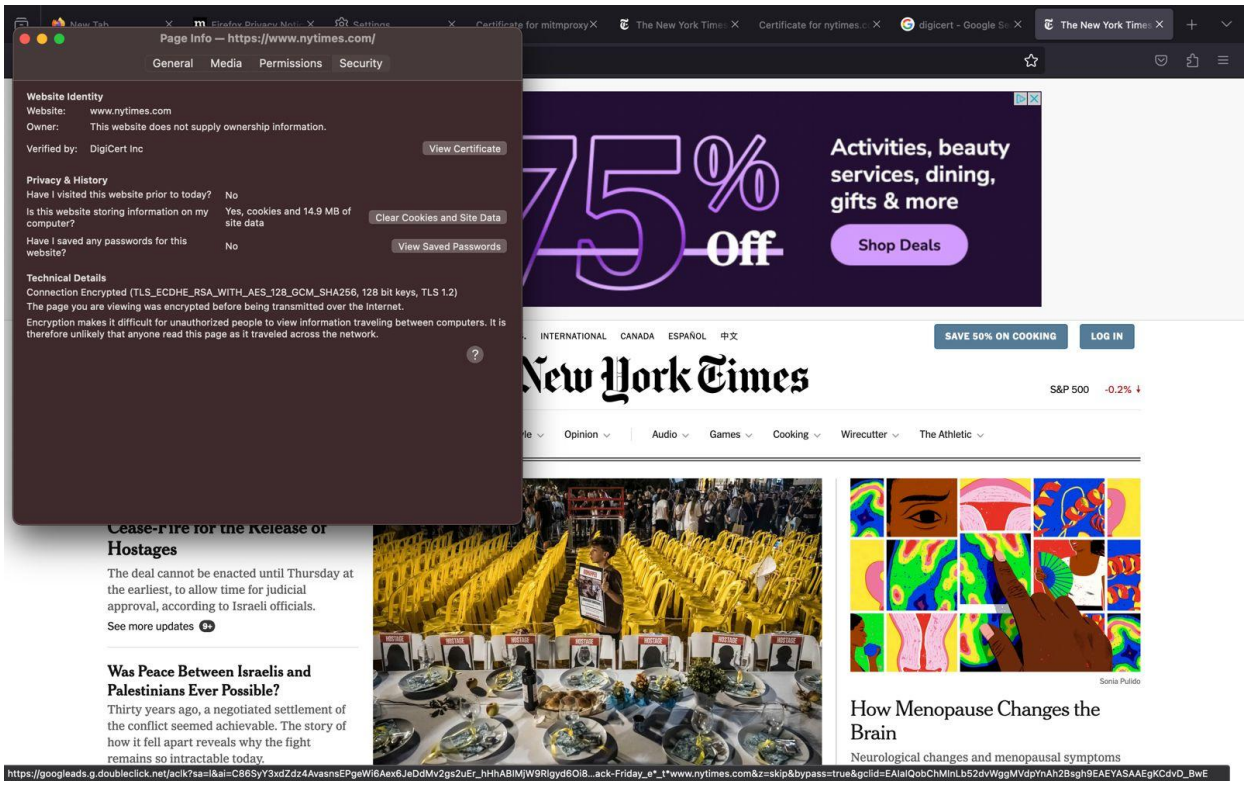


Why do you see the certificate above?

Answer:
Since we are connected to the VM's exit node, the mitmproxy is active and therefore we see that the New York Times certificate is verified by mitmproxy and not the original certificate. The appearance of the mitmproxy certificate while accessing the New York Times indicates that mitmproxy intercepted the HTTPS traffic, replaced the original SSL/TLS certificate with its substitute, and the browser trusts this substitute certificate due to the prior installation of mitmproxy's CA certificate.

Disconnect from the VM's exit node. On Firefox, visit https://www.nytimes.com/. Take a screenshot of the certificate presented by NY Times to your Firefox.

Answer:
Screenshot:



Is this certificate the same or different compared with the certificate in [1]? How? Why?

Answer:
Once we are disconnect from the VM's exit node, we see that the New York Times certificate is now verified by DigiCert Inc (the original SSL/TLS certificate of New York Times website). This change in certificate presentation indicates that the browser is no longer routed through the mitmproxy setup, allowing it to establish direct connections to the New York Times website and receive the original SSL/TLS certificates issued by trusted certificate authorities.

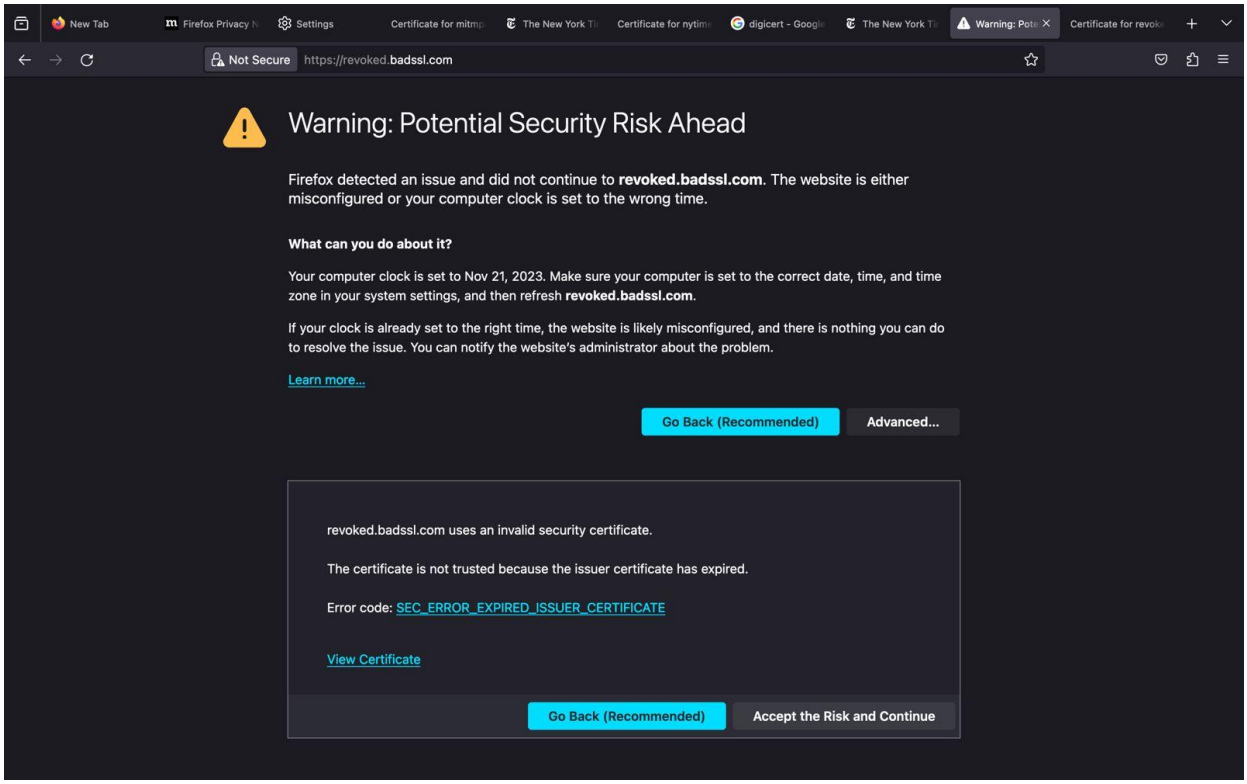
Keep your computer disconnected from the VM's exit node.

On Firefox, visit https://revoked.badssl.com

What do you see? Show your screenshot.

Answer:
While the computer is disconnected from the VM's exit node, we see that below (screenshot) message. This website intentionally presents revoked or invalid certificates to assess how browsers handle such scenarios. Therefore, while disconnected from the VM's exit node, visiting https://revoked.badssl.com triggers a security warning, indicating that the certificate cannot be trusted due to its invalid status.

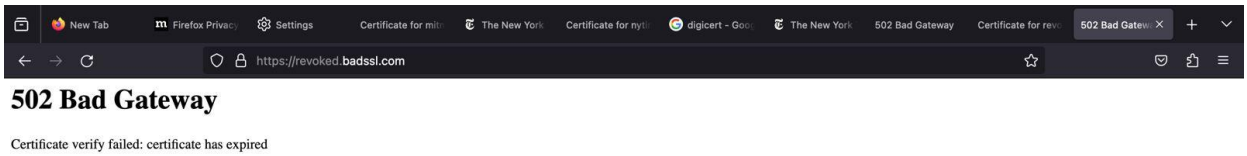
Screenshot:



Now, connect to the VM's exit node. Again, visit https://revoked.badssl.com/ on Firefox. What do you see? Show your screenshot.

Answer:
When we reconnect to the VM's exit node and then visit https://revoked.badssl.com/ in Firefox, we see the below (screenshot) message. The traffic will be routed through the VM's exit node and it will be intercepted by mitmproxy. The outcome depends on how mitmproxy handles intercepted traffic and therefore we see a substitute certificate error related to the revoked certificate.

Screenshot:



502 Bad Gateway

Certificate verify failed: certificate has expired

Explain the similarities or differences between these two screenshots.

Answer:
While both screenshots show scenarios involve SSL/TLS warnings related to the revoked certificate, the difference lies in the response generated by the interception tool (mitmproxy). When traffic passes through the VM's exit node, it results in a different error message (502 Bad Gateway) due to the interception and manipulation of SSL/TLS certificates by mitmproxy.

Similarities:
The similarities between the two screenshots involve the presence of SSL/TLS-related warnings or errors due to the website's intentionally revoked or invalid certificates.

Differences:
The screenshot while disconnected from the VM's exit node displays a standard browser warning related to an invalid certificate, indicating that the site's certificate cannot be trusted due to its revoked status. Whereas, the screenshot while connected to the exit node displays a 502 Bad Gateway error, which occurs due to mitmproxy's handling of the intercepted traffic, presenting a substitute certificate related to the revoked one, triggering the error message.

Blocking websites

You can actually block certain websites for computers connected to the VM's exit node. This is how censorship roughly works!

On the VM, run the following command to turn on a special firewall [2]:

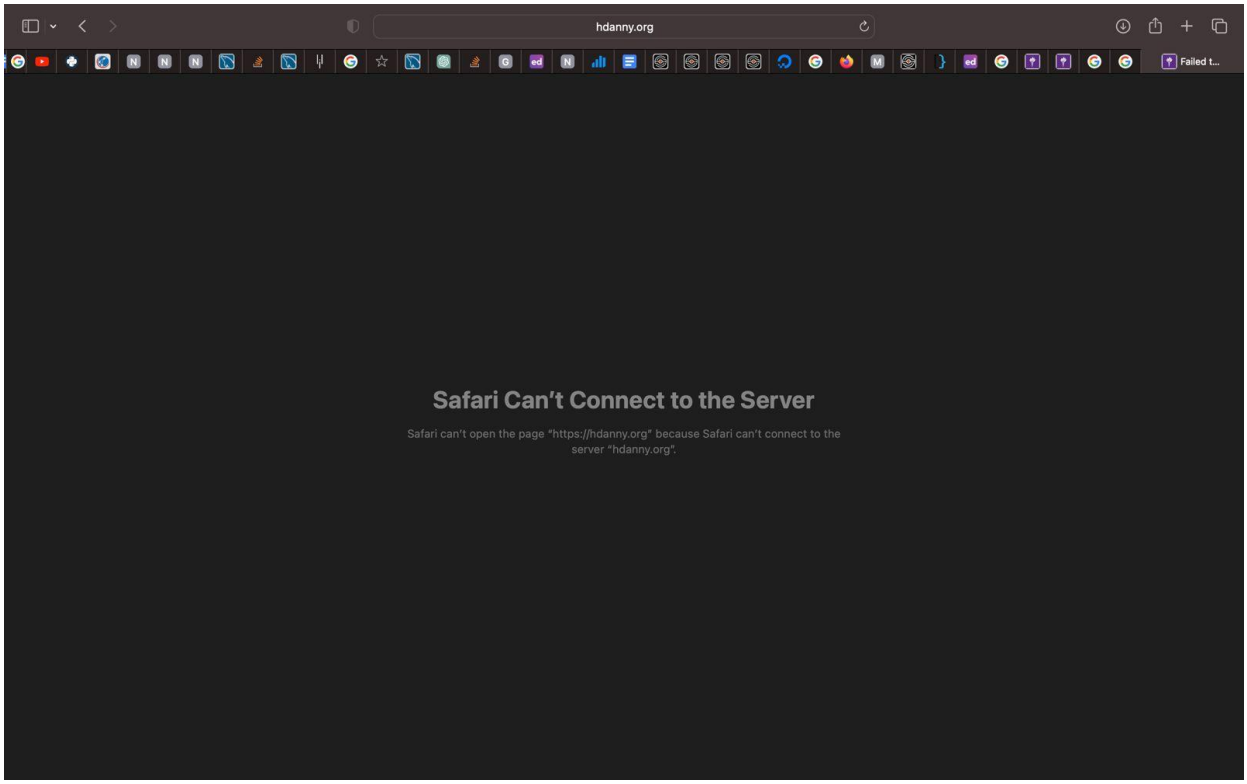
```
iptables -A INPUT -i eth0 -s 128.238.64.23 -j DROP
```

On your computer, keep connected to the VM's exit node. Visit https://hdanny.org on Firefox. What do you see? Show your screenshot.

Answer:
Note: Please note that this task was carried out using the Safari browser due to Firefox not responding as expected (We encountered difficulties loading https://hdanny.org both before and after attempting to block the website).

The above command used on the VM adds a rule to the firewall that drops incoming traffic from the IP address (128.238.64.23) on the network interface eth0. This rule effectively blocks incoming connections from https://hdanny.org.

Screenshot:



Why do you see the above? Explain by referencing the command above.

Answer:

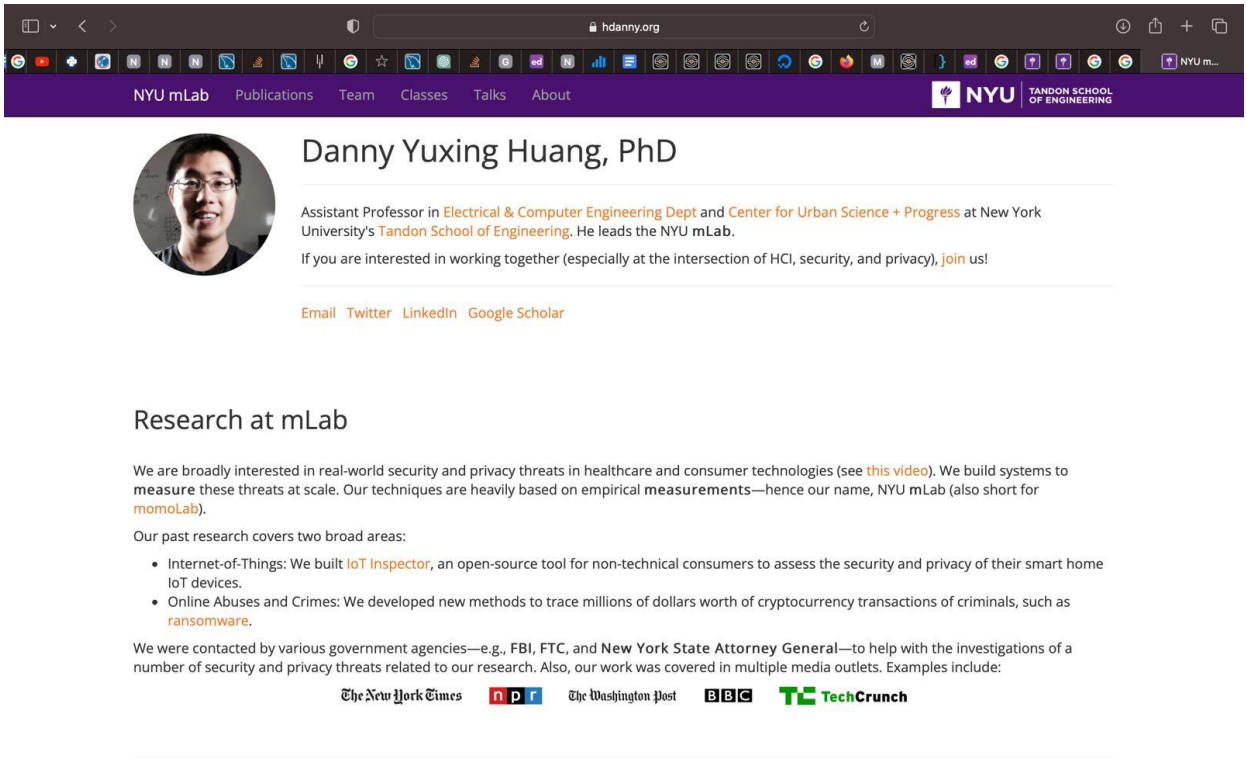
The command ‘iptables -A INPUT -i eth0 -s 128.238.64.23 -j DROP’ used on the VM adds a rule to the firewall that drops incoming traffic (-j DROP) from the specified source IP address (-s 128.238.64.23) on the network interface eth0. This rule effectively blocks incoming connections from the specified IP address that is, https://hdanny.org.
When visiting https://hdanny.org on the computer while connected to the VM's exit node, and with the specified rule blocking traffic from the source IP, we encounter connection issues while attempting to access the website. The browser displays a ‘can’t connect to the server’ error indicating that the site cannot be reached.

Suppose you'd like to restore access to https://hdanny.org.

What command on the VM would you type to allow access to hdanny.org again? (Hint: use `iptables` and delete the rule you just added in [2])

Answer:
The command to delete the rule in iptables and allow access to hdanny.org is:
iptables -D INPUT -i eth0 -s 128.238.64.23 -j DROP

This command deletes (-D) the rule from the INPUT chain so now the traffic from the IP address (128.238.64.23) will no longer be blocked, allowing access to https://hdanny.org again for computers connected through the VM's exit node.



App analysis

Initial setup on the phone

Set up TailScale on your phone. Connect to the VM's exit node.

Make sure to add mitmproxy's CA to your phone's trust store, following the instructions on http://mitm.it/

Verify that your phone's traffic is successfully intercepted by mitmproxy by visiting https://www.nytimes.com/ on your phone and showing the certificate your phone browser sees:

Include screenshots of the above.

Note: On some iOS versions neither Safari nor Chrome lets you view the certificates. If that's the case, please just upload screenshots of the mitmproxy certificates in your system settings.

Answer:

The below screenshot displays the mitmproxy's CA with a warning saying ' Not Trusted' instead of the original certificate from the website (https://www.nytimes.com/), confirming successful interception of phone's traffic through mitmproxy.

Screenshot:

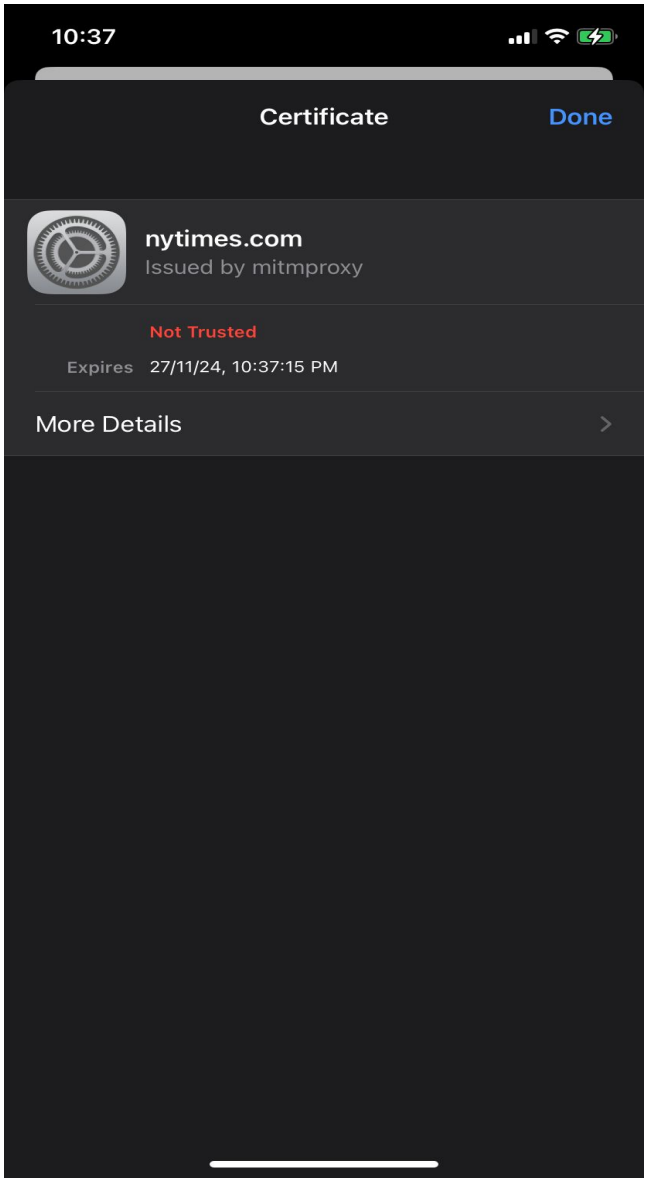
10:37

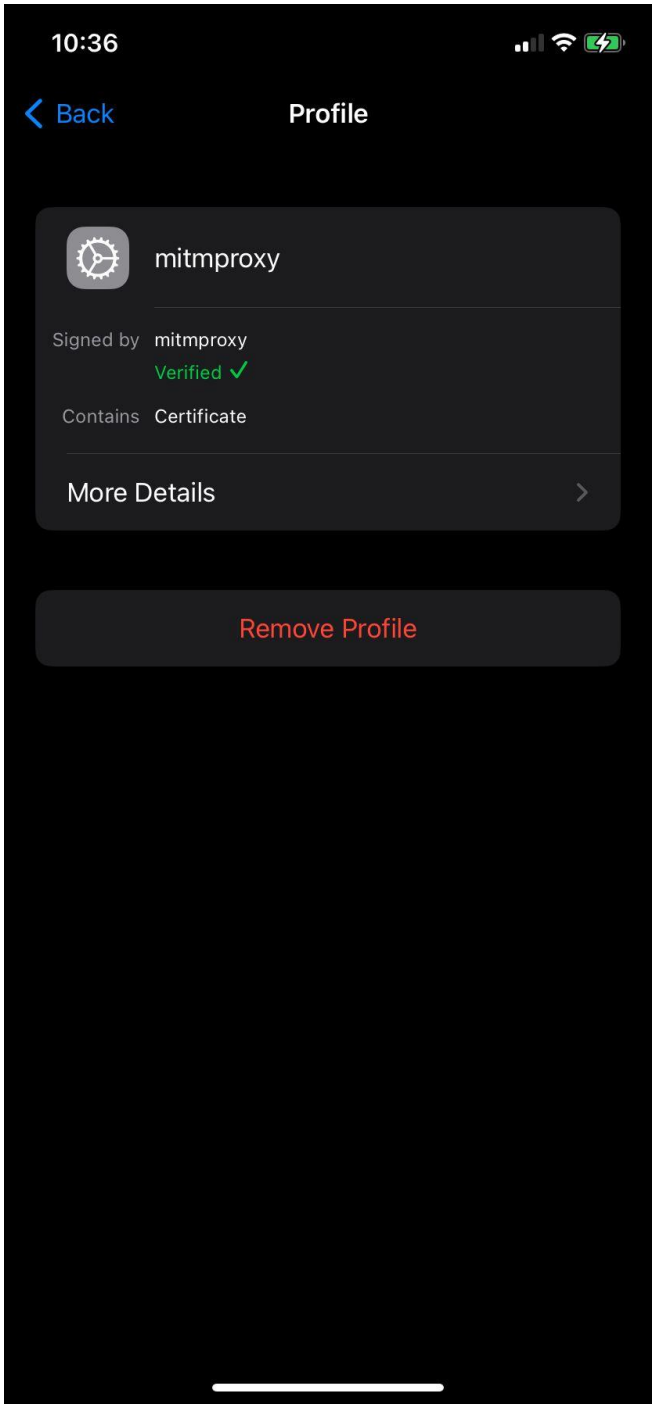
This Connection Is Not Private

This website may be impersonating "www.nytimes.com" to steal your personal or financial information. You should close this page.

Show Details Close Page

AA nytimes.com


10:37
Certificate
Done
nytimes.com
Issued by mitmproxy
Not Trusted
Expires 27/11/24, 10:37:15 PM
More Details



Analyzing Amtrak

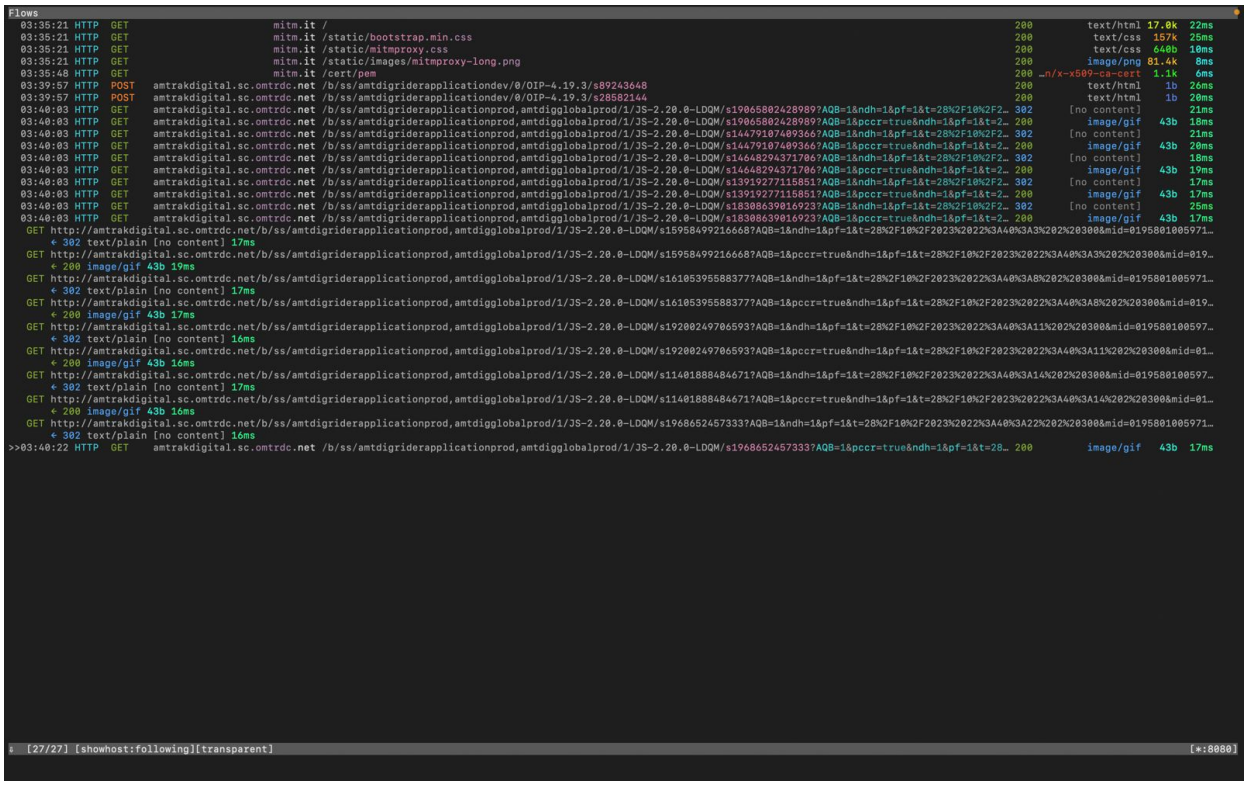
On your phone, download the Amtrak app (either Android or iOS).

On your VM, make sure mitmproxy is running (i.e., `mitmproxy --mode transparent --showhost` or use `mitmweb`) and the Follow mode is on (i.e., by pressing the `F` key while on mitmproxy's user interface in `mitmproxy`). In this way, the screen automatically scrolls to the latest intercepted HTTP traffic.

On your phone, open the Amtrak app and randomly interact with it.

Take a screenshot of mitmproxy's user interface. What do you see?

Screenshot:



Why do you see this?

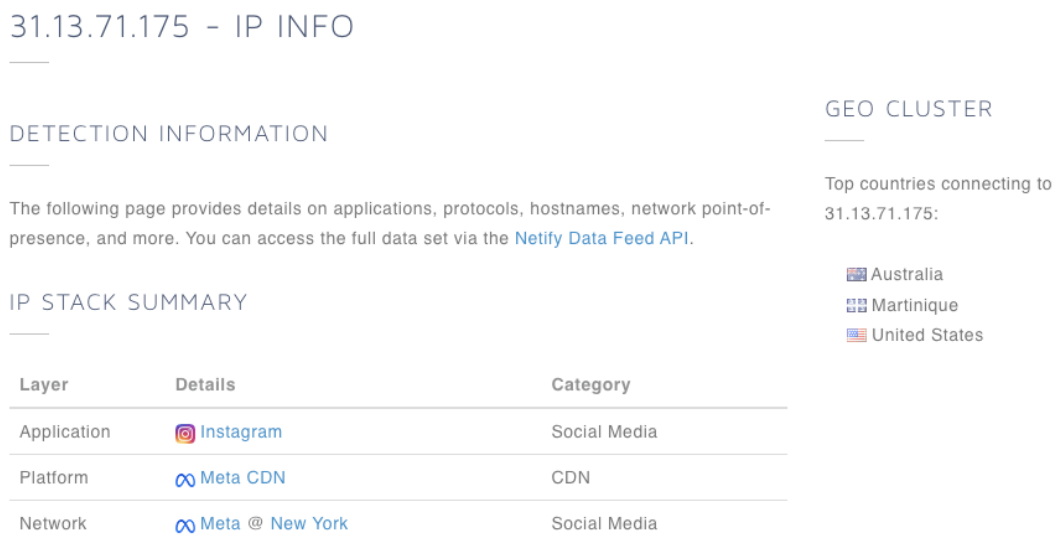
We can see this traffic on mitmproxy's user interface as it intercepts, inspects, and relays the requests and responses between the client (phone running the Amtrak app) and the server (Amtrak's web services). This suggests that the app likely employs less robust encryption and security measures.

Analyzing X (or Twitter or Instagram or your favorite app) [2 bonus points]

Repeat the Amtrak analysis but, instead of Amtrak, try to intercept the traffic an app of your choice. I'd suggest X, Instagram, or WhatsApp.

Take a screenshot of mitmproxy's user interface. What do you see?

Screenshot:



Why do you see this? How is your observation similar or different compared with the Amtrak analysis?

The captured mitmproxy screenshot for Instagram displays primarily TCP connections and IP addresses, indicating potential encryption or secure protocols, restricting detailed visibility into HTTP content. This suggests Instagram likely employs robust security measures, limiting plaintext data visibility through the proxy and emphasizing encryption for its network communications.

Similarities/Differences:
The analysis of both Amtrak and Instagram through mitmproxy reveals basic network connections. However, while Amtrak's analysis showed detailed HTTP content, such as URLs and requests, suggesting limited encryption, the Instagram analysis primarily displays encrypted or obscured data, indicating robust security measures restricting plaintext visibility within the proxy.