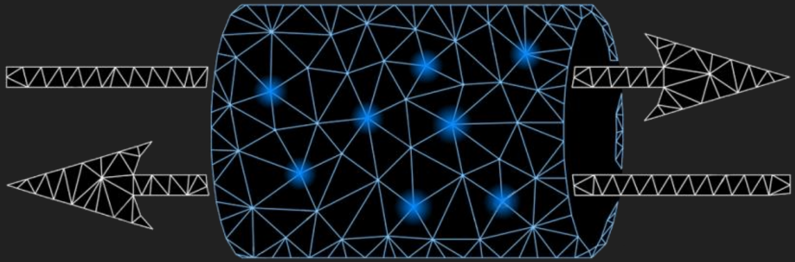
Ali Boraqi
Managing Enterprise Networks
San Francisco State University

What is a VPN?



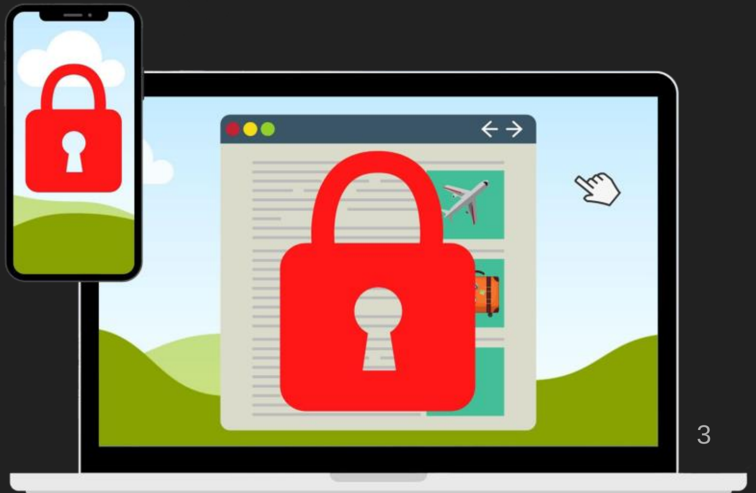
- A Virtual Private Network (VPN) is a private network that runs over the internet using an encrypted connection.

- Think of it as a personal data tunnel that extends between your local network to the exit node located on the server.

- Allows you to appear as if you’re in another location, allowing online freedom and privacy.

Purpose

- While traveling for business, it’s important to have access to work servers and private networks, for professional and personal use.
- The need for a secure and reliable method to access personal file, as security and privacy has become a major concern.
- A VPN would allow users to bypass usage limitations and grant access to the information and services available in their home country.



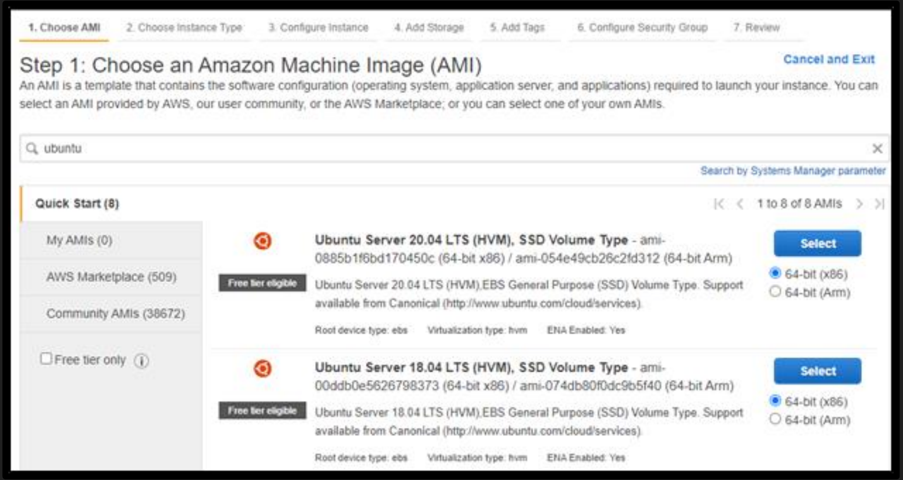
Method

- Utilize Amazon Web Services to create two EC2 instances to run each VPN protocol.
    - One using WireGuard
    - The other using OpenVPN

- Use Wireshark to capture the VPNs during operation to analyze the protocols used by each standard.

WireGuard Installation

- Create an EC2 Instance



Step 1: Choose an Amazon Machine Image (AMI)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. You can select an AMI provided by AWS, our user community, or the AWS Marketplace; or you can select one of your own AMIs.

ubuntu

Quick Start (8)

1 to 8 of 8 AMIs

- My AMIs (0)
- AWS Marketplace (509)
- Community AMIs (38672)

□ Free tier only

Ubuntu Server 20.04 LTS (HVM), SSD Volume Type - ami-0885b1f6bd170450c (64-bit x86) / ami-054e49cb26c2fd312 (64-bit Arm)

Free tier eligible

Ubuntu Server 20.04 LTS (HVM),EBS General Purpose (SSD) Volume Type. Support available from Canonical (http://www.ubuntu.com/cloud/services).

Root device type: ebs Virtualization type: hvm ENA Enabled: Yes

● 64-bit (x86)
○ 64-bit (Arm)

Ubuntu Server 18.04 LTS (HVM), SSD Volume Type - ami-00ddb0e5626798373 (64-bit x86) / ami-074db80f0dc9b5f40 (64-bit Arm)

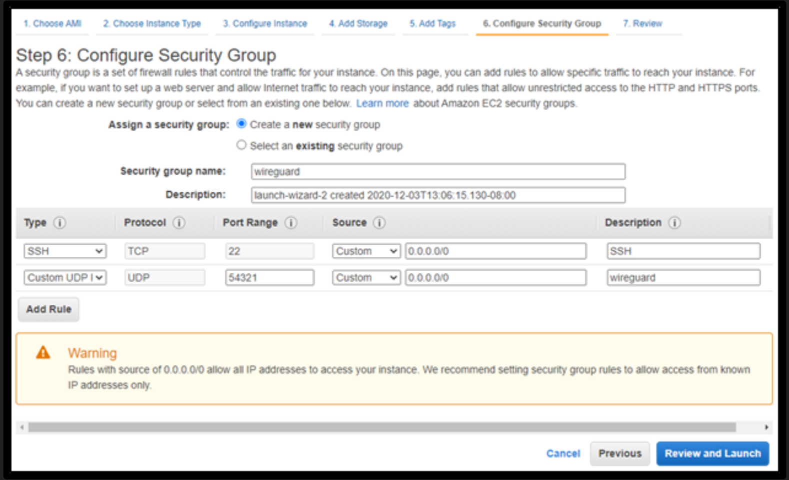
Free tier eligible

Ubuntu Server 18.04 LTS (HVM),EBS General Purpose (SSD) Volume Type. Support available from Canonical (http://www.ubuntu.com/cloud/services).

Root device type: ebs Virtualization type: hvm ENA Enabled: Yes

● 64-bit (x86)
○ 64-bit (Arm)

- Configure Security Group



Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. Learn more about Amazon EC2 security groups.

Assign a security group: ● Create a **new** security group
○ Select an **existing** security group

Security group name: wireguard

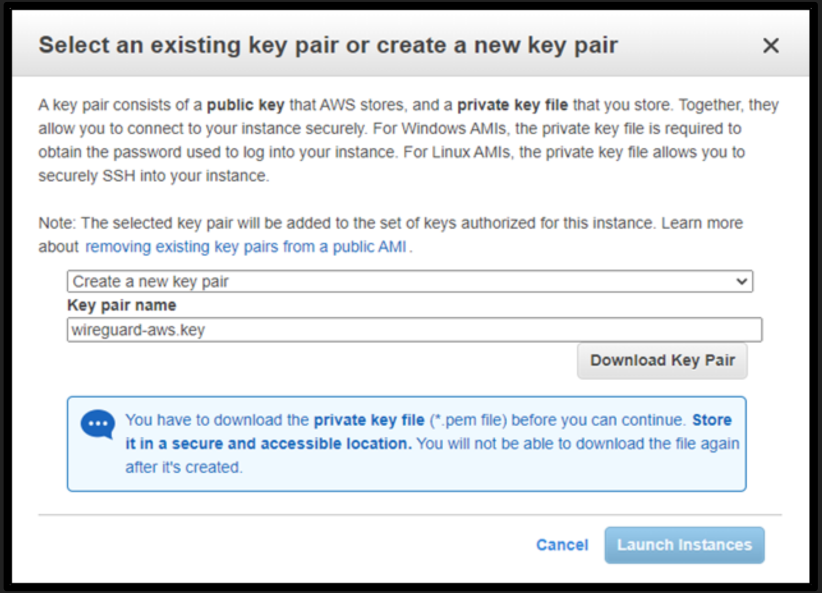
Description: launch-wizard-2 created 2020-12-03T13:06:15.130-08:00

Type	Protocol	Port Range	Source		Description
SSH	TCP	22	Custom	0.0.0.0/0	SSH
Custom UDP I	UDP	54321	Custom	0.0.0.0/0	wireguard

Warning
Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

WireGuard Installation

- Download key pair



Select an existing key pair or create a new key pair ✕

A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.

Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about removing existing key pairs from a public AMI .

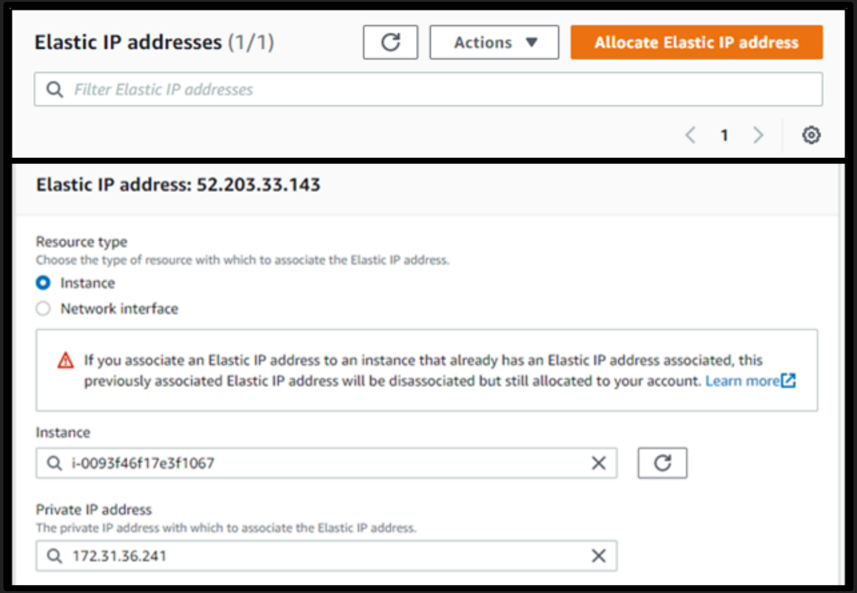
Create a new key pair

Key pair name

wireguard-aws.key

You have to download the **private key file** (*.pem file) before you can continue. **Store it in a secure and accessible location.** You will not be able to download the file again after it's created.

- Assign an external IP address



Elastic IP addresses (1/1)

Elastic IP address: 52.203.33.143

Resource type

Choose the type of resource with which to associate the Elastic IP address.

● Instance

○ Network interface

If you associate an Elastic IP address to an instance that already has an Elastic IP address associated, this previously associated Elastic IP address will be disassociated but still allocated to your account. Learn more

Instance

i-0093f46f17e3f1067

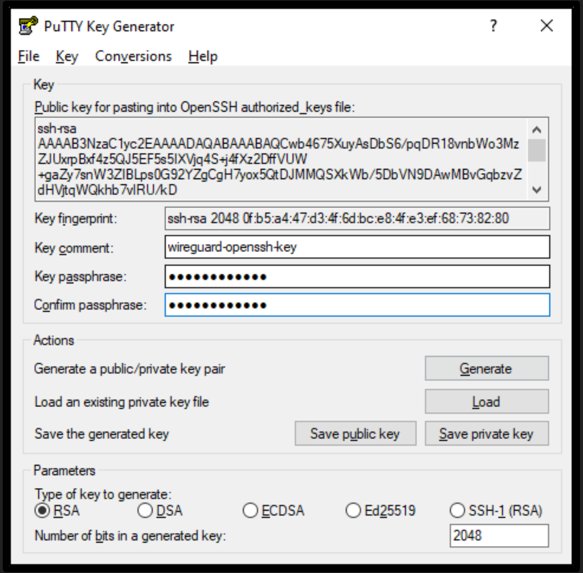
Private IP address

The private IP address with which to associate the Elastic IP address.

172.31.36.241

WireGuard Installation

- Convert key pair



Key

Public key for pasting into OpenSSH authorized_keys file:

```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQCwb4675XuyAsDbS6/pqDR18vnbWo3Mz
ZJUxrpBxf4z5QJ5EF5s5IXVjq4S+j4fXz2DffVUW
+gaZy7snW3ZIBLps0G92YZgCgH7yox5QtDJMMQSXkWb/5DbVN9DAwMBvGqbzvZ
dHVjtqWQkhb7vlRU/kD
```

Key fingerprint: ssh-rsa 2048 0f:b5:a4:47:d3:4f:6d:bc:e8:4f:e3:ef:68:73:82:80

Key comment: wireguard-openssh-key

Key passphrase: •••••••••••••

Confirm passphrase: •••••••••••••

Actions

Generate a public/private key pair

Load an existing private key file

Save the generated key

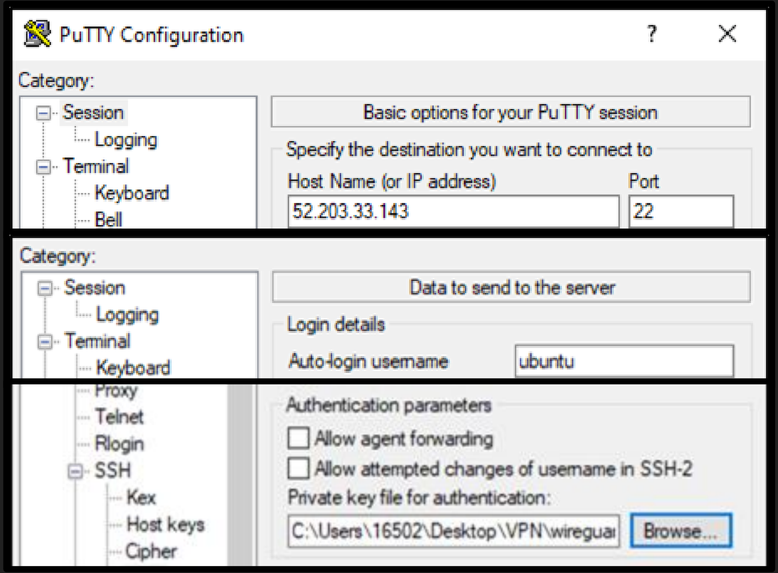
Parameters

Type of key to generate:

● RSA ○ DSA ○ ECDSA ○ Ed25519 ○ SSH-1 (RSA)

Number of bits in a generated key: 2048

- Login to the server



Basic options for your PuTTY session

Specify the destination you want to connect to

Host Name (or IP address): 52.203.33.143

Port: 22

Data to send to the server

Login details

Auto-login username: ubuntu

Authentication parameters

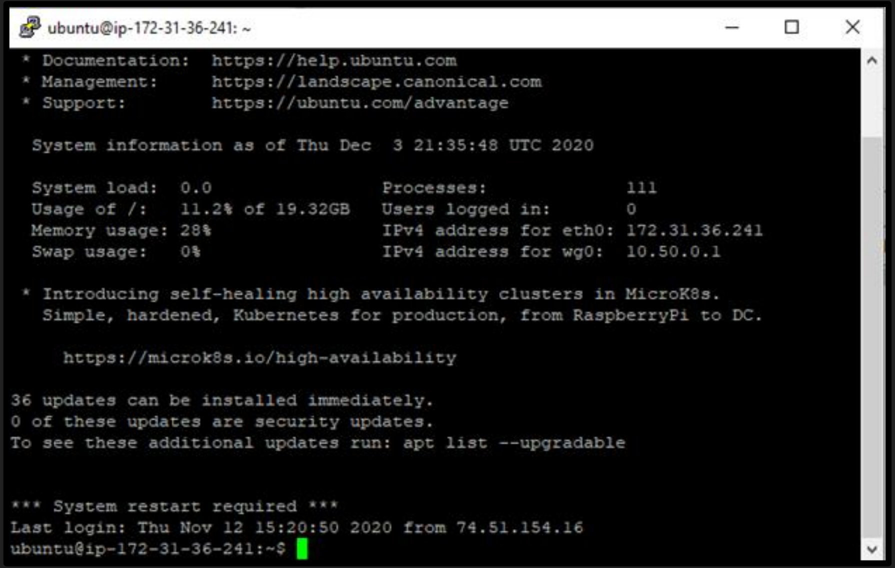
□ Allow agent forwarding

□ Allow attempted changes of username in SSH-2

Private key file for authentication: C:\Users\16502\Desktop\VPN\wireguai

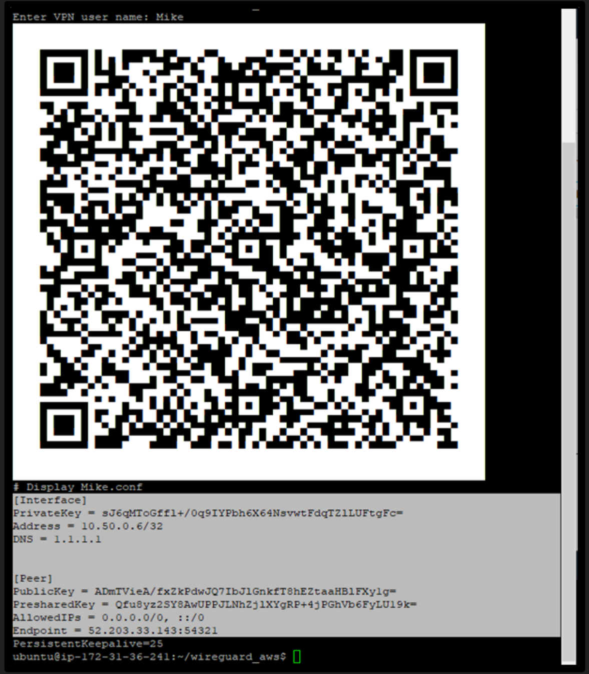
WireGuard Installation

- Configure server



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

  System information as of Thu Dec  3 21:35:48 UTC 2020

  System load:  0.0               Processes:             111
  Usage of /:   11.2% of 19.32GB  Users logged in:       0
  Memory usage: 28%               IPv4 address for eth0: 172.31.36.241
  Swap usage:   0%                IPv4 address for wg0:  10.50.0.1

 * Introducing self-healing high availability clusters in MicroK8s.
   Simple, hardened, Kubernetes for production, from RaspberryPi to DC.

     https://microk8s.io/high-availability

36 updates can be installed immediately.
0 of these updates are security updates.
To see these additional updates run: apt list --upgradable


*** System restart required ***
Last login: Thu Nov 12 15:20:50 2020 from 74.51.154.16
ubuntu@ip-172-31-36-241:~$
```

- Save info as .conf file

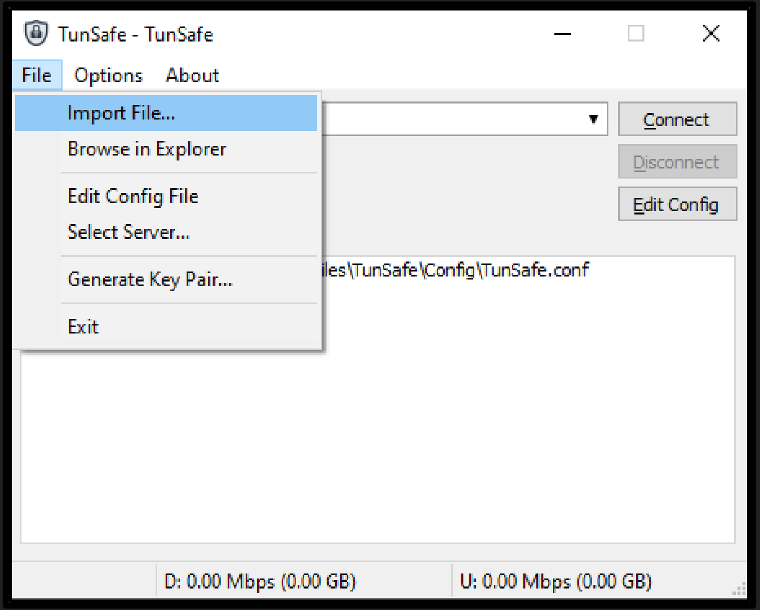


```
Enter VPN user name: Mike
```

```
# Display Mike.conf
[Interface]
PrivateKey = sJ6qMToGffl+/0q9IYPbh6X64NsvwtFdqTZlLUFtgFc=
Address = 10.50.0.6/32
DNS = 1.1.1.1


[Peer]
PublicKey = ADmTVieA/fxZkPdwJQ7IbJlGnkfT8hEZtaaHBlFXylg=
PresharedKey = Qfu8yz2SY8AwUPPJLNhZjlXYgRP+4jPGhVb6FyLUl9k=
AllowedIPs = 0.0.0.0/0, ::/0
Endpoint = 52.203.33.143:54321
PersistentKeepalive=25
ubuntu@ip-172-31-36-241:~/wireguard_aws$
```

WireGuard Installation

- Import .conf file and connect

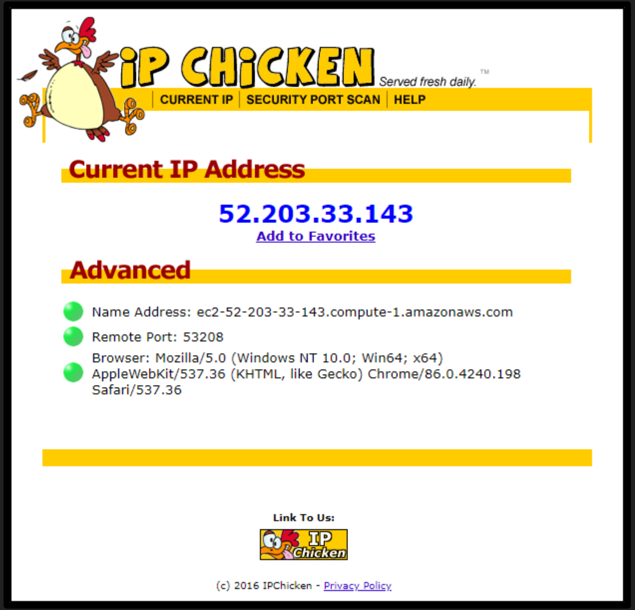


```
iles\TunSafe\Config\TunSafe.conf
```

D: 0.00 Mbps (0.00 GB)

U: 0.00 Mbps (0.00 GB)

- Verify IP address



Current IP Address

52.203.33.143

Add to Favorites

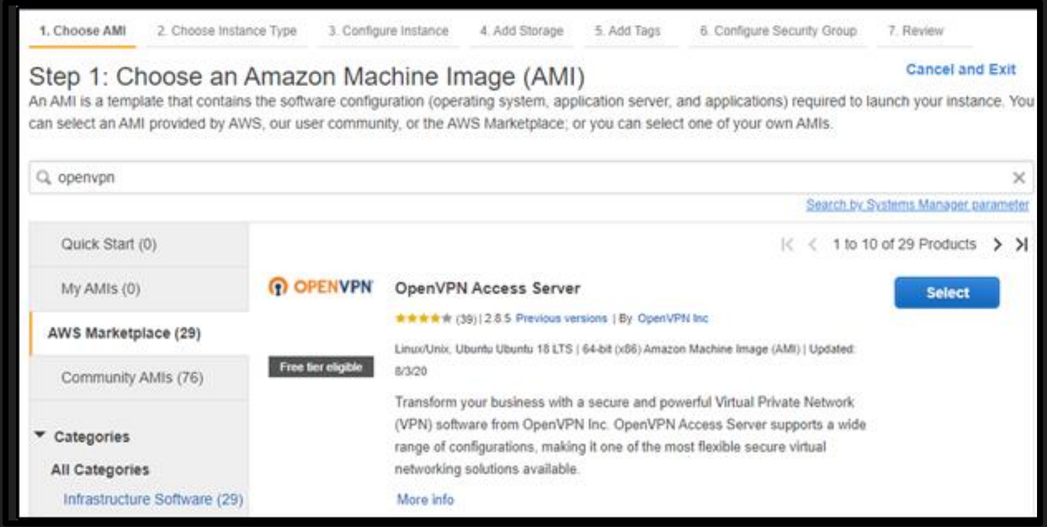
Advanced

- Name Address: ec2-52-203-33-143.compute-1.amazonaws.com
- Remote Port: 53208
- Browser: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/86.0.4240.198 Safari/537.36

(c) 2016 IPChicken - Privacy Policy

OpenVPN Installation

- Create an EC2 Instance



Step 1: Choose an Amazon Machine Image (AMI)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. You can select an AMI provided by AWS, our user community, or the AWS Marketplace; or you can select one of your own AMIs.

Search: openvpn

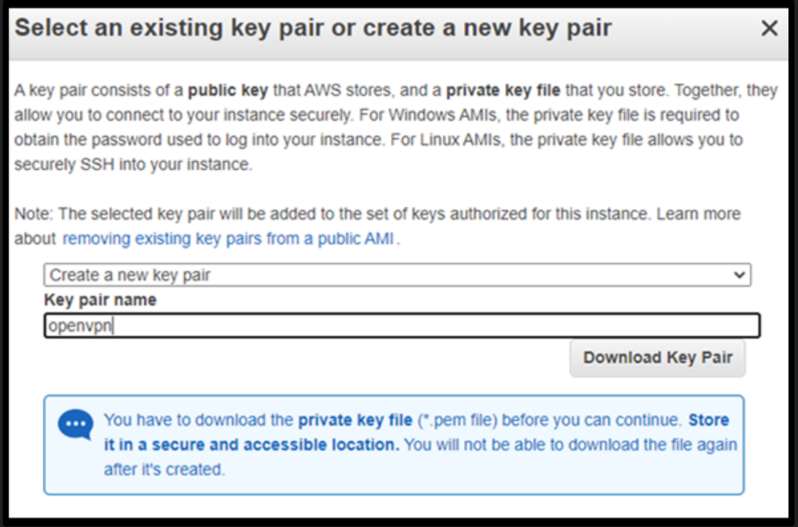
OpenVPN Access Server

★★★★★ (39) | 2.8.5 Previous versions | By OpenVPN Inc

Linux/Unix, Ubuntu Ubuntu 18 LTS | 64-bit (x86) Amazon Machine Image (AMI) | Updated: 8/3/20

Transform your business with a secure and powerful Virtual Private Network (VPN) software from OpenVPN Inc. OpenVPN Access Server supports a wide range of configurations, making it one of the most flexible secure virtual networking solutions available.

- Download key pair



Select an existing key pair or create a new key pair

A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.

Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about removing existing key pairs from a public AMI .

Create a new key pair

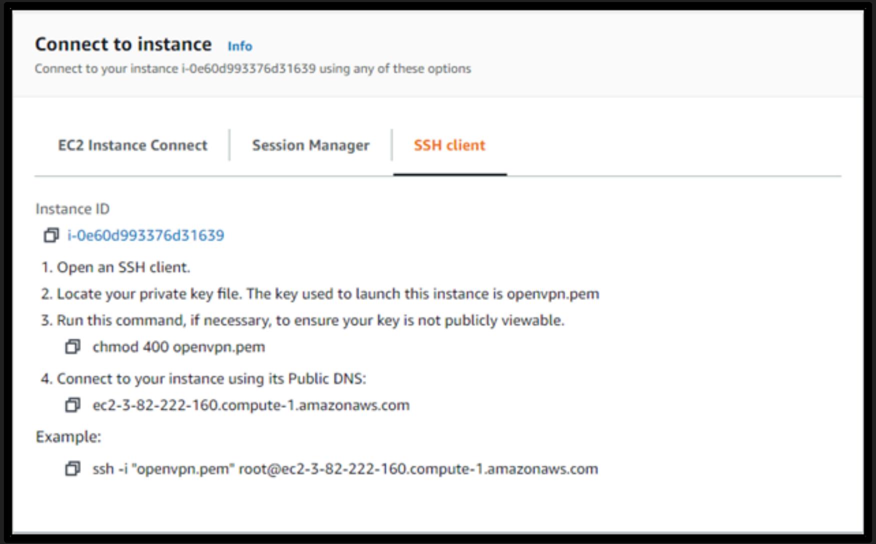
Key pair name

openvpn

You have to download the **private key file** (*.pem file) before you can continue. **Store it in a secure and accessible location.** You will not be able to download the file again after it's created.

OpenVPN Installation

- Connect to instance



Connect to instance Info

Connect to your instance i-0e60d993376d31639 using any of these options

Instance ID

i-0e60d993376d31639

1. Open an SSH client.
2. Locate your private key file. The key used to launch this instance is openvpn.pem
3. Run this command, if necessary, to ensure your key is not publicly viewable.

   chmod 400 openvpn.pem

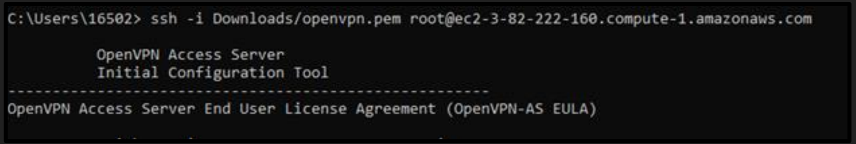
4. Connect to your instance using its Public DNS:

   ec2-3-82-222-160.compute-1.amazonaws.com

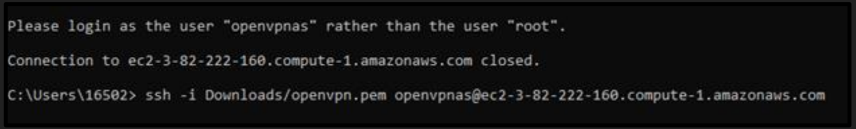
Example:

ssh -i "openvpn.pem" root@ec2-3-82-222-160.compute-1.amazonaws.com

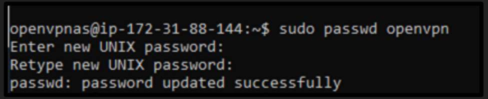
- Connect using SSH



```
C:\Users\16502> ssh -i Downloads/openvpn.pem root@ec2-3-82-222-160.compute-1.amazonaws.com

          OpenVPN Access Server
          Initial Configuration Tool
------------------------------------------------------------
OpenVPN Access Server End User License Agreement (OpenVPN-AS EULA)
```



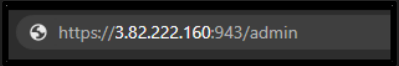
```
Please login as the user "openvpnas" rather than the user "root".

Connection to ec2-3-82-222-160.compute-1.amazonaws.com closed.

C:\Users\16502> ssh -i Downloads/openvpn.pem openvpnas@ec2-3-82-222-160.compute-1.amazonaws.com
```



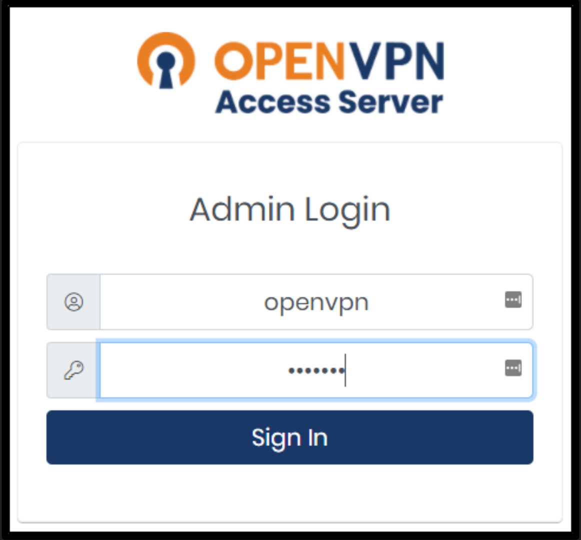
```
openvpnas@ip-172-31-88-144:~$ sudo passwd openvpn
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
```

OpenVPN Installation

- Connect to instance



https://3.82.222.160:943/admin

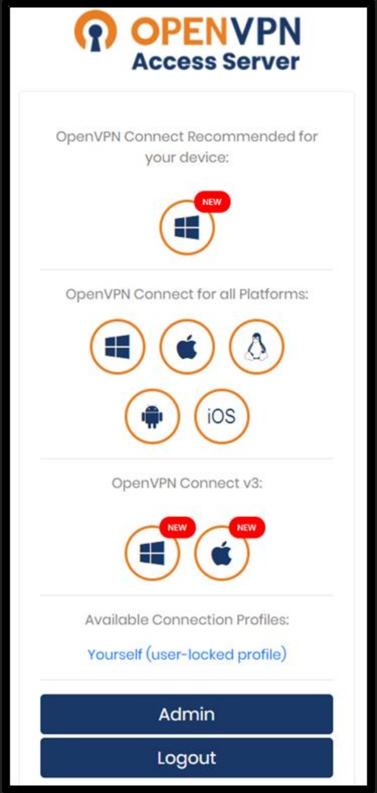


OPENVPN Access Server

Admin Login

openvpn

•••••••



OPENVPN Access Server

OpenVPN Connect Recommended for your device:

OpenVPN Connect for all Platforms:

OpenVPN Connect v3:

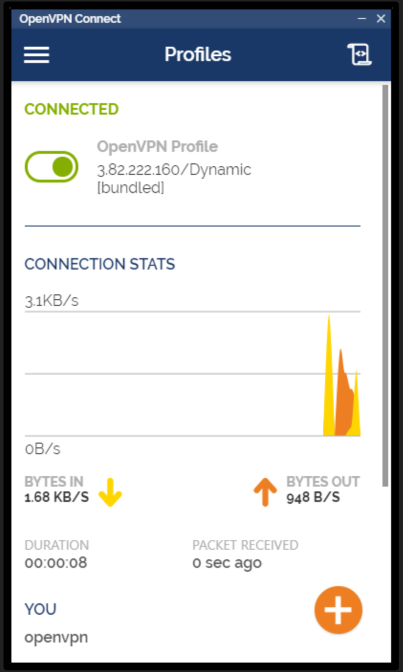
Available Connection Profiles:

Yourself (user-locked profile)

- Download client

OpenVPN Installation

- Connect



OpenVPN Connect

Profiles

CONNECTED

OpenVPN Profile
3.82.222.160/Dynamic
[bundled]

CONNECTION STATS

BYTES IN 1.68 KB/S

BYTES OUT 948 B/S

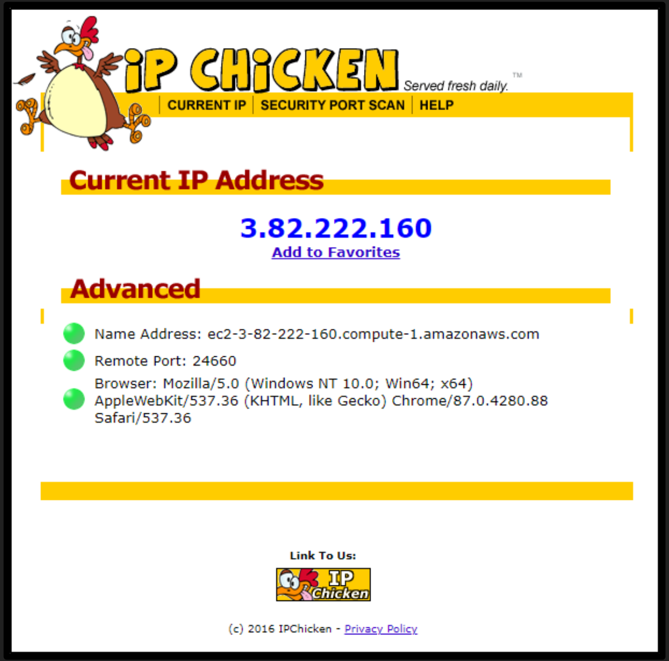
DURATION 00:00:08

PACKET RECEIVED 0 sec ago

YOU

openvpn

- Verify IP address



iP CHiCKEN Served fresh daily.™
CURRENT IP | SECURITY PORT SCAN | HELP

Current IP Address

3.82.222.160

Add to Favorites

Advanced

● Name Address: ec2-3-82-222-160.compute-1.amazonaws.com

● Remote Port: 24660

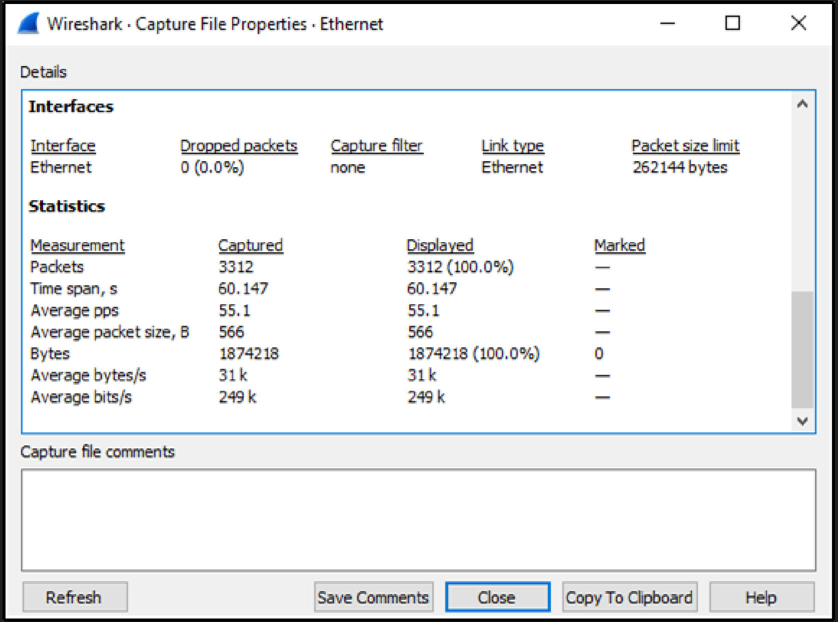
● Browser: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/87.0.4280.88 Safari/537.36

Link To Us:

(c) 2016 IPChicken - Privacy Policy

Wireshark Captures

WireGuard



Wireshark · Capture File Properties · Ethernet

Details

Interfaces

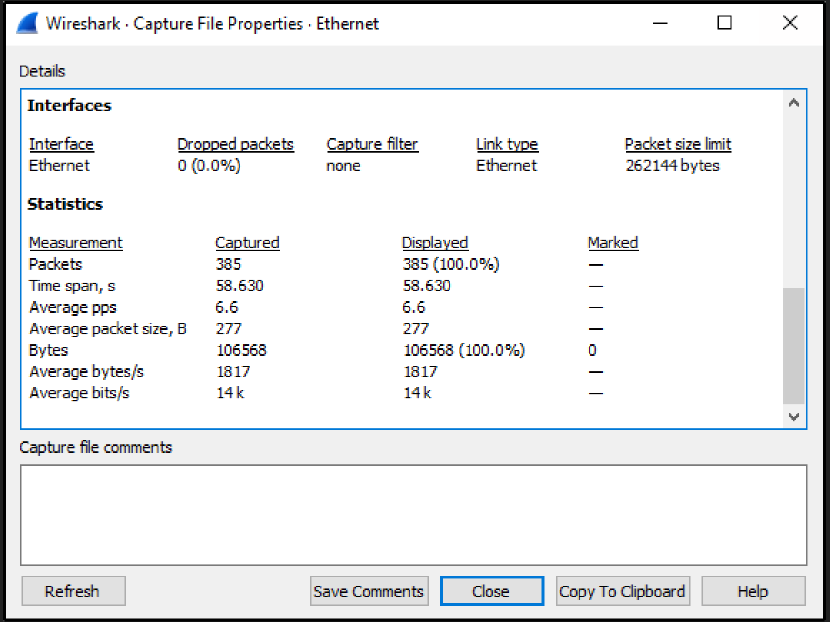
Interface	Dropped packets	Capture filter	Link type	Packet size limit
Ethernet	0 (0.0%)	none	Ethernet	262144 bytes

Statistics

Measurement	Captured	Displayed	Marked
Packets	3312	3312 (100.0%)	—
Time span, s	60.147	60.147	—
Average pps	55.1	55.1	—
Average packet size, B	566	566	—
Bytes	1874218	1874218 (100.0%)	0
Average bytes/s	31 k	31 k	—
Average bits/s	249 k	249 k	—

Capture file comments

OpenVPN



Wireshark · Capture File Properties · Ethernet

Details

Interfaces

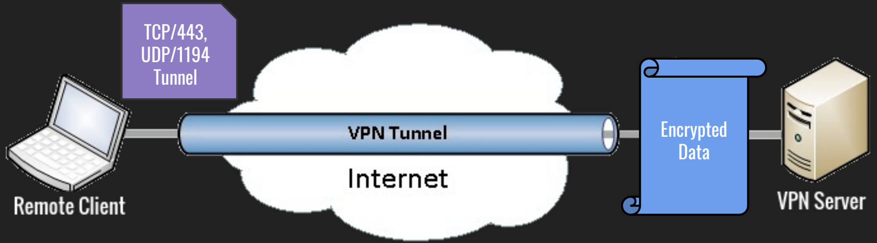
Interface	Dropped packets	Capture filter	Link type	Packet size limit
Ethernet	0 (0.0%)	none	Ethernet	262144 bytes

Statistics

Measurement	Captured	Displayed	Marked
Packets	385	385 (100.0%)	—
Time span, s	58.630	58.630	—
Average pps	6.6	6.6	—
Average packet size, B	277	277	—
Bytes	106568	106568 (100.0%)	0
Average bytes/s	1817	1817	—
Average bits/s	14 k	14 k	—

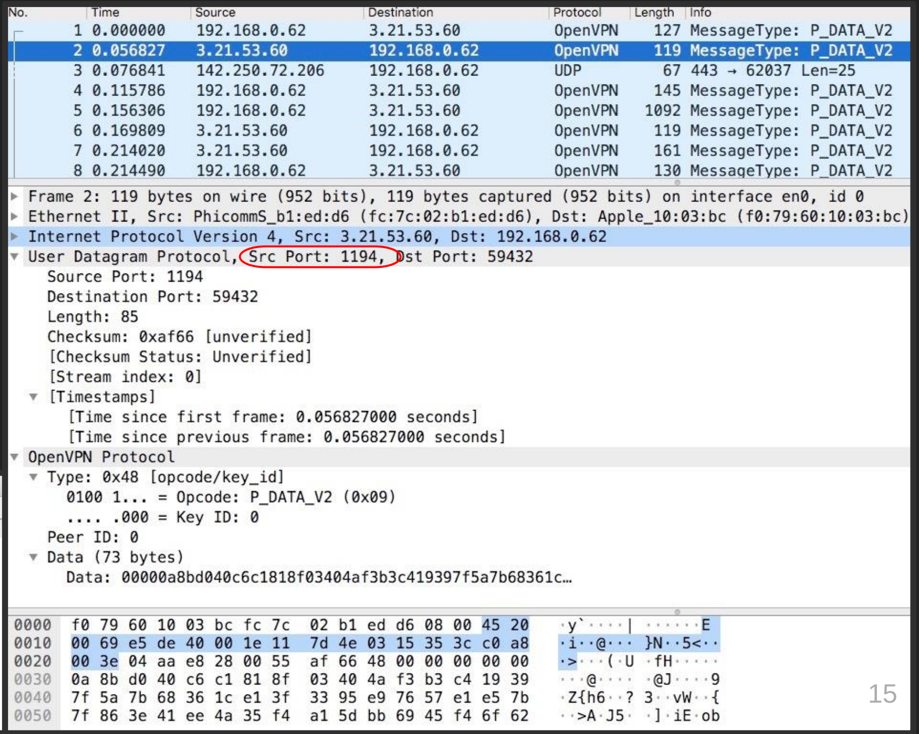
Capture file comments

Protocols



OpenVPN

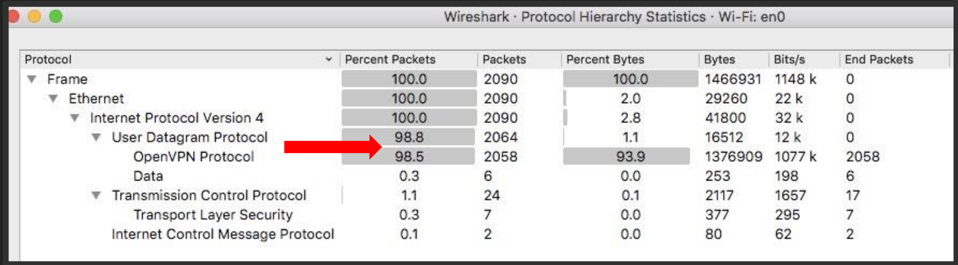
- Implements the OSI layer 2 and 3 for secure network extension. This enables any IP subnetwork to be tunneled over a single UDP or TCP port and depends entirely on the security of OpenSSL.



```
No.  Time      Source          Destination     Protocol  Length Info
   1 0.000000  192.168.0.62    3.21.53.60      OpenVPN     127 MessageType: P_DATA_V2
   2 0.056827  3.21.53.60      192.168.0.62    OpenVPN     119 MessageType: P_DATA_V2
   3 0.076841  142.250.72.206  192.168.0.62    UDP          67 443 → 62037 Len=25
   4 0.115786  192.168.0.62    3.21.53.60      OpenVPN     145 MessageType: P_DATA_V2
   5 0.156306  192.168.0.62    3.21.53.60      OpenVPN    1092 MessageType: P_DATA_V2
   6 0.169809  3.21.53.60      192.168.0.62    OpenVPN     119 MessageType: P_DATA_V2
   7 0.214020  3.21.53.60      192.168.0.62    OpenVPN     161 MessageType: P_DATA_V2
   8 0.214490  192.168.0.62    3.21.53.60      OpenVPN     130 MessageType: P_DATA_V2
```

```
▶ Frame 2: 119 bytes on wire (952 bits), 119 bytes captured (952 bits) on interface en0, id 0
▶ Ethernet II, Src: PhicommS_b1:ed:d6 (fc:7c:02:b1:ed:d6), Dst: Apple_10:03:bc (f0:79:60:10:03:bc)
▶ Internet Protocol Version 4, Src: 3.21.53.60, Dst: 192.168.0.62
▼ User Datagram Protocol, Src Port: 1194, Dst Port: 59432
    Source Port: 1194
    Destination Port: 59432
    Length: 85
    Checksum: 0xaf66 [unverified]
    [Checksum Status: Unverified]
    [Stream index: 0]
  ▼ [Timestamps]
      [Time since first frame: 0.056827000 seconds]
      [Time since previous frame: 0.056827000 seconds]
▼ OpenVPN Protocol
  ▼ Type: 0x48 [opcode/key_id]
      0100 1... = Opcode: P_DATA_V2 (0x09)
      .... .000 = Key ID: 0
    Peer ID: 0
  ▼ Data (73 bytes)
      Data: 00000a8bd040c6c1818f03404af3b3c419397f5a7b68361c…
```

```
0000  f0 79 60 10 03 bc fc 7c  02 b1 ed d6 08 00 45 20   ·y`····|  ······E
0010  00 69 e5 de 40 00 1e 11  7d 4e 03 15 35 3c c0 a8   ·i··@···  }N··5<··
0020  00 3e 04 aa e8 28 00 55  af 66 48 00 00 00 00 00   ·>···(·U  ·fH·····
0030  0a 8b d0 40 c6 c1 81 8f  03 40 4a f3 b3 c4 19 39   ···@····  ·@J····9
0040  7f 5a 7b 68 36 1c e1 3f  33 95 e9 76 57 e1 e5 7b   ·Z{h6··?  3··vW··{
0050  7f 86 3e 41 ee 4a 35 f4  a1 5d bb 69 45 f4 6f 62   ··>A·J5·  ·]·iE·ob
```



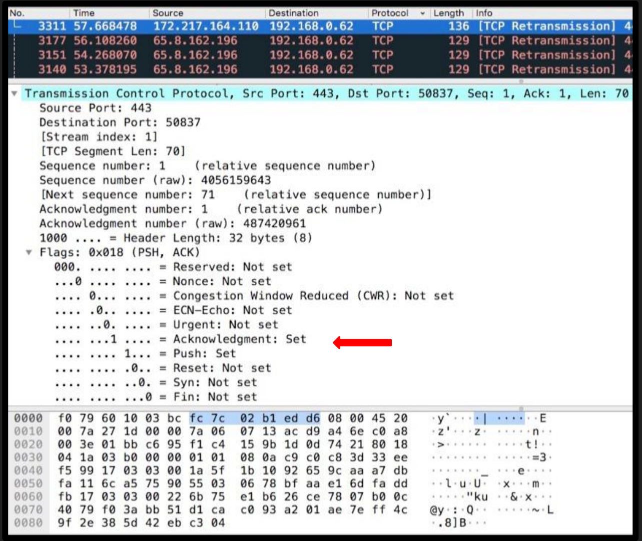
Wireshark · Protocol Hierarchy Statistics · Wi-Fi: en0

Protocol	Percent Packets	Packets	Percent Bytes	Bytes	Bits/s	End Packets
▼ Frame	100.0	2090	100.0	1466931	1148 k	0
▼ Ethernet	100.0	2090	2.0	29260	22 k	0
▼ Internet Protocol Version 4	100.0	2090	2.8	41800	32 k	0
▼ User Datagram Protocol	98.8	2064	1.1	16512	12 k	0
OpenVPN Protocol	98.5	2058	93.9	1376909	1077 k	2058
Data	0.3	6	0.0	253	198	6
▼ Transmission Control Protocol	1.1	24	0.1	2117	1657	17
Transport Layer Security	0.3	7	0.0	377	295	7
Internet Control Message Protocol	0.1	2	0.0	80	62	2

Protocols

TCP (Transmission Control Protocol)

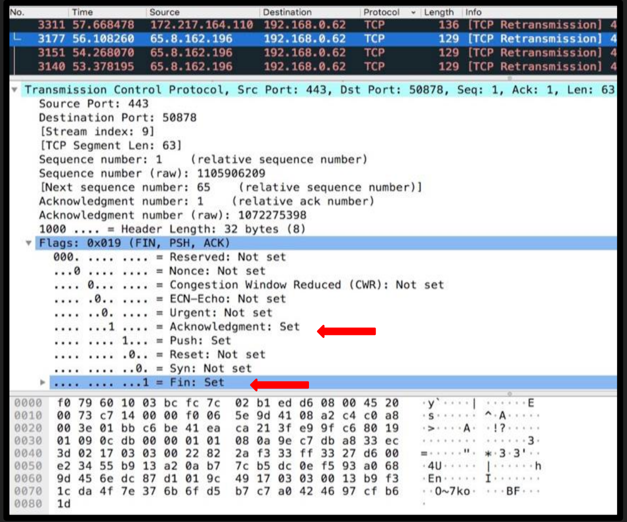
- Specializes in data transmission and is located in layer 4 of the OSI model, where it runs on top of the IP address.
- Works along with an IP address to maintain the connection between the source and the target and ensures all packets are being transmitted without missing any data.



```
No.   Time       Source          Destination   Protocol  Length Info
3311 57.668478 172.217.164.110 192.168.0.62  TCP          136 [TCP Retransmission] 4
3177 56.108260 65.8.162.196    192.168.0.62  TCP          129 [TCP Retransmission] 4
3151 54.268070 65.8.162.196    192.168.0.62  TCP          129 [TCP Retransmission] 4
3140 53.378195 65.8.162.196    192.168.0.62  TCP          129 [TCP Retransmission] 4
```

```
▼ Transmission Control Protocol, Src Port: 443, Dst Port: 50837, Seq: 1, Ack: 1, Len: 70
    Source Port: 443
    Destination Port: 50837
    [Stream index: 1]
    [TCP Segment Len: 70]
    Sequence number: 1    (relative sequence number)
    Sequence number (raw): 4056159643
    [Next sequence number: 71    (relative sequence number)]
    Acknowledgment number: 1    (relative ack number)
    Acknowledgment number (raw): 487420961
    1000 .... = Header Length: 32 bytes (8)
  ▼ Flags: 0x018 (PSH, ACK)
      000. .... .... = Reserved: Not set
      ...0 .... .... = Nonce: Not set
      .... 0... .... = Congestion Window Reduced (CWR): Not set
      .... .0.. .... = ECN-Echo: Not set
      .... ..0. .... = Urgent: Not set
      .... ...1 .... = Acknowledgment: Set
      .... .... 1... = Push: Set
      .... .... .0.. = Reset: Not set
      .... .... ..0. = Syn: Not set
      .... .... ...0 = Fin: Not set
```

```
0000  f0 79 60 10 03 bc fc 7c  02 b1 ed d6 08 00 45 20   ·y`···|  ······E
0010  00 7a 27 1d 00 00 7a 06  07 13 ac d9 a4 6e c0 a8   ·z'···z· ·····n··
0020  00 3e 01 bb c6 95 f1 c4  15 9b 1d 0d 74 21 80 18   ·>······ ····t!··
0030  04 1a 03 b0 00 00 01 01  08 0a c9 c0 c8 3d 33 ee   ········ ·····=3·
0040  f5 99 17 03 03 00 1a 5f  1b 10 92 65 9c aa a7 db   ·······_ ···e····
0050  fa 11 6c a5 75 90 55 03  06 78 bf aa e1 6d fa dd   ··l·u·U· ·x···m··
0060  fb 17 03 03 00 22 6b 75  e1 b6 26 ce 78 07 b0 0c   ·····"ku ··&·x···
0070  40 79 f0 3a bb 51 d1 ca  c0 93 a2 01 ae 7e ff 4c   @y·:·Q·· ·····~·L
0080  9f 2e 38 5d 42 eb c3 04                            ·.8]B···
```



```
No.   Time       Source          Destination   Protocol  Length Info
3311 57.668478 172.217.164.110 192.168.0.62  TCP          136 [TCP Retransmission] 4
3177 56.108260 65.8.162.196    192.168.0.62  TCP          129 [TCP Retransmission] 4
3151 54.268070 65.8.162.196    192.168.0.62  TCP          129 [TCP Retransmission] 4
3140 53.378195 65.8.162.196    192.168.0.62  TCP          129 [TCP Retransmission] 4
```

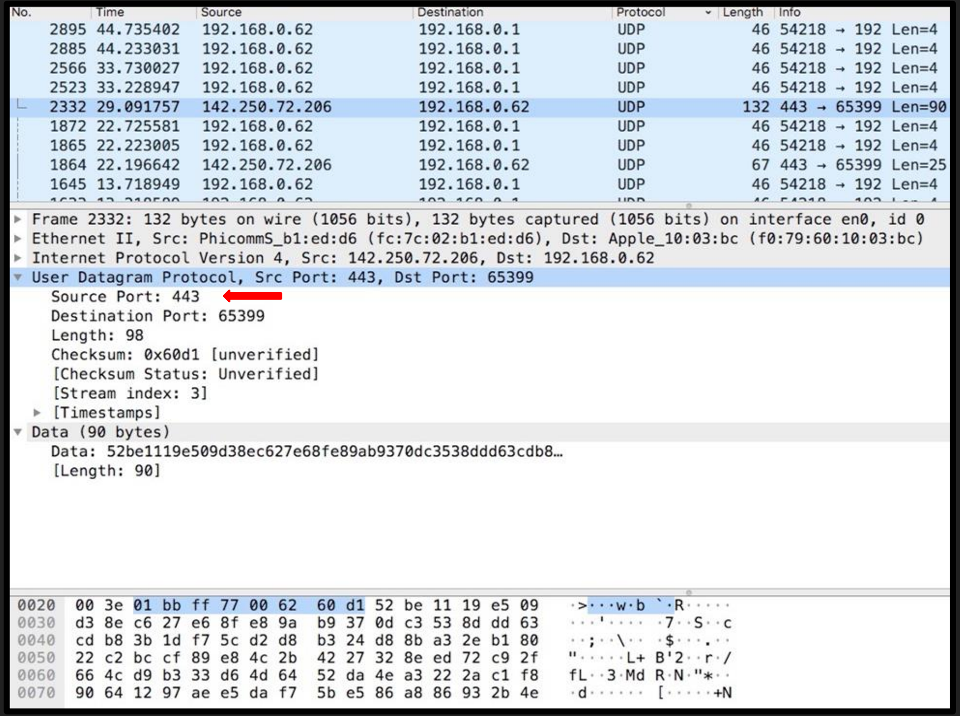
```
▼ Transmission Control Protocol, Src Port: 443, Dst Port: 50878, Seq: 1, Ack: 1, Len: 63
    Source Port: 443
    Destination Port: 50878
    [Stream index: 9]
    [TCP Segment Len: 63]
    Sequence number: 1    (relative sequence number)
    Sequence number (raw): 1105906209
    [Next sequence number: 65    (relative sequence number)]
    Acknowledgment number: 1    (relative ack number)
    Acknowledgment number (raw): 1072275398
    1000 .... = Header Length: 32 bytes (8)
  ▼ Flags: 0x019 (FIN, PSH, ACK)
      000. .... .... = Reserved: Not set
      ...0 .... .... = Nonce: Not set
      .... 0... .... = Congestion Window Reduced (CWR): Not set
      .... .0.. .... = ECN-Echo: Not set
      .... ..0. .... = Urgent: Not set
      .... ...1 .... = Acknowledgment: Set
      .... .... 1... = Push: Set
      .... .... .0.. = Reset: Not set
      .... .... ..0. = Syn: Not set
    ▶ .... .... ...1 = Fin: Set
```

```
0000  f0 79 60 10 03 bc fc 7c  02 b1 ed d6 08 00 45 20   ·y`····|  ······E
0010  00 73 c7 14 00 00 f0 06  5e 9d 41 08 a2 c4 c0 a8   ·s······ ^·A·····
0020  00 3e 01 bb c6 be 41 ea  ca 21 3f e9 9f c6 80 19   ·>····A· ·!?·····
0030  01 09 0c db 00 00 01 01  08 0a 9e c7 db a8 33 ec   ········ ······3·
0040  3d 02 17 03 03 00 22 82  2a f3 33 ff 33 27 d6 00   =·····"· *·3·3'··
0050  e2 34 55 b9 13 a2 0a b7  7c b5 dc 0e f5 93 a0 68   ·4U····· |······h
0060  9d 45 6e dc 87 d1 01 9c  49 17 03 03 00 13 b9 f3   ·En····· I·······
0070  1c da 4f 7e 37 6b 6f d5  b7 c7 a0 42 46 97 cf b6   ··O~7ko· ···BF···
0080  1d                                                 ·
```

Protocols

UDP (User Datagram Protocol)

- A communication protocol that is used to quickly transfer data between two devices and is commonly used for video and voice. This protocol sends transmission data without establishing a connection.
- UDP is faster but less reliable than TCP

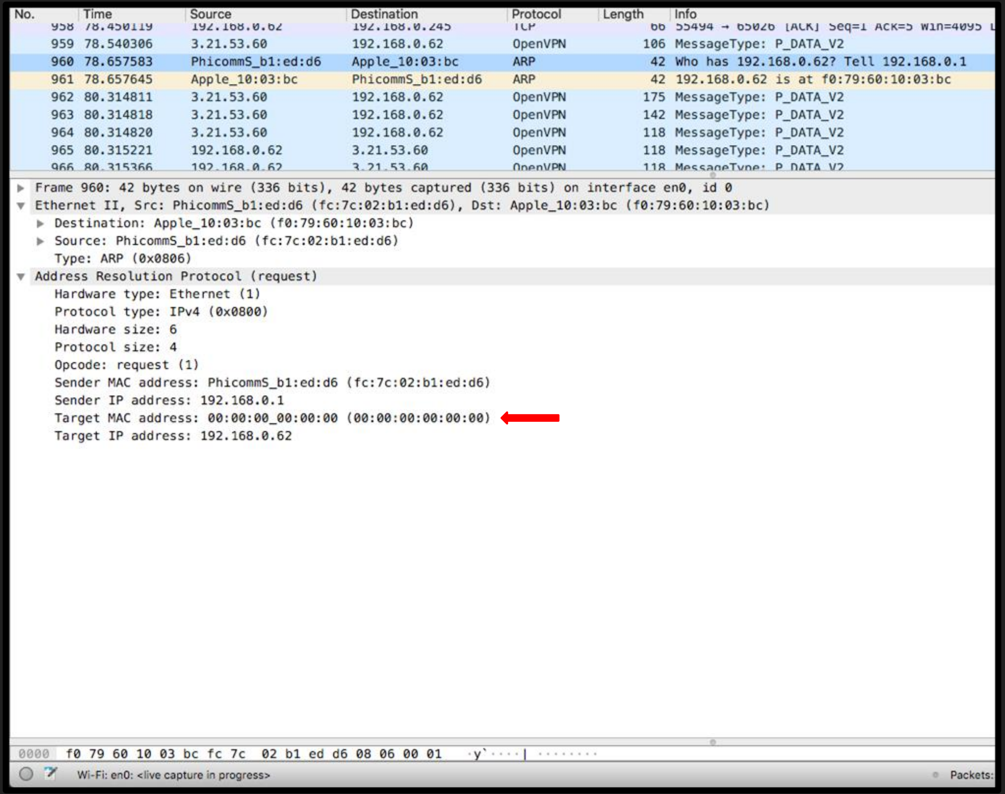


```
No.   Time       Source          Destination   Protocol  Length Info
 2895 44.735402  192.168.0.62    192.168.0.1   UDP           46 54218 → 192 Len=4
 2885 44.233031  192.168.0.62    192.168.0.1   UDP           46 54218 → 192 Len=4
 2566 33.730027  192.168.0.62    192.168.0.1   UDP           46 54218 → 192 Len=4
 2523 33.228947  192.168.0.62    192.168.0.1   UDP           46 54218 → 192 Len=4
 2332 29.091757  142.250.72.206  192.168.0.62  UDP          132 443 → 65399 Len=90
 1872 22.725581  192.168.0.62    192.168.0.1   UDP           46 54218 → 192 Len=4
 1865 22.223005  192.168.0.62    192.168.0.1   UDP           46 54218 → 192 Len=4
 1864 22.196642  142.250.72.206  192.168.0.62  UDP           67 443 → 65399 Len=25
 1645 13.718949  192.168.0.62    192.168.0.1   UDP           46 54218 → 192 Len=4
```

```
▸ Frame 2332: 132 bytes on wire (1056 bits), 132 bytes captured (1056 bits) on interface en0, id 0
▸ Ethernet II, Src: PhicommS_b1:ed:d6 (fc:7c:02:b1:ed:d6), Dst: Apple_10:03:bc (f0:79:60:10:03:bc)
▸ Internet Protocol Version 4, Src: 142.250.72.206, Dst: 192.168.0.62
▾ User Datagram Protocol, Src Port: 443, Dst Port: 65399
    Source Port: 443
    Destination Port: 65399
    Length: 98
    Checksum: 0x60d1 [unverified]
    [Checksum Status: Unverified]
    [Stream index: 3]
  ▸ [Timestamps]
▾ Data (90 bytes)
    Data: 52be1119e509d38ec627e68fe89ab9370dc3538ddd63cdb8…
    [Length: 90]
```

```
0020  00 3e 01 bb ff 77 00 62  60 d1 52 be 11 19 e5 09   ·>···w·b `·R·····
0030  d3 8e c6 27 e6 8f e8 9a  b9 37 0d c3 53 8d dd 63   ···'···· ·7··S··c
0040  cd b8 3b 1d f7 5c d2 d8  b3 24 d8 8b a3 2e b1 80   ··;··\·· ·$···.··
0050  22 c2 bc cf 89 e8 4c 2b  42 27 32 8e ed 72 c9 2f   "·····L+ B'2··r·/
0060  66 4c d9 b3 33 d6 4d 64  52 da 4e a3 22 2a c1 f8   fL··3·Md R·N·"*··
0070  90 64 12 97 ae e5 da f7  5b e5 86 a8 86 93 2b 4e   ·d······ [·····+N
```

Protocols



```
No.    Time       Source               Destination          Protocol  Length Info
   958 78.450119  192.168.0.62         192.168.0.245        TCP           66 55494 → 65026 [ACK] Seq=1 Ack=5 Win=4095 L
   959 78.540306  3.21.53.60           192.168.0.62         OpenVPN      106 MessageType: P_DATA_V2
   960 78.657583  PhicommS_b1:ed:d6    Apple_10:03:bc       ARP           42 Who has 192.168.0.62? Tell 192.168.0.1
   961 78.657645  Apple_10:03:bc       PhicommS_b1:ed:d6    ARP           42 192.168.0.62 is at f0:79:60:10:03:bc
   962 80.314811  3.21.53.60           192.168.0.62         OpenVPN      175 MessageType: P_DATA_V2
   963 80.314818  3.21.53.60           192.168.0.62         OpenVPN      142 MessageType: P_DATA_V2
   964 80.314820  3.21.53.60           192.168.0.62         OpenVPN      118 MessageType: P_DATA_V2
   965 80.315221  192.168.0.62         3.21.53.60           OpenVPN      118 MessageType: P_DATA_V2
   966 80.315366  192.168.0.62         3.21.53.60           OpenVPN      118 MessageType: P_DATA_V2

▶ Frame 960: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface en0, id 0
▼ Ethernet II, Src: PhicommS_b1:ed:d6 (fc:7c:02:b1:ed:d6), Dst: Apple_10:03:bc (f0:79:60:10:03:bc)
  ▶ Destination: Apple_10:03:bc (f0:79:60:10:03:bc)
  ▶ Source: PhicommS_b1:ed:d6 (fc:7c:02:b1:ed:d6)
    Type: ARP (0x0806)
▼ Address Resolution Protocol (request)
    Hardware type: Ethernet (1)
    Protocol type: IPv4 (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: request (1)
    Sender MAC address: PhicommS_b1:ed:d6 (fc:7c:02:b1:ed:d6)
    Sender IP address: 192.168.0.1
    Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00) ←
    Target IP address: 192.168.0.62

0000  f0 79 60 10 03 bc fc 7c  02 b1 ed d6 08 06 00 01   ·y`····| ········
```

ARP (Address Resolution Protocol)

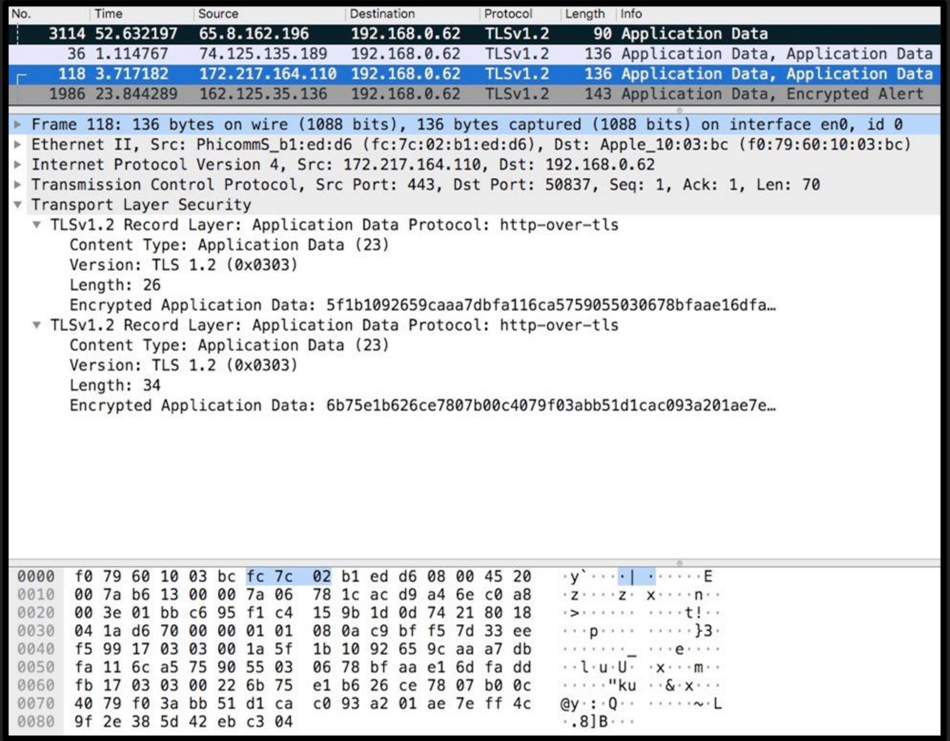
- Functions as a communication map between two devices connecting over the internet. Includes requesting, responding, and storing data of the target IP address.
- ARP works by broadcasting a packet request in order to find the MAC address destination from layer 2 of the OSI model.

Notice: starting with a (request)
Sender Mac Address: & Sender IP address: (are coming from my computer)
Target Mac address: (includes only zeros since we don't have it yet) but when looking the replay should reveal the Mac address in it

Protocols

TLS (Transport Layer Security)

- Located within the application layer and is responsible for securing the data communications over the internet.
- Its primary function is encrypting the communication between the web application and server like the web browsers search the websites.
- This is where the handshake occurs between users and the web server.



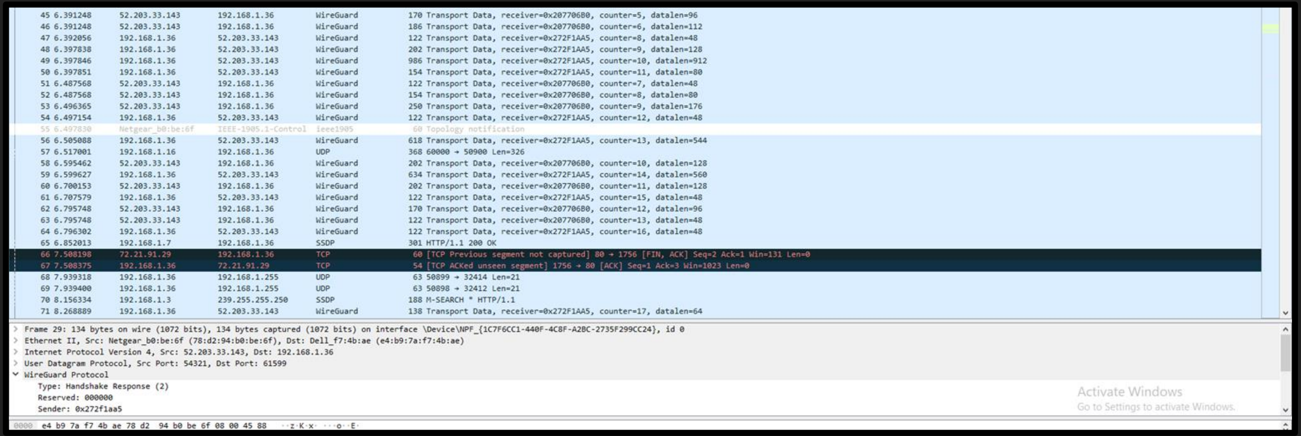
```
No.   Time       Source          Destination   Protocol  Length Info
3114 52.632197  65.8.162.196    192.168.0.62  TLSv1.2     90 Application Data
  36 1.114767   74.125.135.189  192.168.0.62  TLSv1.2    136 Application Data, Application Data
 118 3.717182   172.217.164.110 192.168.0.62  TLSv1.2    136 Application Data, Application Data
1986 23.844289  162.125.35.136  192.168.0.62  TLSv1.2    143 Application Data, Encrypted Alert
```

```
▸ Frame 118: 136 bytes on wire (1088 bits), 136 bytes captured (1088 bits) on interface en0, id 0
▸ Ethernet II, Src: PhicommS_b1:ed:d6 (fc:7c:02:b1:ed:d6), Dst: Apple_10:03:bc (f0:79:60:10:03:bc)
▸ Internet Protocol Version 4, Src: 172.217.164.110, Dst: 192.168.0.62
▸ Transmission Control Protocol, Src Port: 443, Dst Port: 50837, Seq: 1, Ack: 1, Len: 70
▾ Transport Layer Security
  ▾ TLSv1.2 Record Layer: Application Data Protocol: http-over-tls
      Content Type: Application Data (23)
      Version: TLS 1.2 (0x0303)
      Length: 26
      Encrypted Application Data: 5f1b1092659caaa7dbfa116ca5759055030678bfaae16dfa…
  ▾ TLSv1.2 Record Layer: Application Data Protocol: http-over-tls
      Content Type: Application Data (23)
      Version: TLS 1.2 (0x0303)
      Length: 34
      Encrypted Application Data: 6b75e1b626ce7807b00c4079f03abb51d1cac093a201ae7e…
```

```
0000  f0 79 60 10 03 bc fc 7c  02 b1 ed d6 08 00 45 20   ·y`···|· ······E
0010  00 7a b6 13 00 00 7a 06  78 1c ac d9 a4 6e c0 a8   ·z····z· x····n··
0020  00 3e 01 bb c6 95 f1 c4  15 9b 1d 0d 74 21 80 18   ·>······ ····t!··
0030  04 1a d6 70 00 00 01 01  08 0a c9 bf f5 7d 33 ee   ···p···· ·····}3·
0040  f5 99 17 03 03 00 1a 5f  1b 10 92 65 9c aa a7 db   ·······_ ···e····
0050  fa 11 6c a5 75 90 55 03  06 78 bf aa e1 6d fa dd   ··l·u·U· ·x···m··
0060  fb 17 03 03 00 22 6b 75  e1 b6 26 ce 78 07 b0 0c   ·····"ku ··&·x···
0070  40 79 f0 3a bb 51 d1 ca  c0 93 a2 01 ae 7e ff 4c   @y·:·Q·· ·····~·L
0080  9f 2e 38 5d 42 eb c3 04                            ·.8]B···
```

Notice the data application is encrypted

Protocols



```
45 6.391248   52.203.33.143      192.168.1.36        WireGuard     170 Transport Data, receiver=0x207706B0, counter=5, datalen=96
46 6.391248   52.203.33.143      192.168.1.36        WireGuard     186 Transport Data, receiver=0x207706B0, counter=6, datalen=112
47 6.392056   192.168.1.36       52.203.33.143       WireGuard     122 Transport Data, receiver=0x272F1AA5, counter=8, datalen=48
48 6.397838   192.168.1.36       52.203.33.143       WireGuard     202 Transport Data, receiver=0x272F1AA5, counter=9, datalen=128
49 6.397846   192.168.1.36       52.203.33.143       WireGuard     986 Transport Data, receiver=0x272F1AA5, counter=10, datalen=912
50 6.397851   192.168.1.36       52.203.33.143       WireGuard     154 Transport Data, receiver=0x272F1AA5, counter=11, datalen=80
51 6.487568   52.203.33.143      192.168.1.36        WireGuard     122 Transport Data, receiver=0x207706B0, counter=7, datalen=48
52 6.487568   52.203.33.143      192.168.1.36        WireGuard     154 Transport Data, receiver=0x207706B0, counter=8, datalen=80
53 6.496365   52.203.33.143      192.168.1.36        WireGuard     250 Transport Data, receiver=0x207706B0, counter=9, datalen=176
54 6.497154   192.168.1.36       52.203.33.143       WireGuard     122 Transport Data, receiver=0x272F1AA5, counter=12, datalen=48
55 6.497830   Netgear_b0:be:6f   IEEE-1905.1-Control ieee1905       60 Topology notification
56 6.505088   192.168.1.36       52.203.33.143       WireGuard     618 Transport Data, receiver=0x272F1AA5, counter=13, datalen=544
57 6.517001   192.168.1.16       192.168.1.36        UDP           368 60000 → 50900 Len=326
58 6.595462   52.203.33.143      192.168.1.36        WireGuard     202 Transport Data, receiver=0x207706B0, counter=10, datalen=128
59 6.599627   192.168.1.36       52.203.33.143       WireGuard     634 Transport Data, receiver=0x272F1AA5, counter=14, datalen=560
60 6.700153   52.203.33.143      192.168.1.36        WireGuard     202 Transport Data, receiver=0x207706B0, counter=11, datalen=128
61 6.707579   192.168.1.36       52.203.33.143       WireGuard     122 Transport Data, receiver=0x272F1AA5, counter=15, datalen=48
62 6.795748   52.203.33.143      192.168.1.36        WireGuard     170 Transport Data, receiver=0x207706B0, counter=12, datalen=96
63 6.795748   52.203.33.143      192.168.1.36        WireGuard     122 Transport Data, receiver=0x207706B0, counter=13, datalen=48
64 6.796302   192.168.1.36       52.203.33.143       WireGuard     122 Transport Data, receiver=0x272F1AA5, counter=16, datalen=48
65 6.852013   192.168.1.7        192.168.1.36        SSDP          301 HTTP/1.1 200 OK
66 7.508198   72.21.91.29        192.168.1.36        TCP            60 [TCP Previous segment not captured] 80 → 1756 [FIN, ACK] Seq=2 Ack=1 Win=131 Len=0
67 7.508375   192.168.1.36       72.21.91.29         TCP            54 [TCP ACKed unseen segment] 1756 → 80 [ACK] Seq=1 Ack=3 Win=1023 Len=0
68 7.939318   192.168.1.36       192.168.1.255       UDP            63 50899 → 32414 Len=21
69 7.939400   192.168.1.36       192.168.1.255       UDP            63 50898 → 32412 Len=21
70 8.156334   192.168.1.3        239.255.255.250     SSDP          188 M-SEARCH * HTTP/1.1
71 8.268889   192.168.1.36       52.203.33.143       WireGuard     138 Transport Data, receiver=0x272F1AA5, counter=17, datalen=64

> Frame 29: 134 bytes on wire (1072 bits), 134 bytes captured (1072 bits) on interface \Device\NPF_{1C7F6CC1-440F-4C8F-A2BC-2735F299CC24}, id 0
> Ethernet II, Src: Netgear_b0:be:6f (78:d2:94:b0:be:6f), Dst: Dell_f7:4b:ae (e4:b9:7a:f7:4b:ae)
> Internet Protocol Version 4, Src: 52.203.33.143, Dst: 192.168.1.36
> User Datagram Protocol, Src Port: 54321, Dst Port: 61599
v WireGuard Protocol
    Type: Handshake Response (2)
    Reserved: 000000
    Sender: 0x272f1aa5

0000  e4 b9 7a f7 4b ae 78 d2  94 b0 be 6f 08 00 45 88   ··z·K·x· ···o··E·
```

Activate Windows
Go to Settings to activate Windows.

WireGuard

- A simpler, safer, faster, and stable VPN protocol that uses UDP for transport.
- Functions on layer 3 of the OSI model
- Uses newer algorithms for encryption, including:
    - ChaCha20 for encryption
    - Curve25519 for key exchange
    - BLAKE2s for hashing
    - SipHash24 for hashable keys
    - Poly1305 for data authentication

VPN Protocols

PPTP (Point-to-Point Tunneling Protocol)

- Operates on TCP port 1723 and is one of the oldest VPN protocols still in use.
- Used for audio and video streaming on older devices.
- Lacks security.

L2TP/IPSec (Layer 2 Tunnel Protocol)

- An upgrade for L2F and PPTP, but lacks encryption and authentication.
- Usually requires IPSec for security.
  - Provides end-to-end security that solves the problems of L2TP
  - IPsec encrypts and authenticates every IP packet. L2TP/IPsec is more secure than PPTP.

VPN Protocols

SSTP (Secure Socket Tunneling Protocol)

- Developed by Microsoft, meant to provide secure online data and traffic which is supposed to surpass PPTP and L2TP/IPsec. SSTP creates a secure connection within the VPN client and server and all data that goes through this tunnel is encrypted. Servers are authenticated when a connection is taken place because of SSL/TLS. SSTP uses TCP port 443 and only supports user authentication.

IKEv2 (Internet Key Exchange version 2)

- Based on IPsec tunneling that has a secure VPN channel. Version 2 has 256-bit data encryption and also uses IPsec for security, with a more stable connection and better speed. IKEv2 will not act until it recognizes user identity, which helps prevent attacks.

Comparison

WireGuard

- Released in 2019
- 4,000 lines of code
- Not Crypto-agile
- Security, less complexity
- Fast performance

OpenVPN

- Released in 2001
- 70,000+ lines of code
- Crypto-agile
- Security, more complexity
- Moderate performance

Results

No VPN vs WireGuard

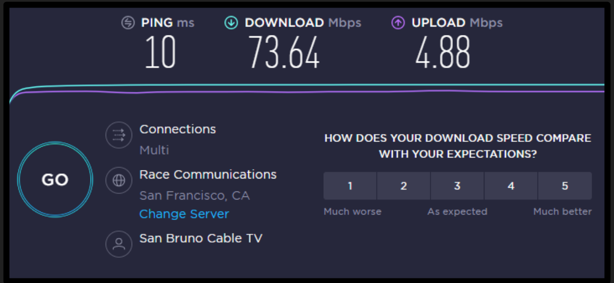
- WireGuard saw a 5.4% decrease in download speed compared to no VPN
- WireGuard saw a 3.9% decrease in upload speed compared to no VPN

No VPN vs OpenVPN

- OpenVPN saw a 16.2% decrease in download speed compared to no VPN
- OpenVPN saw a 4.9% decrease in upload speed compared to no VPN

WireGuard vs OpenVPN

- OpenVPN saw a 11.5% decrease in download speed compared to WireGuard
- OpenVPN saw a 1.1% decrease in upload speed compared to WireGuard



PING ms 10 DOWNLOAD Mbps 73.64 UPLOAD Mbps 4.88

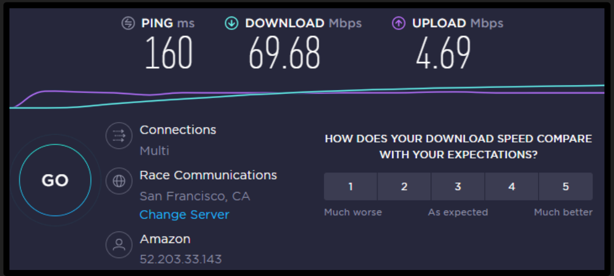
Connections
Multi

Race Communications
San Francisco, CA
Change Server

San Bruno Cable TV

HOW DOES YOUR DOWNLOAD SPEED COMPARE WITH YOUR EXPECTATIONS?

1 2 3 4 5
Much worse As expected Much better



PING ms 160 DOWNLOAD Mbps 69.68 UPLOAD Mbps 4.69

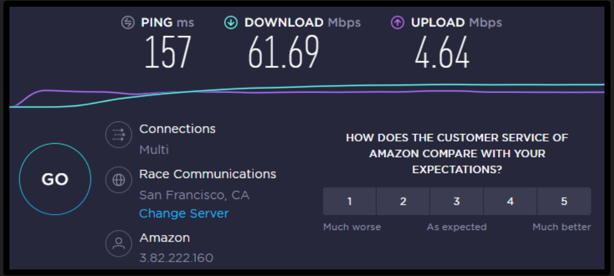
Connections
Multi

Race Communications
San Francisco, CA
Change Server

Amazon
52.203.33.143

HOW DOES YOUR DOWNLOAD SPEED COMPARE WITH YOUR EXPECTATIONS?

1 2 3 4 5
Much worse As expected Much better



PING ms 157 DOWNLOAD Mbps 61.69 UPLOAD Mbps 4.64

Connections
Multi

Race Communications
San Francisco, CA
Change Server

Amazon
3.82.222.160

HOW DOES THE CUSTOMER SERVICE OF AMAZON COMPARE WITH YOUR EXPECTATIONS?

1 2 3 4 5
Much worse As expected Much better

Recommendation

- If performance and using the newest protocols are the primary concern, WireGuard would be our recommendation
- If you want something tested and reliable, with ease of setup, due to an available Amazon Machine Image (AMI) on the AWS Marketplace, then OpenVPN would be our recommendation

Thank you!