Network Protocols - Revision

Luke Anderson

luke@lukeanderson.com.au

10th May 2019

University Of Sydney

Overview

1. Crypto-Bulletin

2. The Layers

2.1 OSI Model

2.2 Layer 1: Physical

2.3 Layer 2: Data Link

MAC Addresses

2.4 Layer 3: Network

IP Addresses

Subnets

Private Networks

2.5 Routing

2.6 NAT

2.7 VLANs

2.8 Layer 4: Transport Layer

TCP

UDP

Crypto-Bulletin

Crypto-Bulletin

Crypto exchange Binance raided, 7000 Bitcoin lost
https://www.itnews.com.au/news/crypto-exchange-binance-raided-7000-bitcoin-lost-524804

Unsecure git repo clones behind data-wiping ransom attacks
https://www.itnews.com.au/news/unsecure-git-repo-clones-behind-data-wiping-ransom-attacks-524664

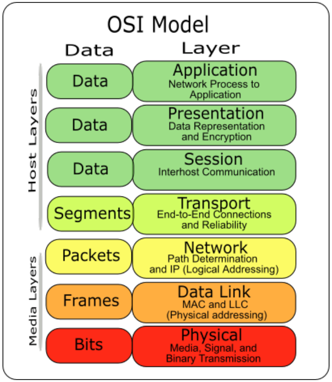
How to Weaponize the Yubikey
https://www.blackhillsinfosec.com/how-to-weaponize-the-yubikey/

The Layers

Network Revision: OSI Model

The OSI Model is a way to conceptualise network systems, whereby each layer is an abstraction from the previous.

You must be quite familiar with this model in order to properly understand interactions between the technologies we discuss.


OSI Model
Data
Layer
Host Layers
Media Layers
Data
Application
Network Process to Application
Data
Presentation
Data Representation and Encryption
Data
Session
Interhost Communication
Segments
Transport
End-to-End Connections and Reliability
Packets
Network
Path Determination and IP (Logical Addressing)
Frames
Data Link
MAC and LLC (Physical addressing)
Bits
Physical
Media, Signal, and Binary Transmission

The OSI Model - Wikimedia

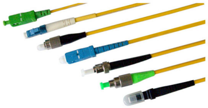
Layer 1: Physical Layer

Layer 1 is the physical layer.

It deals with **bits** and how they are modulated, multiplexed, and encoded onto the medium.

(a) Wireless

(b) Copper Twisted Pair


(c) Optic Fibre

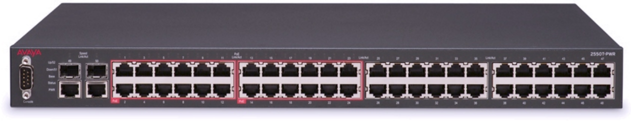
Layer 2: Data Link Layer

Layer 2 is the data link layer.

It deals with **frames** and is concerned with how data is transmitted between devices on the same LAN.

It provides services such as:

- Data transfer
- Error checking
- Collision detection



A network switch forwards frames

Layer 2: MAC Addresses

Devices communicate on layer 2 by use of MAC addresses.

Every device has a unique MAC address, which is programmed in the factory.

Different manufacturers have different MAC ranges allocated to them, which can allow remote identification of hardware.

Address Resolution Protocol (ARP)

Before sending data on a LAN, a sending device must identify the MAC address of the destination device.

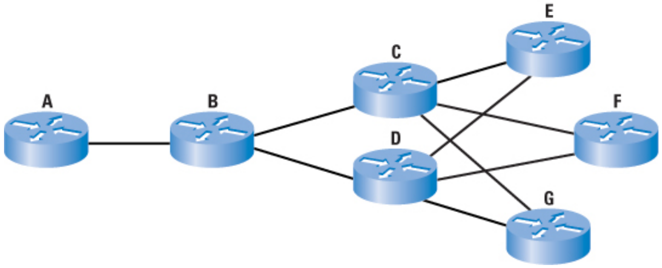
If the sender only has an IP address, they can use ARP to determine the corresponding MAC address.

Layer 3: Network Layer

Layer 3 is the network layer.

It deals with packets and is concerned with how data is routed between networks.

The network layer involves IP addresses and determines how packets can be routed via the most efficient route between two points.



Network routers forward packets

Layer 3: IP Addresses

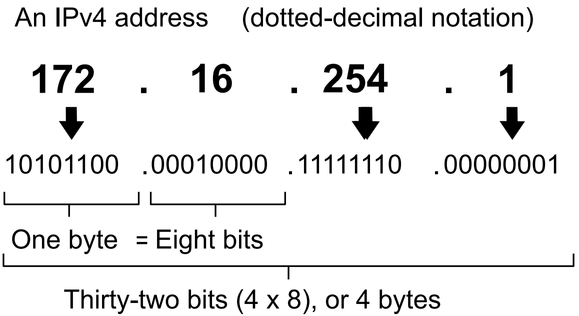
An IP address is a 32-bit or 128-bit number, which uniquely identifies a device on a network.

There are two types:

IPv4: 172.16.254.1
The original 32-bit addresses commonly seen today.
Dot-decimal notation splits the 32-bit number into 4 bytes, each represented as a decimal number.

IPv6: 2001:db8:0:1234:0:567:8:1
A much longer 128-bit address, to cater for the growth of the internet and exhaustion of available IPv4 addresses.

Layer 3: IPv4 Addresses



IPv4 address Wikimedia

An IPv4 address contains two separate addresses:

Network: defines what network a particular host is

Host: defines the number of a host in a particular network

The *subnet mask* identifies the boundary between the two.

Layer 3: Subnets

In your home LAN, you probably only have one subnet.

In larger networks, it is necessary to break the network into smaller subnetworks.

This can assist with:

- Scalability
- Communication Overheads
- Task Seggregation
- Security

Layer 3: Subnet Addresses

When breaking a network up into subnets, we break an IP address into two components:

- **Network Address**
  The *beginning* part of the IP address
- **Host Address**
  The *end* part of the IP address

Where is the *beginning* and *end* of the IP address?
That is defined by the **subnet mask**

```
192.168.000.010
255.255.255.000
```

or

```
192.168.000.010/24
```

[1]

[1] CIDR notation is much more compact and used more frequently.

Layer 3: Subnet Mask

The subnet mask is a set of bits that is the same length as an IP address. The '`1`' bits specify the bits of the IP address that apply to the network, and the '`0`' bits define the host.

For example:

```
192.168.1.150/24
255.255.255.0
```

The network here is: `192.168.1.0/24`, and it is host number `150` in that network.

You do not need to specify the CIDR notation "`/24`" *and* the subnet mask, since they are redundant.

There is no host "`0`" or "`255`", since that is reserved for the *network identifier* and *broadcast address* respectively.

If you are in one network, you cannot talk to hosts on another network without having your traffic pass through a *router*.

Layer 3: Special Subnet Addresses

When talking about a particular subnet, the *first* and *last* IP address in the range are special.

For example, with `192.168.1.0/24`:

Network: `192.168.1.0`
The subnet itself is described by the first IP address inside the range.
A device cannot assume this address as it signifies the whole network, not a particular device on it.[2]

Broadcast: `192.168.1.255`
The broadcast address signifies that the packet should be transmitted to all hosts on the network.
A device cannot assume the broadcast address.[3]

[2]NB: `192.168.1.0/16` is still a valid IP address in the `192.168.0.0/16` network.

[3]NB: `192.168.1.255/16` is not a broadcast address, it would be `192.168.255.255/16`

Layer 3: Subnet Mask

Some more subnet mask examples, with the network address in red and the host address in blue.

```
192.168.53.231/18
11000000.10101000.00110101.11100111

255.255.192.0
11111111.11111111.11000000.0000000
```

```
10.0.0.15/9
00001010.00000000.00000000.00001111

255.128.0.0
11111111.10000000.0000000.0000000
```

See Classless Inter-Domain Routing (CIDR) for more information.

Layer 3: Private Networks

The IANA has reserved the following IP ranges for *private networks*, as defined in RFC 1918.

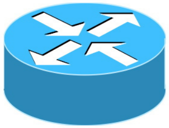
CIDR block (subnet mask)	**Num Addr.**	**IP range**
10.0.0.0/8 (255.0.0.0)	16,777,216	10.0.0.0 - 10.255.255.255
172.16.0.0/12 (255.240.0.0)	1,048,576	172.16.0.0 - 172.31.255.255
192.168.0.0/16 (255.255.0.0)	65,536	192.168.0.0 - 192.168.255.255

Any IP addresses that fall within these ranges are **not available on the internet**, and are reserved for private networks only.

Routing

In order for hosts on different networks to communicate, packets must be passed between networks by a *router*.

A router is simply a computer with more than one network interface that has rules about *forwarding* traffic between them.


The conventional pictographic for a router.

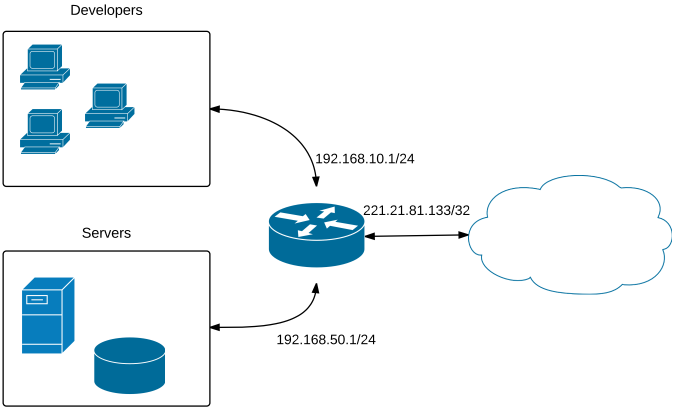
Rules are listed in *routing tables*[4] which can become very complex, potentially going into fine-grained detail like which TCP ports are permitted for communication between particular computers in different networks.

[4]These tables can also be referred to as *firewalls* or an ACL, depending on the context. These terms are more general however and could refer to different components that are not related to routing.

Routing: Example

For example, you might have a scenario where a router passes traffic between three networks:

`192.168.10.0/24`:	"developer" subnet
`192.168.50.0/24`:	"server" subnet
`0.0.0.0/32`:	the Internet



Network Address Translation (NAT)

In our previous slide, the router's public internet-facing IP address was `221.21.81.133`. When someone from the developer network, say `192.168.10.105`, wants to access Google's public DNS server at `8.8.8.8`, they will need to ask their *default gateway*, which is the router's developer network IP address: `192.168.10.1`.

So that the developer can receive a response from Google, the router rewrites the *source* IP address (`192.168.10.105`) to it's public IP (`221.21.81.133`) so that Google knows where to send the response over the public internet.

When the router receives the response, the 'destination' in the IP packet will be the router's public IP address. The router remembers the computer that sent the original request, so that it can rewrite the destination address to be the original host (`192.168.10.105`) and passes Google's response back to the originator.

This process of Network Address Translation (NAT) is extremely common in home networks.

NAT: Port Forwarding

In the case that a network service is to be accessible from the internet, there is no originating request to tell the router where the packet should be rewritten to. In this case we need to use NAT's *port forwarding* feature.

This may be useful if we have a file server on the "servers" subnet, `192.168.50.20`. We may need to configure the router's NAT so that any incoming request on it's public facing IP that arrives on port 21 (FTP), is always forwarded to `192.168.50.20`.

This system is often called a "firewall", though in this case only specifies *inbound* rules, not *outbound* rules. It's a kind of implicit firewall whereby incoming packets which don't have an originating request and don't have a port forwarding rule, are simply dropped.

Virtual LANs

If everyone was connected to the same switch, but were allocated to different subnets, it would be possible for a host to simply change their IP address assume a different subnet. This arrangement is called a “flat” switch and presents a security vulnerability.

To properly separate network hosts and prevent them from assuming different subnets, we need to connect them to separate switches...or do we. The concept of a Virtual LAN allows us to easily seggregate hosts on layer 2.

VLANs on Switches

In the network switch, we can define different ports to have different VLAN IDs.

A different VLAN ID effectively causes a port to be on a different virtual switch, similarly all ports with the same VLAN ID are on the same virtual switch.

With VLANs, traffic on certain physical switch ports can be “tagged” as being part of a VLAN or virtual switch. Physical switches will then ensure traffic cannot pass between different VLANs.

Virtual LANs

Segregation with VLANs is much simpler than with layer 3. Frames are simply tagged with a number (usually by using 802.1Q) that determines what LAN the frame belongs to.

Frames from multiple segregated VLANs can then be combined on a single link (e.g. cable), which is called a “trunk”.

VLANs are used extensively in industry and a working knowledge of them is essential for any security professional.

Layer 4: Transport Layer

Layer 4 is the transport layer and defines how packets are sent between two endpoints which are not necesissarily on the same network segment (e.g. between two IP addresses on the internet).

The most common protocols on layer 4 are:

TCP: Transmission Control Protocol

UDP: User Datagram Protocol

- A best-effort connectionless protocol
- Sends packets to the destination with no additional checks or delivery guarantees
- Operates using UDP **datagrams**

Layer 4: Transmission Control Protocol (TCP)

The Transmission Control Protocol (TCP) lives on layer 4 of the OSI Model. It is not responsible for addressing or segregation like the layers below it, but is instead responsible for delivery of packets.

Properties of TCP:

- Establishes a connection using the *TCP 3-way handshake*
- Ensures in-order delivery of data for the application layer(s) above it.
- Retransmits lost packets
- Has a scalable ‘sliding window’ which is responsible for achieving the fastest possible transfer speed, while avoiding congestion.[5]
- Operates using TCP **segments**

[5] This why downloads take a few seconds to “get up to speed”

Layer 4: TCP Ports

TCP segments are assigned to particular “ports”, which define what application they are intended for.

Any application can run on any port between 1 - 65535 (16-bit port number).

Usually there are two TCP port numbers:

- The destination port: usually a low-numbered standard port number.
- The source port: usually high-numbered port that handles the response.

Common Port Numbers

Many applications have conventional TCP ports, such as HTTP being on port 80.

You should commit the conventional ports for common protocols to memory. The following are some common protocols for which you should should know the port:
HTTP, HTTPS, FTP, SSH, DNS, POP3, SMTP.

Layer 4: User Datagram Protocol (UDP)

The User Datagram Protocol (UDP) also lives on layer 4 with TCP, however it does not guarantee delivery at all.

UDP is a “best effort” protocol connection-less, which sends “datagrams” instead of TCP’s “segments”.

With UDP, packets are not acknowledged or re-transmitted (unless performed explicitly at the application layer).

UDP also operates on a particular “port”, though a TCP and UDP port are considered different. E.g. DNS operates on UDP port 53, not TCP.