

DAILY ASSESSMENT

Date:	21/07/2020	Name:	Davis S. Patel
Course:	Network Security & Database Vulnerabilities	USN:	4AL16EC045
Topic:	Week 2	Semester & Section:	8 th - A
GitHub Repository:	Davis		

FORENOON SESSION DETAILS

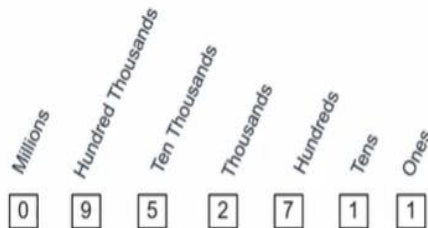
Image of session

Binary System

- Human beings think in terms of 10 based (decimal) numbering because we have 10 fingers/toes.
- Computers only know if the power is on (1) or off (0).
- Computers they don't know any other state.



Decimal



REPORT –

Most applications out there run on the HTTP protocol, so having a solid understanding of this protocol will make your testing work much more manageable. We explored this in a previous post: [What is HTTP protocol – introduction to HTTP for Testers](#). But there's more to networks than just HTTP. In this post, we are going to dive deeper into networks by exploring the OSI model.

LAN is a local network that consists of a group of computers and devices connected via a single physical network (cables). It is limited to a specific geographic area/location.

An excellent example of this kind of network would be a library, office, or home. I don't think most of us use a LAN in our homes these days, because a LAN connects devices via cables. Nowadays, our devices are connected wirelessly via WIFI, so we're talking about WLAN.

WAN combines numerous sites and covers large geographic regions (connecting physically distant locations). The best example of this is the internet itself – that is, thousands of local networks (LAN / WLAN) connected.

Differences between IP and MAC address

IP (internet protocol)

We use IP for communication between different networks (to address and transport data from one network to another). It performs the role of routing, i.e., searches for the fastest route to pass a data packet. An IP address is a logical address – this means that it is allocated depending on which network the device has been connected to. If a device is in two networks, it will have two IP addresses.

MAC address (Media Access Control)

MAC is a physical address with a unique identifier burned out on the network card. It identifies specific devices and is assigned by the manufacturer. MAC addresses are used for communication within one network, e.g., in a home network, if you want to connect a computer to a printer or other devices, it will use MAC addresses to do that.

OSI model

The OSI model has never been directly implemented as it's mostly a reference architecture on how data should flow from one application to another through a network. TCP/IP is used, and these days it's the most popular. After the OSI model, I will say more about TCP/IP. But it's good to start with the OSI because it's easier to understand some of the concepts.

The OSI model consists of 7 layers divided into two groups:

- Host layers (happening on the computer side. Responsible for accurate data delivery between devices)
- Media layers (happening on the network side. Responsible for making sure that the data has arrived at its destination)

Application layer

In this layer, the user directly interacts with applications. Here is decided which interfaces are used to interact with the network through the corresponding protocols in this layer.

Examples of such applications are chrome or Gmail:

- Chrome uses the HTTP / HTTPS protocol
- Gmail uses email protocols like SMTP, IMAP.

The applications themselves are not in the application layer – in this layer, there are only the protocols or services that the applications use.

Presentation layer

The task of this layer is proper data representation, compression/decompression, and encryption/decryption. This ensures that the data sent from the X system application layer can be read by the Y system application layer.

Session layer

This layer is responsible for creating, managing, and then closing sessions between two applications that want to communicate with each other.

Transport layer

The task of this layer is to make sure that the data has arrived safely from the sender to the recipient. When it sends data, it breaks it into segments. When it accepts data, it puts it back into a stream of data.

Network layer

Provides addressing and routing services. It defines which routes connect individual computers and decides how much information to send using one connection or another. Data transferred through this layer are called packets.

Places two addresses in the packet sent:

- Source address
- Destination address

This layer is based on IP (internet protocol).

Data-link layer

This layer deals with packing data into frames and sending them to the physical layer. It also oversees the quality of the information provided by the physical layer. It recognizes errors related to losing packages and damaging frames and deals with their repair.

Physical layer

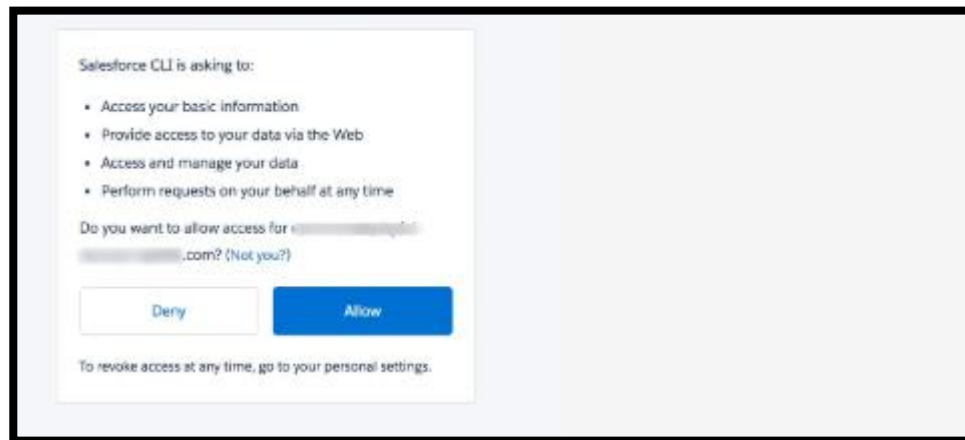
This is the physical aspect of the network. This applies to cables, network cards, WIFI, etc. It is only used to send logical zeros and ones (bits). It determines how fast the data flows. When this layer receives frames from the data link layer, it changes them to a bit stream.

DAILY ASSESSMENT

Date:	21/07/2020	Name:	Davis S. Patel
Course:	Salesforce Developer	USN:	4AL16EC045
Topic:	Salesforce DX	Semester & Section:	8 th - A
GitHub Repository:	Davis		

AFTERNOON SESSION DETAILS

Image of Session



```
~/Documents/workspace/my_sfdx_project/dreamhouse-sfdx (my_branch)
> sfdx force --help
Usage: sfdx force: [-v] [--json] [--loglevel <string>] [Flags]

Flags:
  -v, --version      display the Salesforce API version
  --json             format output as json
  --loglevel LOGLEVEL logging level for this command invocation (error,trace,debug,info,warn,fatal)

Usage: sfdx force:COMMAND

Help topics, type sfdx help TOPIC for more details:

force:alias          manage username aliases
force:apex            work with Apex code
force:auth            authorize an org for use with the Salesforce CLI
force:config          configure the Salesforce CLI
force:data            manipulate records in your org
force:doc             display help for force commands
force:lightning       create and test Lightning component bundles
force:limits          view your org's limits
force:mdapi           retrieve and deploy metadata using Metadata API
force:org             manage your orgs
force:package         develop and install packages
force:packagel        develop first-generation managed and unmanaged packages
force:project         set up a Salesforce DX project
force:schema          view standard and custom objects
force:source          sync your project with your orgs
force:user            perform user-related admin tasks
force:visualforce     create and edit Visualforce files
```

REPORT –

The Salesforce Developer Experience (DX) is a set of tools that streamlines the entire development life cycle. It improves team development and collaboration, facilitates automated testing and continuous integration, and makes the release cycle more efficient and agile.

This Salesforce DX quick start begins with source code living in your version control system (VCS). It doesn't matter which VCS you use, only that you use one. In this quick start, we assume you're using Git and GitHub, as this is where we've stored the sample application, called the DreamHouse app.

Much of the setup you do for Salesforce DX enables you to use a new type of org called a scratch org. A scratch org is a dedicated, configurable, and short-term Salesforce environment that you can quickly spin up when starting a new project, a new feature branch, or a feature test.

A **Developer Hub** (Dev Hub) is the main Salesforce org that you and your team use to create and manage your scratch orgs.

Enable Dev Hub in Your Trailhead Playground

While you can enable Dev Hub in any paid org, it's always best to practice somewhere other than production. Instead, go ahead and enable Dev Hub in a Developer Edition org or Trailhead Playground to use with this project.

Get Your Trailhead Username and Password

You need your Dev Hub login credentials for this project. If you don't already know the username and password for the Developer Edition or Trailhead Playground org where you enabled Dev Hub.

Download a Project from GitHub

If you're a Windows user, first install [git for Windows](#) tooling. This tool helps you run git from the command line.

1. Open a new Terminal (Mac) or Command Prompt (Windows) window and use the following command to create a unique directory to do your work in:
2. `mkdir my_sfdx_project`

```
cd my_sfdx_project
```

Copy

3. Next, use this command to clone the app repository:

```
git clone https://github.com/trailheadapps/dreamhouse-lwc.git
```

Copy

4. Next, open the directory:

```
cd dreamhouse-sfdx
```

Copy

```
(base) edegroot-ltm:~ edegroot$ git clone https://github.com/trailheadapps/dreamhouse-lwc.git
Cloning into 'dreamhouse-lwc'...
remote: Enumerating objects: 98, done.
remote: Counting objects: 100% (98/98), done.
remote: Compressing objects: 100% (78/78), done.
remote: Total 1248 (delta 39), reused 35 (delta 11), pack-reused 1150
Receiving objects: 100% (1248/1248), 956.65 KiB | 2.93 MiB/s, done.
Resolving deltas: 100% (769/769), done.
(base) edegroot-ltm:~ edegroot$ cd dreamhouse-lwc
(base) edegroot-ltm:dreamhouse-lwc edegroot$
```

Cloning the repository pulls all the source code into your local file system. But before you start editing, you first create your own branch. This is a best practice as defined by [GitHub Flow](#), because it helps ensure your Master branch is a clean and production-ready version of your code.