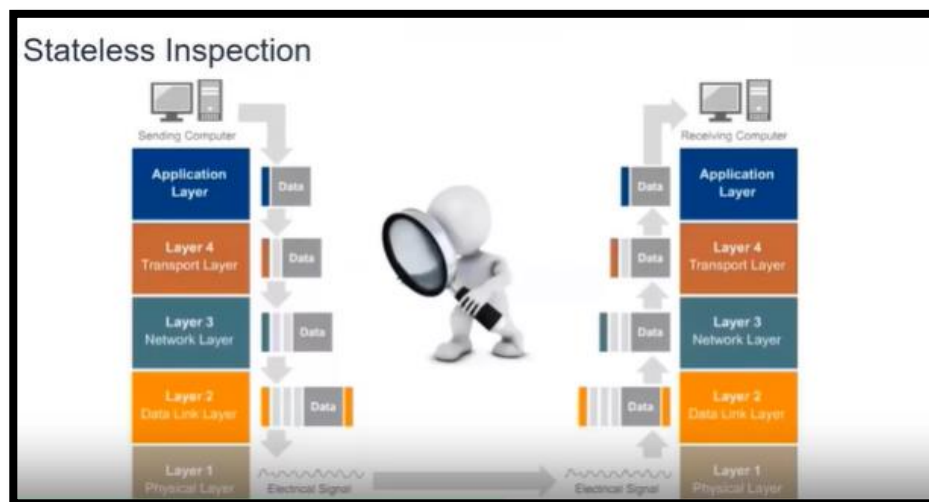
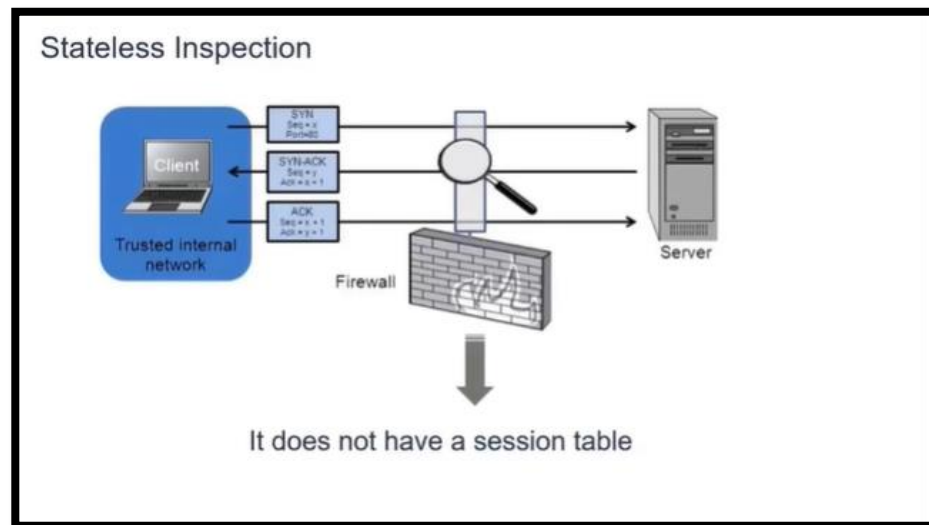


DAILY ASSESSMENT

Date:	20/07/2020	Name:	Davis S. Patel
Course:	Network Security & Database Vulnerabilities	USN:	4AL16EC045
Topic:	Week 1	Semester & Section:	8 th - A
GitHub Repository:	Davis		

FORENOON SESSION DETAILS

Image of session



REPORT –

Network Interface layer

The lowest level of the TCP/IP architecture is the Network Interface layer. It corresponds to the OSI Physical and Data Link layers. You can use many different TCP/IP protocols at the Network Interface layer, including Ethernet and Token Ring for LANs and protocols such as X.25, Frame Relay, and ATM for wide area networks (WANs).

The Network Interface layer is assumed to be unreliable.

Network layer

The Network layer is where data is addressed, packaged, and routed among networks. Several important Internet protocols operate at the Network layer:

- **Internet Protocol (IP):** A routable protocol that uses IP addresses to deliver packets to network devices. IP is an intentionally unreliable protocol, so it doesn't guarantee delivery of information.
- **Address Resolution Protocol (ARP):** Resolves IP addresses to hardware MAC addresses, which uniquely identify hardware devices.
- **Internet Control Message Protocol (ICMP):** Sends and receives diagnostic messages. ICMP is the basis of the ubiquitous ping command.
- **Internet Group Management Protocol (IGMP):** Used to multicast messages to multiple IP addresses at once.

Transport layer

The Transport layer is where sessions are established and data packets are exchanged between hosts. Two core protocols are found at this layer:

- **Transmission Control Protocol (TCP):** Provides reliable connection-oriented transmission between two hosts. TCP establishes a session between hosts, and then ensures delivery of packets between the hosts.
- **User Datagram Protocol (UDP):** Provides connectionless, unreliable, one-to-one or one-to-many delivery.

Application layer

The Application layer of the TCP/IP model corresponds to the Session, Presentation, and Application layers of the OSI Reference Model. A few of the most popular Application layer protocols are

- **HyperText Transfer Protocol (HTTP):** The core protocol of the World Wide Web

- **File Transfer Protocol (FTP):** A protocol that enables a client to send and receive complete files from a server
- **Telnet:** The protocol that lets you connect to another computer on the Internet in a terminal emulation mode
- **Simple Mail Transfer Protocol (SMTP):** One of several key protocols that are used to provide e-mail services
- **Domain Name System (DNS):** The protocol that allows you to refer to other host computers by using names rather than numbers

TCP/IP provides the necessary framework for two points connected through a network to exchange information. In the Protocol Stack, the set of protocols used in a communications network, TCP/IP plays a particularly important role in two specific layers:

In the Transport Layer, where correct delivery of data is ensured and in the Network Layer, where the correct recipient is located

TCP is activated with every network request/response. For example, in an HTTP request, TCP takes over as soon as the browser knows where the request should be routed, i.e. after DNS resolution has been completed. Based on the socket provided (combination of IP address and server port), the request will reach the target computer and application through the network. The necessary communication channel will open up and data will be broken down to appropriately sized packets. Then, they will be sent over to the server. While the server handles the request and prepares the response accordingly, TCP makes sure that this particular connection channel remains open until the response reaches the source of the request successfully.

While moving data around, TCP/IP protocols annotate segments with extra information (headers) in order to be able to perform all the above tasks successfully. Headers include information regarding the segment sequence number, a number (checksum) to allow confirmation of data validity and information about sender and recipient.

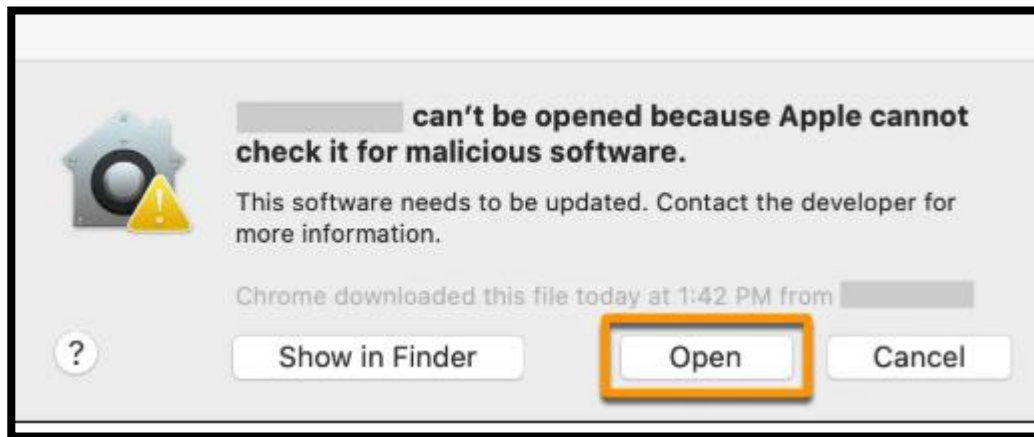
This added information allows data to be segmented and transmitted as efficiently as possible, making sure it is correctly restructured at the destination, without worrying about structure during transportation.

DAILY ASSESSMENT

Date:	20/07/2020	Name:	Davis S. Patel
Course:	Salesforce Developer	USN:	4AL16EC045
Topic:	Lightning Web Components	Semester & Section:	8 th - A
GitHub Repository:	Davis		

AFTERNOON SESSION DETAILS

Image of Session



```
1  Salesforce CLI
2  VERSION
3    sfdx-cli/7.36.0-sfdx-cli/7.36.0-b1c4e9c7c8 darwin-x64 node-v10.15.3
4  USAGE
5    $ sfdx [COMMAND]
6  COMMANDS
7    commands  list all the commands
8    force     tools for the Salesforce developer
9
```

REPORT –

Lightning Web Components is a new programming model for building Lightning components. It uses web standards breakthroughs, can coexist and interoperate with the Aura programming model, and delivers unparalleled performance. To create and develop Lightning Web Components and use their powerful features and performance benefits, you need to set up Salesforce DX. For this Quick Start, you also use Visual Studio Code, which is the recommended code editor for developing on the Salesforce platform. Once your developer environment is ready, you learn how to write a simple Lightning web component and add it to a page in Lightning Experience.

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.

Set Up Your Trailhead Playground

1. Create a new Trailhead Playground.
2. To create a new Trailhead Playground, click the dropdown at the end of this step and select Create a Trailhead Playground.
3. The Create a Trailhead Playground dropdown in a project's verify step challenge.
4. Once you have a Trailhead Playground, click Launch.
5. Reset your password. You need both your username and password to authenticate your Trailhead Playground with the Salesforce CLI in a later step.
6. If you see a tab in your org labeled Get Your Login Credentials, great! Follow the steps below. If not, click App Launcher to launch the App Launcher, then click Playground Starter and follow the steps. If you don't see the Playground Starter app, check out Find the Username and Password for Your Trailhead Playground on Trailhead Help.
7. Click the Get Your Login Credentials tab and take note of your username.

8. Click Reset My Password. This sends an email to the address associated with your username.
9. Click the link in the email.
10. Enter a new password, confirm it, and click Change Password.

Set Up Visual Studio Code

Install Salesforce Extensions for Visual Studio Code

Visual Studio Code is the go-to code editor for Salesforce developers. It's free, open-source, and available for Windows, Linux, and macOS. This editor has easy-to-install extensions for syntax highlighting, code completion, and more.

We install Visual Studio Code and the recommended Salesforce Extension Pack.

1. Download and install the latest version of Visual Studio Code for your operating system. If you already have Visual Studio Code installed, there's no need to reinstall it.
2. Launch Visual Studio Code.
3. On the left toolbar, click the Extensions icon Visual Studio Code's Extension icon. .
4. Search for Salesforce Extension Pack and click Install. If you already have it installed, then you just need to click on the Reload button.

Create a Lightning Web Component

1. In Visual Studio Code, open the Command Palette by pressing Ctrl+Shift+P (Windows) or Cmd+Shift+P (macOS).
2. Type SFDX.
3. Select SFDX: Create Lightning Web Component. Don't use SFDX: Create Lightning Component. (This creates an Aura component.)
4. Enter helloWorld for the name of the new component.
5. Press Enter to accept the default force-app/main/default/lwc.
6. Press Enter.
7. View the newly created files in Visual Studio Code.