🕐

Day2

PEH

`Networking Refresher`

`IPv4`

IP Addresses 32 BIT 4Byte address consist of 8 octets

`2^32 = 4,294,967,296`

`IPv6`

IP Address 128Bits

2^128

We communicate over Layer3

`We are using something called NAT to handle the issue of space out of range`

With NAT what we are doing is we are assigned these private IP address spaces.

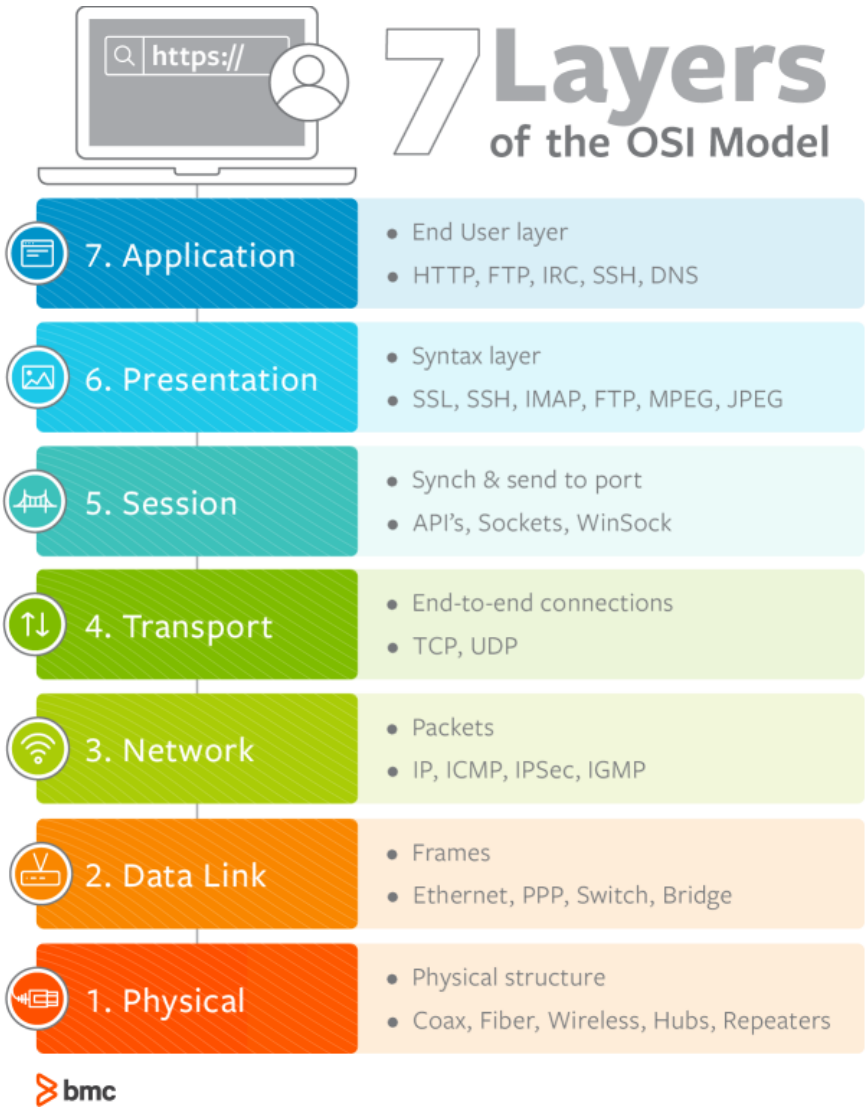
Just for Reference

`128 64 32 16 8 4 2 1`

`0 0 0 0 0 1 1 1` = 7 because 1+2+4

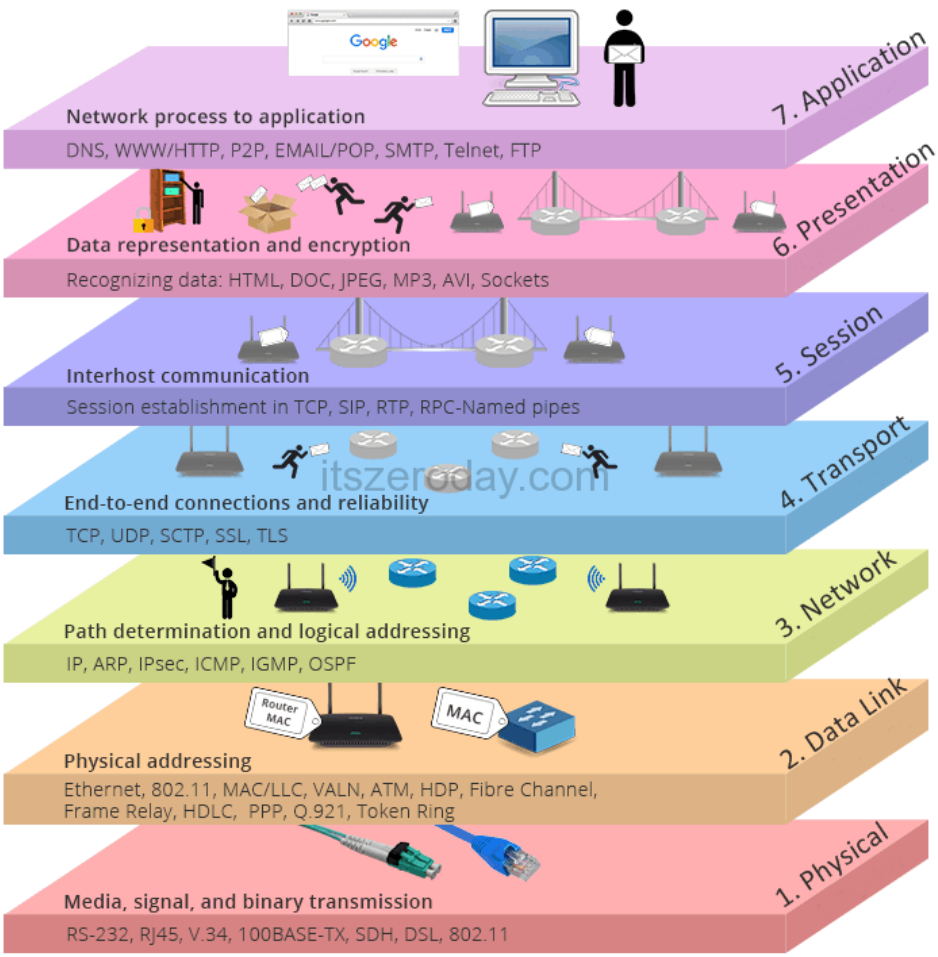
I passed the PEH Course for now and i'll move back to it tonight and will update this day progress too.

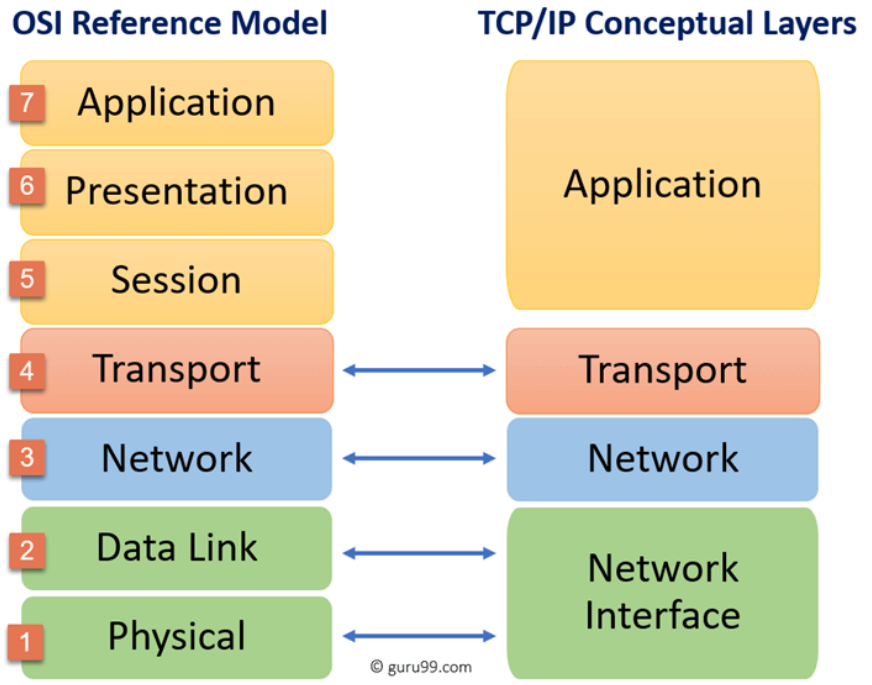
OSI MODEL and TCP IP MODEL



TCP/IP MODEL
Application Layer
Transport Layer
Internet Layer
Network Access Layer

OSI MODEL
Application Layer
Presentation Layer
Session Layer
Transport Layer
Network Layer
Data Link Layer
Physical Layer


Google
7. Application
Network process to application
DNS, WWW/HTTP, P2P, EMAIL/POP, SMTP, Telnet, FTP
6. Presentation
Data representation and encryption
Recognizing data: HTML, DOC, JPEG, MP3, AVI, Sockets
5. Session
Interhost communication
Session establishment in TCP, SIP, RTP, RPC-Named pipes
4. Transport
End-to-end connections and reliability
TCP, UDP, SCTP, SSL, TLS
3. Network
Path determination and logical addressing
IP, ARP, IPsec, ICMP, IGMP, OSPF
2. Data Link
Router MAC
MAC
Physical addressing
Ethernet, 802.11, MAC/LLC, VALN, ATM, HDP, Fibre Channel,
Frame Relay, HDLC, PPP, Q.921, Token Ring
1. Physical
Media, signal, and binary transmission
RS-232, RJ45, V.34, 100BASE-TX, SDH, DSL, 802.11


OSI Reference Model
TCP/IP Conceptual Layers
7 Application
6 Presentation
5 Session
4 Transport
3 Network
2 Data Link
1 Physical
Application
Transport
Network
Network Interface
© guru99.com

Learned about Git & Github

Commands

`git config –global user.name “Ishfaq Fariq”`

`git config –global user. email “Ishfaqfariq@protonmail.com”`

`git config –global – edit`

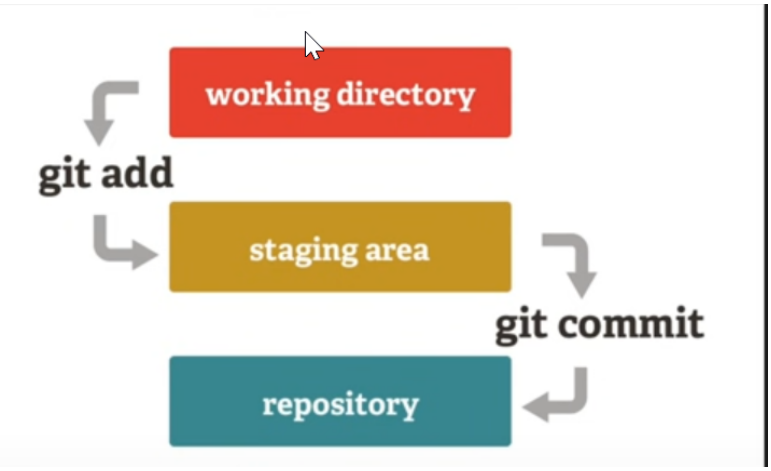
`git init “make your xyz folder or project a repo/Initialize it”`

`before making any change or pushing your code always do git status`

`git status “this will inform about unchanged/untracked files”`

`git add {file name} “this will add the file to staging area”`

`git commit -m {Message of commit}`



`git log “This will tell us about the commits performed”`

`git add . “This will include all the new files in staging”`

`git checkout {hashcode/branchname} “This will take you back to the added branch/commit also for moving to different branches “`

`git branch {name} “to make new branch”`

`git checkout -b {name} “this will make new branch will checkout/move to that branch”`

`git merge {name of the branch that you want to merger with current head} “this will merge the entered branch with the current branch where you are currently residing “`

`.gitignore and add the secret stuff or files in it to make it secure, I mean to not add in your repo`

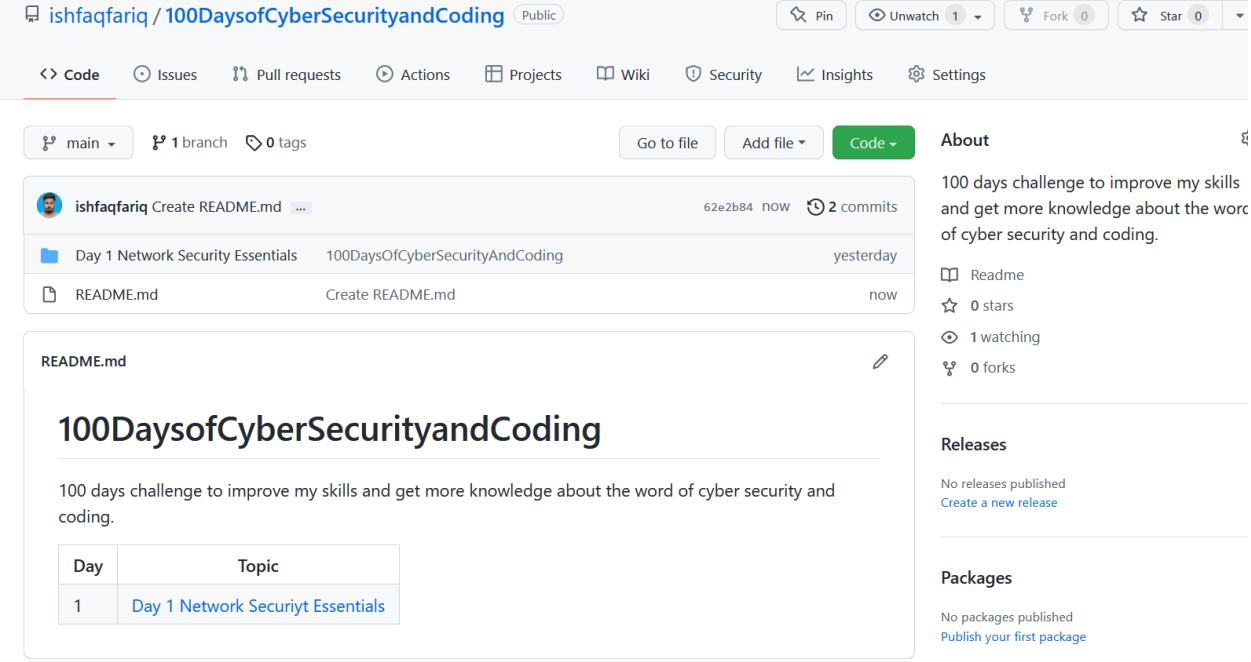
`git remote -v “to see the origin for fetching and pushing”`

`git remote add {can be anything but prefer origin/origin} {enter your remote origin address, where you want to push code }`

`git branch -M {name of branch/main}`

`git push -u origin {name of branch/main} "this will add the file/code in origin in the specified branch”`

`git clone https://github.com/USERNAME/REPOSITORY.git “This command copies all the files in a repository to your computer, and begins tracking them in git. You do this by typing in”`



Pushed the Day 1 PDF in Github Repo and Made the Readme file and will gonna add all the links to my notes(notion) in the table in Readme.

Tested a WordPress Plugin for flaw and vulnerabilities and checked it against Secure coding standard.

The tools I used for doing Dynamic Testing and Code Audit is PHP Code Sniffer and WP Bullet. But unfortunately the WpBullet Didn’t work in this case.

for phpcs cbf need to install php:

download php and extract, rename it till version put in c:// and add that dir to environment.

https://www.youtube.com/watch?v=QMWb_Wn2g5k

Installation

The easiest way to get started with PHP_CodeSniffer is to download the Phar files for each of the commands:

```
# Download using curl
curl -OL https://squizlabs.github.io/PHP_CodeSniffer/phpcs.phar
curl -OL https://squizlabs.github.io/PHP_CodeSniffer/phpcbf.phar

# Or download using wget
wget https://squizlabs.github.io/PHP_CodeSniffer/phpcs.phar
wget https://squizlabs.github.io/PHP_CodeSniffer/phpcbf.phar

# Then test the downloaded PHARs
php phpcs.phar -h
php phpcbf.phar -h
```

php cs tells about flaws in code in command line, while cbf try to solve this issues as much as possible but the chances of false positive is high

Simply clone the repository, install requirements and run the script

- `$ git clone https://github.com/webarx-security/wpbullet wpbullet`
- `$ cd wpbullet`
- `$ pip install -r requirements.txt`
- `$ python wpbullet.py`

Usage

Available options:

```
--path (required) System path or download URL
Examples:
--path="/path/to/plugin"
--path="https://wordpress.org/plugins/example-plugin"
--path="https://downloads.wordpress.org/plugin/example-plugin.1.5.zip"

--enabled (optional) Check only for given modules, ex. --enabled="SQLInjection,CrossSiteScripting"
--disabled (optional) Don't check for given modules, ex. --disabled="SQLInjection,CrossSiteScripting"
--cleanup (optional) Automatically remove content of .temp folder after scanning remotely downloaded plugin (
--report (optional) Saves result inside reports/ directory in JSON format (boolean)

$ python wpbullet.py --path="/var/www/wp-content/plugins/plugin-name"
```

==========