# Networking Intro

Cyber Dawgs Fall 2018

Zack Orndorff

# Why learn about networking?

Whether defending or pentesting a network, it’s important to be familiar with the way different components communicate

That communication… well… is networking

Agenda:

- Look at the layers of the network stack. Define them.
- Then looking at how they work
- Along the way (and at the end), learn at tools that help you inspect traffic

# OSI Model - Conceptual Model for Thinking

7. Application - HTTP, SMTP, DNS, POP3, etc.
6. Presentation
5. Session
4. Transport - TCP, UDP
3. Network - IP
2. Data Link
1. Physical

# 7 Application

These are the things you normally see, for example:

- HTTP
- SMTP
- IMAP
- POP3
- DNS
- FTP
- SSH

```
:)-23:13-zack@sperfari:89988:0:~
$ nc -C umbc.edu 80
GET / HTTP/1.1
Host: umbc.edu

HTTP/1.1 301 Moved Permanently
Date: Mon, 24 Sep 2018 03:14:14 GMT
Server: Apache/2.4.6 (CentOS) OpenSSL/1.0.2k-fips mod_fcgid/2.3.9 PHP/5.4.16 mod_perl/2.0.10 Perl/v5.16.3
Location: http://www.umbc.edu//
Content-Length: 229
Connection: close
Content-Type: text/html; charset=iso-8859-1

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>301 Moved Permanently</title>
</head><body>
<h1>Moved Permanently</h1>
<p>The document has moved <a href="http://www.umbc.edu//">here</a>.</p>
</body></html>
^C
130 :(-23:14-zack@sperfari:89989:0:~
$ nc -C umbc.edu 80
GET / HTTP/1.1
Host: www.umbc.edu

HTTP/1.1 200 OK
Date: Mon, 24 Sep 2018 03:14:26 GMT
Server: Apache/2.4.6 (CentOS) OpenSSL/1.0.2k-fips mod_fcgid/2.3.9 PHP/5.4.16 mod_perl/2.0.10 Perl/v5.16.3
X-Powered-By: PHP/5.4.16
Connection: close
Transfer-Encoding: chunked
Content-Type: text/html; charset=UTF-8

3959
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <title>UMBC: An Honors University In Maryland</title>
    <!-- Google Tag Manager -->
<script>(function(w,d,s,l,i){w[l]=w[l]||[];w[l].push({'gtm.start':
new Date().getTime(),event:'gtm.js'});var f=d.getElementsByTagName(s)[0],
j=d.createElement(s),dl=l!='dataLayer'?'&l='+l:'';j.async=true;j.src=
'https://www.googletagmanager.com/gtm.js?id='+i+dl;f.parentNode.insertBefore(j,f);
```

# Physical/Data Link

Interesting terms you'll hear at this layer are:

- MAC addresses
  - Look like this: 01:23:45:67:89:AB
  - First 3 octets are the OUI (organizationally unique identifier), aka the manufacturer
- Switches operate at this layer
- A unit of data in Ethernet is called a *frame*.
- VLANs are separated at this layer
- You use *Wireshark* to capture at this layer and examine traffic

# Network Layer: IP!

Interesting things at this layer:

- Units of data are called *packets*.
- Routers operate at this layer
  - They route traffic between subnets.
  - We'll discuss subnets at a later date. For now, just know that to go between them, a router is needed.
- IPv4 vs IPv6
  - IPv4 addresses look like this: 127.0.0.1
    - We're running out of these
  - IPv6 addresses look ~~awful~~ like this: 2001:0db8:85a3:0000:0000:8a2e:0370:7334
    - There is an obscene number of these, we're never running out
  - Firewalls sometimes operate differently on IPv4 and IPv6, so be careful.

# IP addresses & routing

Typical LAN settings:

IP address: 192.168.1.103
Netmask: 255.255.255.0
Gateway: 192.168.1.1

--

Address is 192.168.1.103/24
Gateway is 192.168.1.1

CIDR notation

When packets must leave the subnet, the router consults its routing table

Table filled in by BGP, OSPF, RIP, IGRP

# ARP (Address Resolution Protocol)

ARP converts IP addresses to MAC addresses

Susceptible to ARP spoofing

# Transport: TCPs and UDPs and ICMPs, oh my!

IP is designed to get packets from host A to host B. That's all it cares about -- packets.

What if you want to send a stream of data? Or distinguish what application packets are being sent to?

TCP - reliable transmission of data
UDP - fast but not so reliable
ICMP - interesting things are Time Exceeded, Echo Request/Reply, Destination Unreachable

Each of these has a IP *protocol number* that identifies it in an IP packet

# Fun things common to TCP and UDP

Ports are a thing! The theory is that you have a bunch of little mailboxen that packets can fly into

Really a field in the TCP or UDP header processed by the receiving computer

Range: 1-65535 (0 is special)

All packets have source and destination port

- Well-known ports: 1-1023
- Registered Ports: 1024-49151
- Ephemeral Ports: 49152-65535

# TCP

TCP abstracts over the “packet” feature of IP to provide a reliable stream of data

TCP connections begin with the 3-way handshake

Client sends SYN packet
Server responds with SYN-ACK packet
Client sends ACK packet

Now the connection is established

If packets arrive out-of-order or get dropped, TCP reorders or resends them such that the original stream is reconstructed.

# UDP

Source port

Destination port

Length

Not much else, definitely no connection logic

Used by DNS, DHCP, many streaming applications, etc.

# 5/6 Session / Presentation

I have yet to hear of a good use for these layers

# Application layer

Lower layers exist to abstract annoying details from the application

Thanks to TCP, applications can just say “send this data to host B” and it works

We’re going to focus on HTTP, SMTP, and DNS today

Protocols you should have heard of:

- FTP - File Transfer Protocol
- Telnet
- SSH - Secure Shell
- DNS - Domain Name Service
- DHCP - Dynamic Host Configuration Protocol
- HTTP - HyperText Transfer Protocol
- SMTP - Simple Mail Transfer Protocol
- IMAP - Internet Message Access Protocol
- POP3 - Post Office Protocol 3
- LDAP - Lightweight Directory Access Protocol

# Fun things about Application layer protocols

They can be loosely categorized as *binary* or *text* protocols

HTTP, SMTP, IMAP, and many more are text

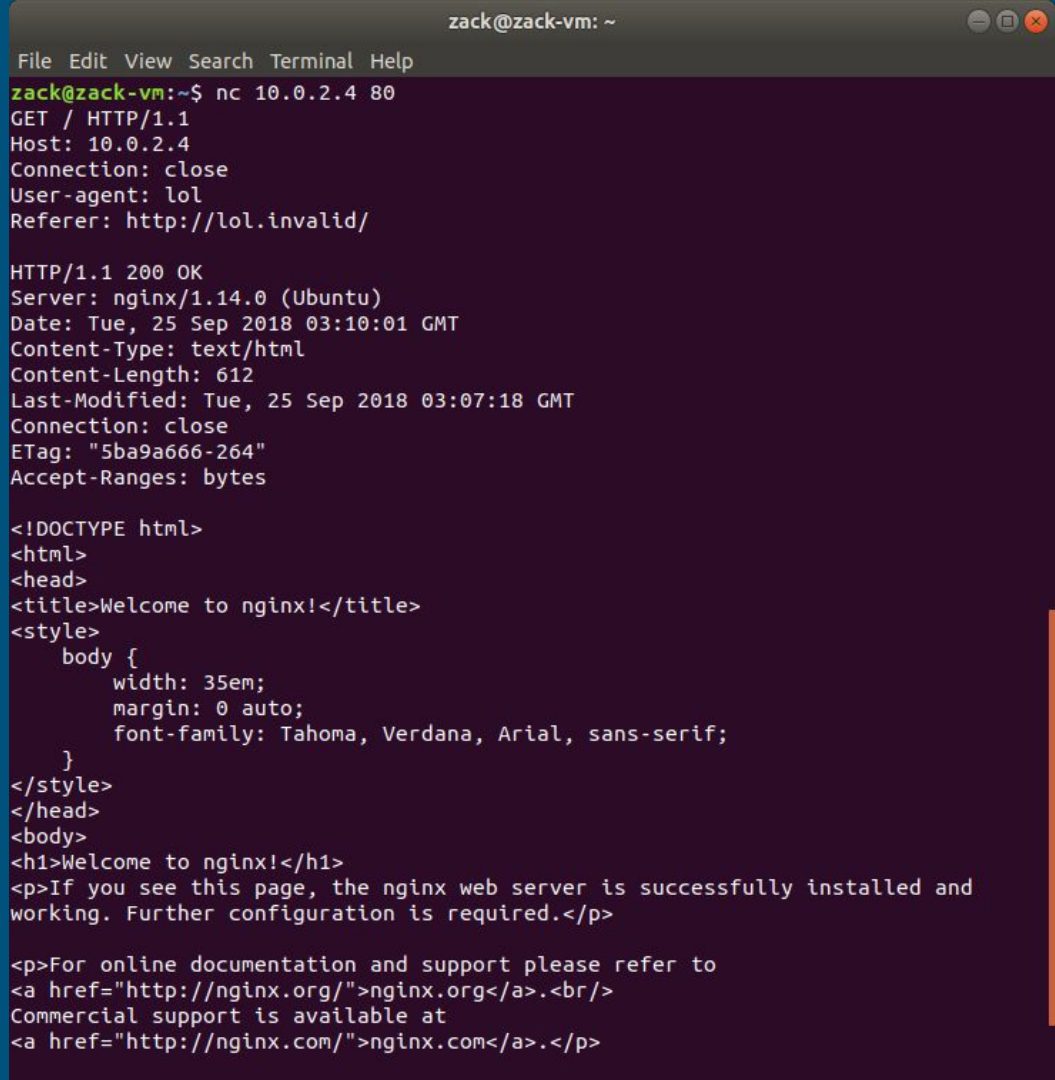
DNS, LDAP, SSH, and many more are binary

You can interact with text-based protocols by hand!

# HTTP

Transfers web pages

Usually runs on port 80



# SMTP

Sends email

Standard port is 25, but if you're sending mail as a user, it goes over the "submission" port, 587

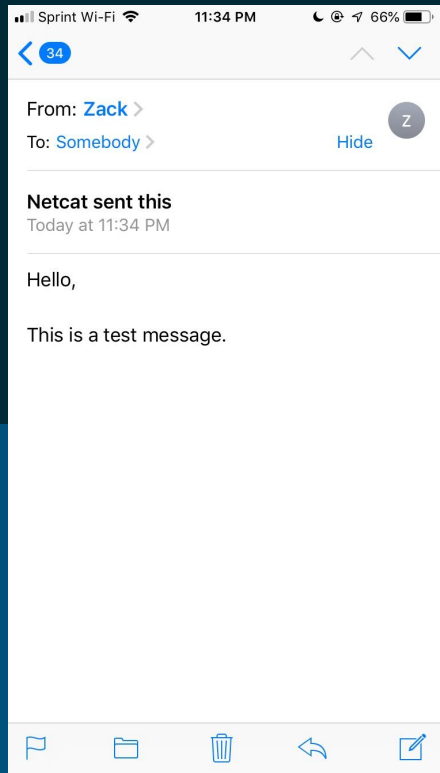
```
130 :(-23:31-zack@sperfari:90004:0:~
$ nc www.zackorndorff.com 25
220 www.zackorndorff.com ESMTP
HELO test.z10f.com
250 www.zackorndorff.com
MAIL FROM: <zo1@umbc.edu>
250 2.1.0 Ok
RCPT TO: <hi@sqordfish.com>
250 2.1.5 Ok
DATA
354 End data with <CR><LF>.<CR><LF>
From: "Zack" <zo1@umbc.edu>
To: "Somebody" <hi@sqordfish.com>
Subject: Netcat sent this
Date: now

Hello,

This is a test message.

.

250 2.0.0 Ok: queued as 1BE31A1BF
```



# DNS

Maps domain names to IP addresses

Also stores other “resource reco
(RRs)

Hierarchical

Interesting types of records: A, A
(quad-A), MX, TXT, NS, SOA, SRV
CNAME

```
$ dig umbc.edu A

; <<>> DiG 9.10.3-P4-Debian <<>> umbc.edu A
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 47592
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 1, AUTHORITY: 13, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;umbc.edu.                      IN      A

;; ANSWER SECTION:
umbc.edu.               59      IN      A       130.85.12.160
```

```
:)-23:42-zack@sperfari:90016:0:~
$ dig umbc.edu TXT

; <<>> DiG 9.10.3-P4-Debian <<>> umbc.edu TXT
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 2736
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 5, AUTHORITY: 13, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;umbc.edu.                      IN      TXT

;; ANSWER SECTION:
umbc.edu.               3599    IN      TXT     "docusign=bce67f31-88f6-4139-a4af-9036
ef6b9aa5"
umbc.edu.               3599    IN      TXT     "docusign=139f743c-ee8c-4242-bc29-0183
80e725d1"
umbc.edu.               3599    IN      TXT     "adobe-idp-site-verification=9eb97446-
9d37-4b21-bbc8-c8e6d5578992"
umbc.edu.               3599    IN      TXT     "v=spf1 ip4:63.146.179.205 ip4:130.85.
0.0/16 ip4:66.151.109.15 ip4:66.151.109.16 ip4:66.151.109.77 ip4:66.151.109.78 ip4:207
.75.116.229 ip4:34.194.230.233 ip4:34.230.107.215 " "mx include:_spf.google.com includ
e:mh.blackboard.com ?all"
umbc.edu.               3599    IN      TXT     "MS=ms37655703"
```

# Tools!

- ping - Sends ICMP Echo Request packets
  - ping host.example
- netcat (nc) - Lets you send arbitrary data over TCP or UDP
  - nc host.example 80
- tcpdump - captures and dumps traffic on a network interface
- tshark - more featureful tcpdump
- Wireshark - GUI tool to create and analyze packet captures
- curl/wget/httpie - let you make lots of HTTP requests
  - curl http://google.com
- nmap - network mapper (more on this later)
  - nmap -A -T5 host.you-own.example
- scapy - Python library to do all kinds of things with packets

# Nmap

Maps networks.

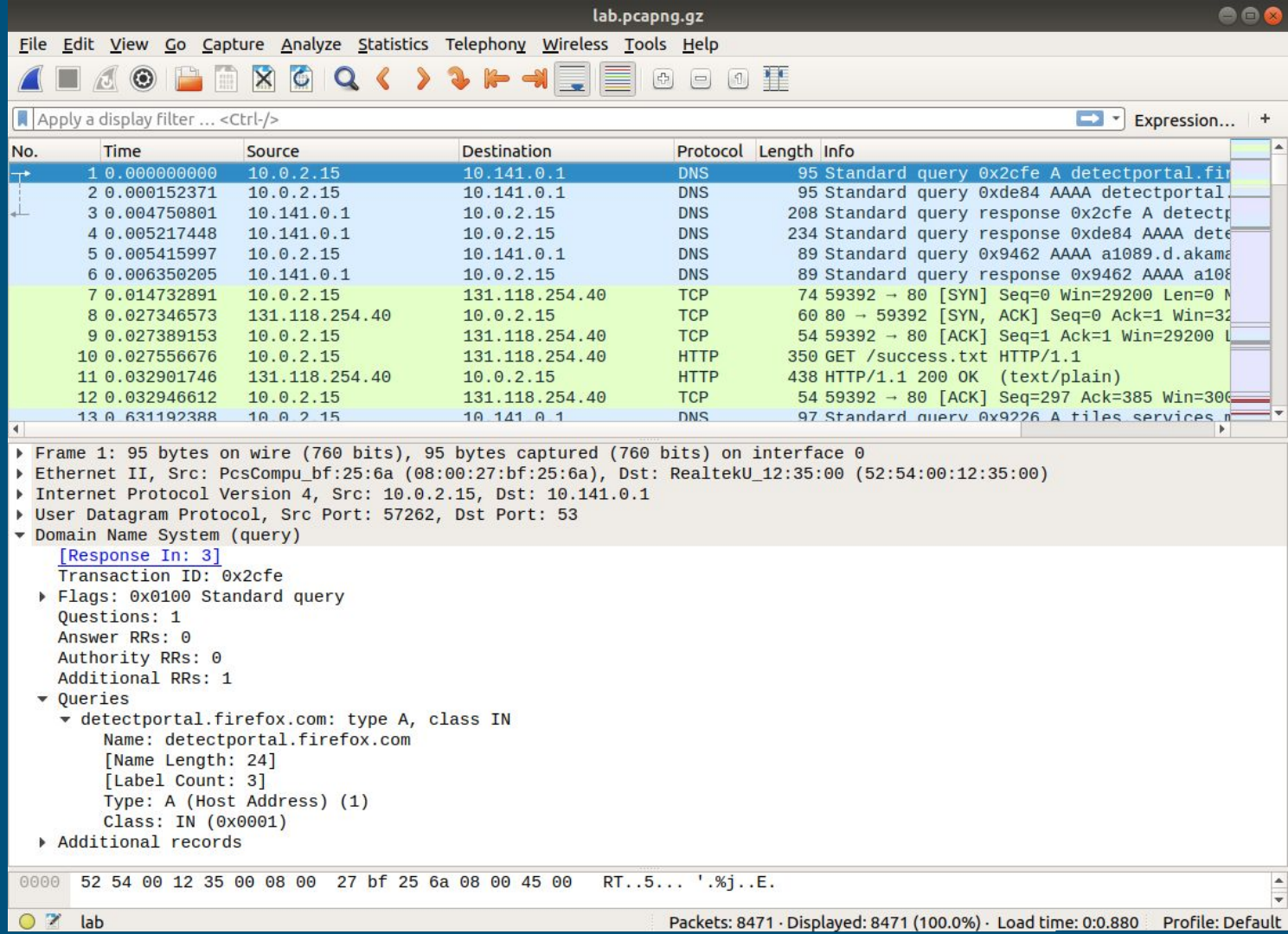
Lets you determine what's running on a given host, or on a given network.

```
:)-00:11-zack@sperfari:90023:0:~
$ nmap -A -T5 smb.int.umbccd.net

Starting Nmap 7.40 ( https://nmap.org ) at 2018-09-25 00:11 EDT
Nmap scan report for smb.int.umbccd.net (172.21.0.2)
Host is up (0.0076s latency).
Not shown: 996 closed ports
PORT     STATE SERVICE VERSION
22/tcp   open  ssh     OpenSSH 7.4 (protocol 2.0)
| ssh-hostkey:
|   2048 5b:81:46:f0:27:78:b8:13:00:47:98:12:f0:e1:e6:de (RSA)
|_  256 64:28:33:05:f1:73:5e:4b:d1:c1:da:c6:d5:22:a4:82 (ECDSA)
53/tcp   open  domain  ISC BIND 9.9.4
| dns-nsid:
|_  bind.version: 9.9.4-RedHat-9.9.4-51.el7_4.1
111/tcp  open  rpcbind 2-4 (RPC #100000)
| rpcinfo:
|   program version   port/proto  service
|   100000  2,3,4        111/tcp  rpcbind
|   100000  2,3,4        111/udp  rpcbind
|   100003  3,4         2049/tcp  nfs
|   100003  3,4         2049/udp  nfs
|   100005  1,2,3      20048/tcp  mountd
|   100005  1,2,3      20048/udp  mountd
|   100021  1,3,4      38491/tcp  nlockmgr
|   100021  1,3,4      44361/udp  nlockmgr
|   100024  1          34559/tcp  status
|   100024  1          55752/udp  status
|   100227  3           2049/tcp  nfs_acl
|_  100227  3           2049/udp  nfs_acl
2049/tcp open  nfs_acl 3 (RPC #100227)
Service Info: OS: Red Hat Enterprise Linux 7; CPE: cpe:/o:redhat:enterprise_linux:7

Service detection performed. Please report any incorrect results at https://nmap.org
ubmit/ .
Nmap done: 1 IP address (1 host up) scanned in 14.86 seconds
```

# Wireshark


lab.pcapng.gz
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
Expression...
No. Time Source Destination Protocol Length Info
1 0.000000000 10.0.2.15 10.141.0.1 DNS 95 Standard query 0x2cfe A detectportal.fir
2 0.000152371 10.0.2.15 10.141.0.1 DNS 95 Standard query 0xde84 AAAA detectportal.
3 0.004750801 10.141.0.1 10.0.2.15 DNS 208 Standard query response 0x2cfe A detectp
4 0.005217448 10.141.0.1 10.0.2.15 DNS 234 Standard query response 0xde84 AAAA dete
5 0.005415997 10.0.2.15 10.141.0.1 DNS 89 Standard query 0x9462 AAAA a1089.d.akama
6 0.006350205 10.141.0.1 10.0.2.15 DNS 89 Standard query response 0x9462 AAAA a108
7 0.014732891 10.0.2.15 131.118.254.40 TCP 74 59392 → 80 [SYN] Seq=0 Win=29200 Len=0 M
8 0.027346573 131.118.254.40 10.0.2.15 TCP 60 80 → 59392 [SYN, ACK] Seq=0 Ack=1 Win=32
9 0.027389153 10.0.2.15 131.118.254.40 TCP 54 59392 → 80 [ACK] Seq=1 Ack=1 Win=29200 L
10 0.027556676 10.0.2.15 131.118.254.40 HTTP 350 GET /success.txt HTTP/1.1
11 0.032901746 131.118.254.40 10.0.2.15 HTTP 438 HTTP/1.1 200 OK (text/plain)
12 0.032946612 10.0.2.15 131.118.254.40 TCP 54 59392 → 80 [ACK] Seq=297 Ack=385 Win=306
13 0.631192388 10.0.2.15 10.141.0.1 DNS 97 Standard query 0x9226 A tiles.services.m
Frame 1: 95 bytes on wire (760 bits), 95 bytes captured (760 bits) on interface 0
Ethernet II, Src: PcsCompu_bf:25:6a (08:00:27:bf:25:6a), Dst: RealtekU_12:35:00 (52:54:00:12:35:00)
Internet Protocol Version 4, Src: 10.0.2.15, Dst: 10.141.0.1
User Datagram Protocol, Src Port: 57262, Dst Port: 53
Domain Name System (query)
[Response In: 3]
Transaction ID: 0x2cfe
Flags: 0x0100 Standard query
Questions: 1
Answer RRs: 0
Authority RRs: 0
Additional RRs: 1
Queries
detectportal.firefox.com: type A, class IN
Name: detectportal.firefox.com
[Name Length: 24]
[Label Count: 3]
Type: A (Host Address) (1)
Class: IN (0x0001)
Additional records
0000 52 54 00 12 35 00 08 00 27 bf 25 6a 08 00 45 00 RT..5... '.%j..E.
lab
Packets: 8471 · Displayed: 8471 (100.0%) · Load time: 0:0.880
Profile: Default

# TLS: Transport Layer Security

Runs on top TCP, and it basically ensures the stream is encrypted

Has a handshake after TCP handshake but before application data

Used in HTTPS and many other protocols

# Lab: ctf.notanexploit.club