Cleartext Protocol Analysis
FTP

Looking cleartext protocol traces sounds easy, but when the time comes to investigate a big network trace for incident analysis and response, the game changes.
(huge and messy)

So we need a structured approach, not just follow stream and read

Cleartext protocol = protocols where contents can be read directly in a packet capture

Example: HTTP, FTP, Telnet, SMTP, POP3/IMAP
without HTTPS or TLS

In a lab, we can just chick the packet and follow TCP Stream and read usernames, URLs, commands.

In real incident, the packet capture might have millions of packets, multiple conversations, noise and partial connections.

So just reading isn't enough

So, security analyst needs to produce evidence and conclusions:

- who talked to whom (TOP IPs/hosts)
- what cleartext credentials or sensitive data appeared
- what commands were run
- timelines, counts, anomalies (spikes, unusual ports, repeated logins)
- key indicators: suspicious domains, URls, file transfer

FTP (File Transfer Protocols) is older network protocol used to move files Between a client and server over a network.

- what it's for: uploading / downloading files
- Ports: commonly FTP uses TCP (Transmission Control Portocol)
  21/TCP
  20/TCP
- Issue: usernames, passwords, commands can be seen in Wireshark if someone captures the traffic
  - MIMT attack
  - Credential stealing and unauthorised access
  - Phishing
  - Malware planting
  - Data exfiltration

Wireshark

Global search ftp

- x1x series : Information request responses

221 → 211 → ~~File~~ System status
212 → Diretory status → Directory status
213 → File status

```
ftp. response. code = = 211
ftp. response. code = = 212
ftp. response. code = = 213
```

- x2x series : Connection messages

220 → Service ready
227 → Entering passive mode
228 → Long passive mode
229 → Extended passive mode

```
ftp. response. code == 227
ftp. response. code = = 220
ftp. response. code = = 228
ftp. response. code = = 229
```

Upload = STOR
Download = RETR
List folder = LIST/NLST
Delete = DELE
Rename = RNFR
RNTO

```
ftp.request.command
    == "STOR"
```

SYST → what OS
FEAT → features the server support
HELP → ask for supported command
APPE → upload By appendin to an existing file
SITE → change the permissi
MDTM → last modified time

- X3X series → Authentication messages

230 → User login
231 → Uses logout
331 → Valid usesname
430 → Invalid usesname or password
530 → No login, invalid password

```
ftp.response.code == 230
```

- FTP commands

Uses → Username
Pass → Passwosd
CWD → Current wosk directory
List → List

```
ftp.request.command == "User"
ftp.request.command == "PASS"
ftp.request.arg == "password"
```

- Advanced usage

530 → Not logged in

```
ftp.response.code == 530
(ftp.response.code == 530) and (ftp.response.arg contains "password")
```

Questions

How many incorrect login attempts are there?

```
ftp.response.code == 530
```

what is the size of the file accessed by the "ftp" account?

```
ftp.response.code == 213
```

The adversary uploaded a document to the FTP server. what is the filename?

```
ftp.request.command == "RETR"
```

The adversary tried to assign special flags to change and executing permissions of the uploaded file. What is the command used by the adversary?

```
ftp.request.command == "SITE"
```