TShark Challenge II: Directory

## **TShark Challenge II: Directory — Notes**

### **1. Introduction**

* **TShark** is the command-line interface (CLI) version of Wireshark.
* Purpose: To analyze network captures without GUI, useful for automation, remote servers, and scripting.
* Challenge focuses on:
  + Searching directories for .pcap files.
  + Using TShark to extract information from multiple captures.
  + Combining results for investigation.

### **2. Understanding the Scenario**

* A directory contains several PCAP files.
* Goal: Identify network patterns, suspicious activity, or specific protocol usage across **all** PCAPs.
* Challenges:
  + Multiple files = need for looping or batch processing.
  + Extract only **key fields** relevant to the investigation.

### **3. Key TShark Commands**

* **Basic syntax**:

tshark -r file.pcap

* **List available fields**:

tshark -G fields

* **Extract specific fields**:

tshark -r file.pcap -T fields -e field.name

* **Apply filters** (display filter style, same as Wireshark):

tshark -r file.pcap -Y "http.request"

* **Read all .pcap files in a folder**:

tshark -r \*.pcap

### **4. Directory Processing**

* Loop through all PCAP files in a directory (Linux example):

for file in \*.pcap; do

tshark -r "$file" -Y "http" -T fields -e ip.src -e ip.dst

done

* On Windows (PowerShell):

Get-ChildItem \*.pcap | ForEach-Object {

tshark -r $\_.FullName -Y "http" -T fields -e ip.src -e ip.dst

}

### **5. Useful Filters in This Challenge**

* **HTTP requests**:

http.request

* **DNS queries**:

dns.qry.name

* **FTP traffic**:

k

ftp || ftp-data

* **Suspicious patterns** (example: unusual ports):

tcp.port == 4444

### **6. Extracting Directory-Wide Results**

* Merge output to a single file:

for file in \*.pcap; do

tshark -r "$file" -Y "dns" -T fields -e frame.number -e ip.src -e dns.qry.name

done > all\_dns\_queries.txt

* This helps spot patterns like:
  + Repeated queries to same domain.
  + Multiple IPs contacting a single host.

### **7. Advantages of TShark for Directory Analysis**

* **Speed** — CLI is faster than GUI.
* **Automation** — can be integrated into scripts and cron jobs.
* **Filtering at read time** — no need to open files manually.

### **8. Common Pitfalls**

* Forgetting to **quote filters** in shell commands.
* Misusing -Y (display filter) vs -R (deprecated).
* Field names must match exactly from tshark -G fields.

### **9. Conclusion**

* TShark is essential for bulk PCAP analysis, especially in environments without a GUI.
* In "Directory" challenges, key skills include:
  + Iterating through files.
  + Applying consistent filters.
  + Extracting targeted fields for quick intelligence.
* This process is useful for **incident response**, **malware analysis**, and **network forensics**

 