



A/D traffic analysis with PCAP-over-IP

Eyad Issa

CyberChallenge.IT - Workshop 2024
2024-07-06



Who am I?

- Eyad Issa
- 21 yo
- Ing. Informatica @ Unibo



Partecipato a:

- CyberChallenge.IT 2021 @ Unict
- 4th @ OliCyber 2021



Mi occupo di:

- Infrastruttura
- Tooling
- **Web** / Misc / Rev

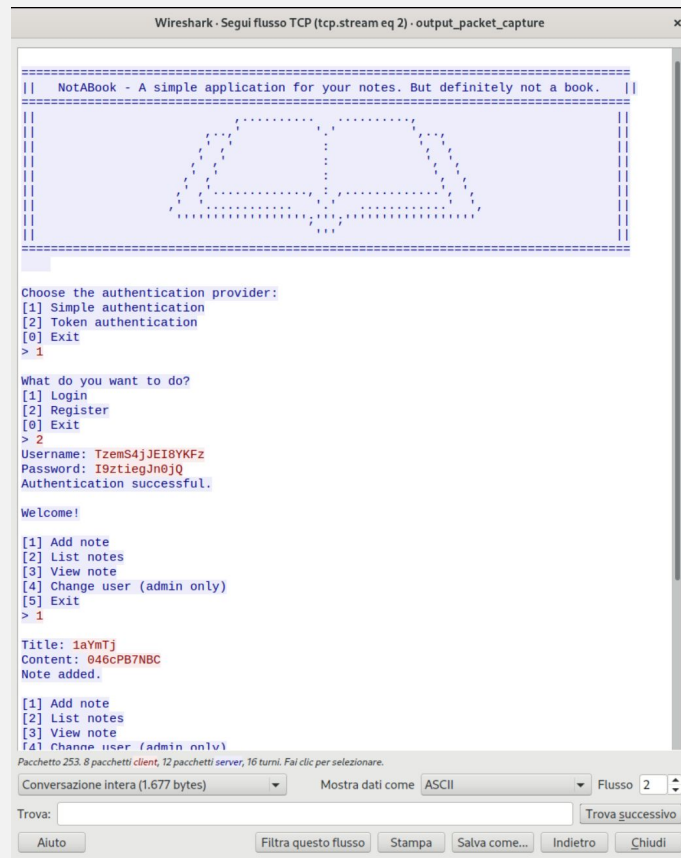
Analisi del traffico

Nelle competizioni Attack Defence, l'**analisi del traffico** ha un ruolo fondamentale per la difesa dei propri servizi (e, se utilizzato bene, per l'attacco di quelli avversari).

Analisi del traffico

La maggior parte del traffico di gara è composta da **flussi TCP non criptati** (con alcune eccezioni).

- Facile da **catturare**: no TLS
- Facile da **analizzare**: la maggior parte dei dati sono caratteri ASCII



```
Wireshark - Segui flusso TCP (tcp.stream eq 2) - output_packet_capture

=====
|| NotABook - A simple application for your notes. But definitely not a book. ||
=====

          _____
         /         \
        /           \
       /             \
      /               \
     /                 \
    /                   \
   /                     \
  /                       \
 /                         \
/                           \
=====

Choose the authentication provider:
[1] Simple authentication
[2] Token authentication
[0] Exit
> 1

What do you want to do?
[1] Login
[2] Register
[0] Exit
> 2
Username: TzemS4jJEI8YKFz
Password: I9ztIiegJn0jQ
Authentication successful.

Welcome!

[1] Add note
[2] List notes
[3] View note
[4] Change user (admin only)
[5] Exit
> 1

Title: 1aYmTj
Content: 046cPB7NBC
Note added.

[1] Add note
[2] List notes
[3] View note
[4] Change user (admin only)

Pacchetto 253. 8 pacchetti client, 12 pacchetti server, 16 turni. Fai clic per selezionare.
Conversazione intera (1.677 bytes)  Mostra dati come ASCII  Flusso 2
Trova:
Aluto  Filtra questo flusso  Stampa  Salva come...  Indietro  Chiudi
```

Analisi del traffico - Tool di analisi tradizionali



Tutti formati da

Dumper

Analizzatore

Analisi del traffico - Tool di analisi per A/D

Caronte

The Caronte interface displays a table of connections with columns for service, srcip, srcport, dstip, dstport, started_at, duration, up, down, and actions. The central pane shows details for a selected connection, including headers like 'Content-Type: application/vnd.tcp.rst+json' and 'User-Agent: Mozilla/5.0'. The bottom pane contains graphs for connections and stream content.

Tulip

The Tulip interface shows a list of traffic items filtered by 'Suricata'. The selected item details include: Message: ICC - Modern Firefox UA observed, Rule ID: 1500006, Action taken: allowed, Source: /traffic/capture-2022-06-16_07:14:39.pcap, and Source - Target: 10.254.0.1:45204 - 10.60.4.1:5000. The detailed view also shows headers like 'Content-Type: application/x-www-form-urlencoded'.

Analisi del traffico - Struttura del network di gioco

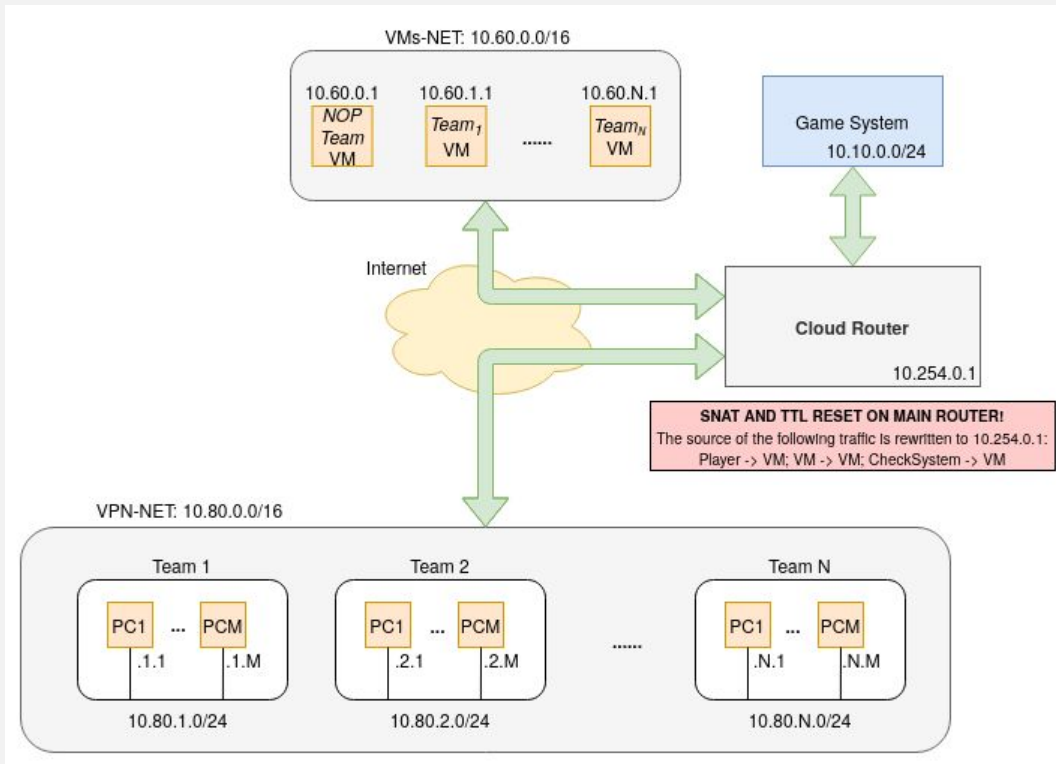


Image taken from rules.ad.cyberchallenge.it. All images are copyright to their respective owners.

Analisi del traffico - Workflow standard

1. Dump dei pacchetti sulla Vuln box
2. Trasferimento dei **PCAP** via rsync o simili
3. Analisi e visualizzazione

Problemi:

- delay tra l'arrivo effettivo del pacchetto e la sua analisi e visualizzazione
- creazione obbligatoria di file PCAP, che devono essere ruotati (o cancellati) per non riempire lo spazio della VM
- script più o meno funzionanti per lanciare rsync ogni tot tempo e avviare l'analisi sui nuovi file (o inotify e altri)

Poco efficiente e complicato senza alcun motivo

Analisi del traffico - Workflow alternativo 1

1. Setup dei tool di analisi direttamente sulla vulnbox
2. Utilizzo (si spera) di tunnel ssh per accedere ai servizi web



Gaspare 03/07/2024 20:10



SUPER MEGA IMPORTANTE!!!!!!!!!!!!!!!!!!!!!!



Problemi:

- overhead per la vulnbox!
- spesso i container docker sono grandi, e lo spazio sulla VM è poco!
- stessa cosa per il database che i vari caronte / tulip utilizzano :/
- non è scalabile: se voglio analizzare il traffico con 3 tool diversi -> tutti sulla macchina

Davvero abbiamo delle vulnbox così potenti?

Analisi del traffico - Workflow alternativo 2

1. Utilizzo di tunnel ssh e pipe per lanciare un dumper sulla macchina remota
2. Analisi i risultati in locale

```
ssh root@10.60.2.1 "tcpdump -i game -n --immediate-mode -s 65535 -U -w - port 1234"  
| wireshark -k -i -
```

- Cosa succede se voglio analizzare con più software diversi?
- Cosa succede se più persone vogliono analizzare il traffico?

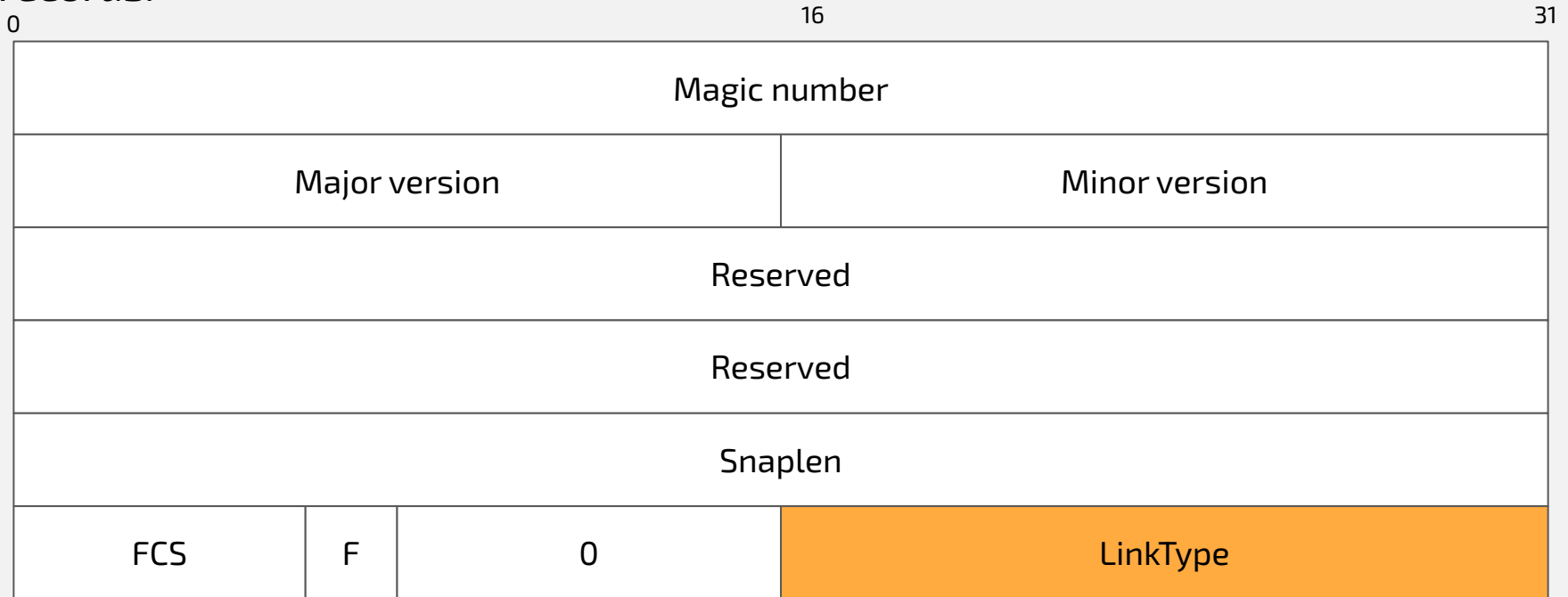
Risposta: Più processi tcpdump, più connessioni ssh (e traffico)

Funziona? Sì! Potremmo fare di meglio? Anche!

I file PCAP

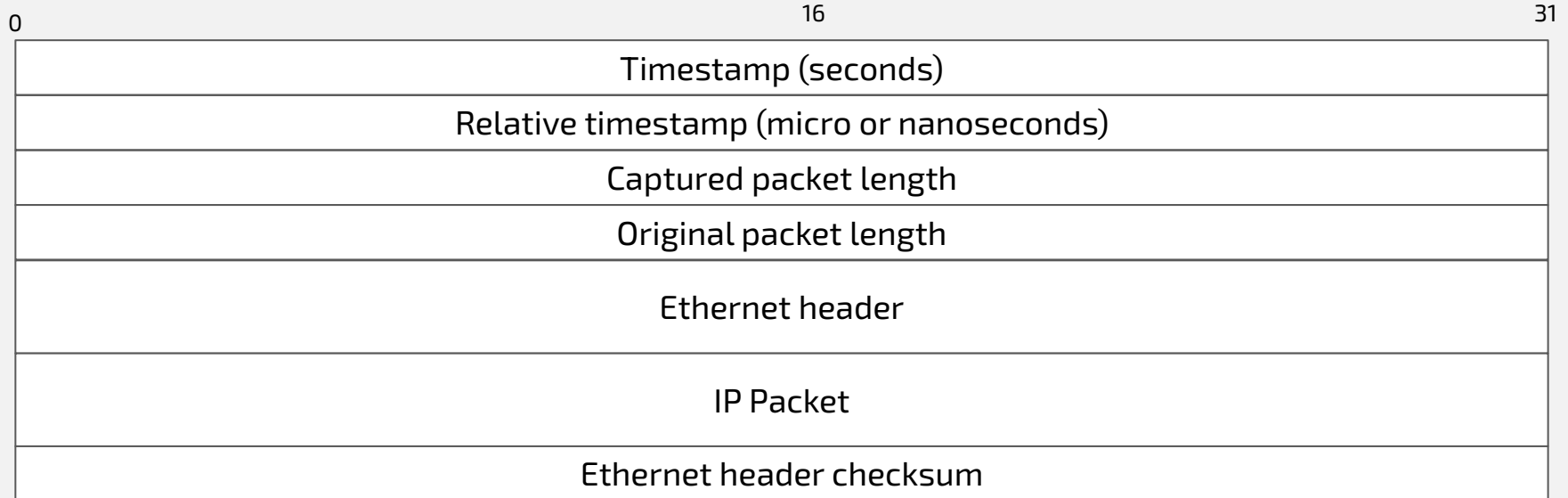
Cos'è un file PCAP?

The PCAP file structure includes a file header followed by zero or more packet records.



Cos'è un file PCAP?

In a PCAP file, the file header is followed by a series of packet records. Each packet record represents a single packet captured from the network, along with a timestamp, the length of the packet, and the length captured from that packet.



PCAP

“The PCAP file is a simple flat file with no index and no extensibility. PCAP files were never designed for managing packet captures larger than a few hundred Mbytes. Beyond this point reading, filtering, and searching a PCAP file can become a slow and tedious exercise – as anyone who’s opened a large PCAP file in Wireshark will tell you!”

From [What is a PCAP file?](#)

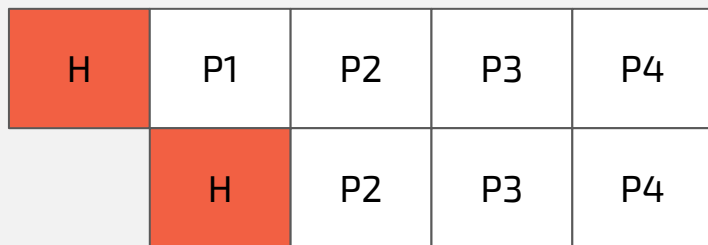
PCAP = tar, ma per i pacchetti

E quindi?

E quindi non sono costretto a leggere tutto il PCAP ogni volta!

Mi bastano:

- L'header originale
- I record di ogni pacchetto



File ben formato

File ben formato

PCAP-over-IP

PCAP-over-IP

“PCAP-over-IP is a method for transmitting captured network traffic through a TCP connection. The captured network traffic is transferred over TCP as a PCAP file in order to preserve relevant metadata about the packets, such as timestamps.”

From [Wikipedia contributors, 'PCAP-over-IP', Wikipedia, The Free Encyclopedia](#)

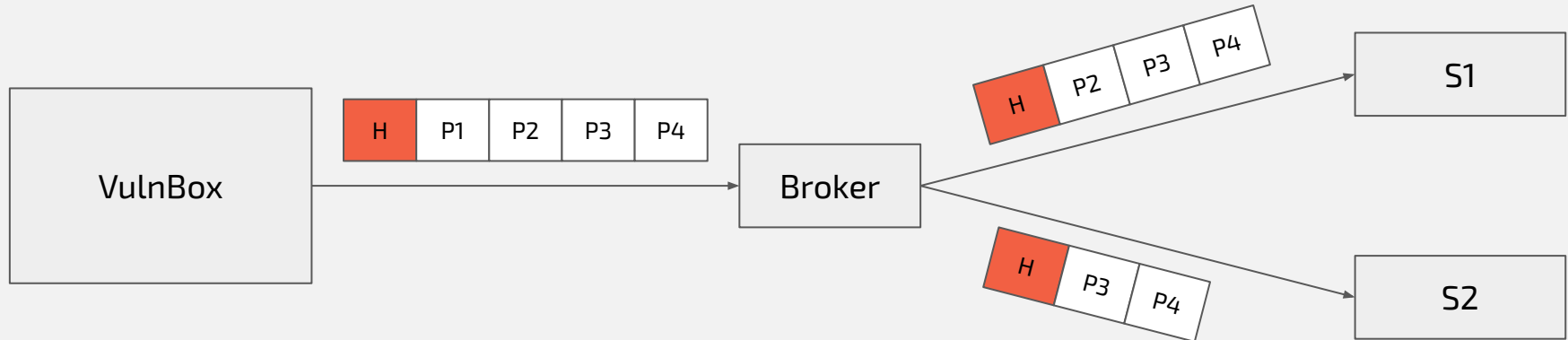
L'idea è quindi quella di **creare un file PCAP** *on the fly* e **trasmetterlo via rete**, un po' quello che facevamo nell'esempio di "workflow alternativo 2"!

Possiamo però sfruttare la struttura dei file PCAP per migliorare l'approccio e renderlo estendibile all'infinito.

Cosa vogliamo fare?

Vorremmo creare un servizio che:

- accetti in ingresso un file PCAP
- fornisca un servizio a cui ci si può connettere (TCP)
- alla connessione, scriva l'header del file PCAP letto dal file in ingresso
- inizi a fare da relay tra i pacchetti ricevuti in ingresso e i servizi in uscita



pcap-broker

Esiste già? Per fortuna si! (fork di [fox-it/pcap-broker](https://github.com/fox-it/pcap-broker))

github.com/UlisseLab/pcap-broker

Come si usa?

```
sudo tcpdump -i eth0 -n --immediate-mode -s 65535 -U -w - | ./pcap-broker
```

Come si usa per catturare il traffico di una macchina remota?

```
ssh root@10.60.2.1 "tcpdump -i game -n --immediate-mode -s 65535 -U -w - not port 22"  
| ./pcap-broker
```

pcap-broker

```
func processPackets(ctx context.Context, source *gopacket.PacketSource, clients clientMap) {
    for {
        select {
        case ←ctx.Done(): return
        case packet := ←source.Packets():
            for conn, client := range clients {
                // if the connection is closed, remove it from the map
                err := client.SendPacket(packet)
                if err ≠ nil {...}
            }
        }
    }
}
```

Ma perchè?

Infiniti vantaggi:

- Overhead minimo nella macchina (1 sola istanza di tcpdump)
- Riutilizzabilità dello stesso flusso dati per più applicazioni
 - tulip + suricata + caronte + ...
 - ogni player può tenere il proprio wireshark aperto
 - voglio salvarmi tutto il traffico per fare *machine learning* 🚀🚀🚀
 - voglio fare tutto ciò su 10 macchine diverse
- Non devo salvare, rsyncare, ruotare, ecc.
- Il delay tra l'arrivo del pacchetto e la sua analisi è pressoché nullo

Pochi svantaggi:

- Non posso bloccare i pacchetti (eh direi, è un file pcap!)
- Non posso “navigare indietro nel tempo”. Devo salvare tutti i dati utili appena li vedo.

Interoperabilità

How to?

Come si guarda il traffico?

Software che supportano nativamente PCAP-over-IP

- `tshark -i TCP@127.0.0.1:4242`
 - oppure `socat TCP:127.0.0.1:4242 - | tshark -i -`
- `wireshark -k -i TCP@127.0.0.1:4242`
 - oppure `socat TCP:127.0.0.1:4242 - | wireshark -k -i -`

Software che non lo supportano:

- `socat TCP:127.0.0.1:4242 - | suricata -r /dev/stdin`
 - oppure (bash only) `suricata -r /dev/tcp/127.0.0.1/4242`
 - <https://redmine.openinfosecfoundation.org/issues/5499>

Esempi

Utilizzo di **tshark** per filtrare il traffico e darlo in pasto ad altro tool!

```
socat TCP:127.0.0.1:4242 - \  
    | tshark -r - -w - -Y 'ip.dst == 10.60.2.1'  
    | suricata -r /dev/stdin
```

Scrivo un sw che accetta file PCAP da stdin:

```
socat TCP:127.0.0.1:4242 - \  
    | go run ./cmd/ -p -data -
```

pcapReadMethod %

libpcap

Specify how packets are read from network cards:

- `afpacketv3` = Use linux `tpacketv3` (`afpacket`) interface
- `daq` = Use `daq`, requires `rootPlugins=reader-daq.so`
- `libpcap` = Use `libpcap`
- `pcap-over-ip-client` = Use `pcap` over `ip`, where Arkime is the client
- `pcap-over-ip-server` = Use `pcap` over `ip`, where Arkime is the server
- `pfring` = Use vanilla `pfring` directly, requires `rootPlugins=reader-pfring.so`
- `snf` = Use Myricom `snf`, requires `rootPlugins=reader-snf.so`
- `tpacketv3` = Use linux `tpacketv3` (`afpacket`) interface - RECOMMENDED
- `tzsp` = Listen for forwarded packets using `tzsp`

Appendix A: L'admin con poca banda

```
ssh -R 3333:localhost:3333 root@...
```

```
inotifywait -m "$PCAP_DIR" -e close_write -e moved_to |  
  while read dir action file; do  
    curl -F "file=@$file" "http://localhost:3333/api/pcap/upload"  
  done
```

Appendix A: L'admin con poca banda

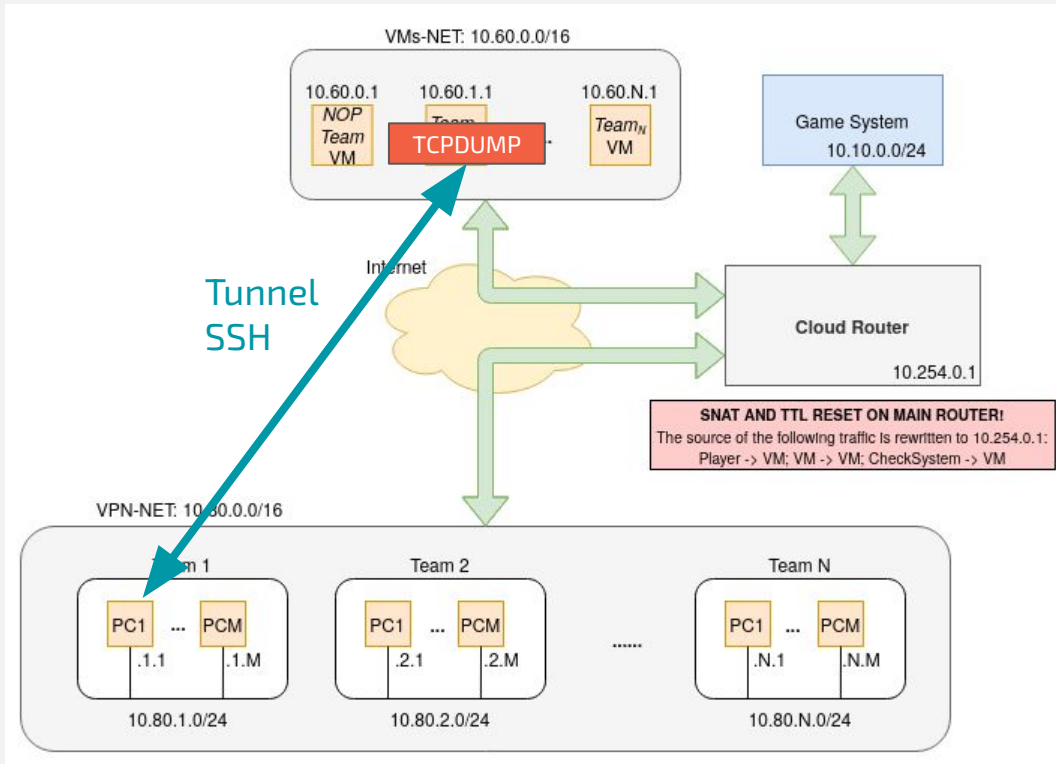


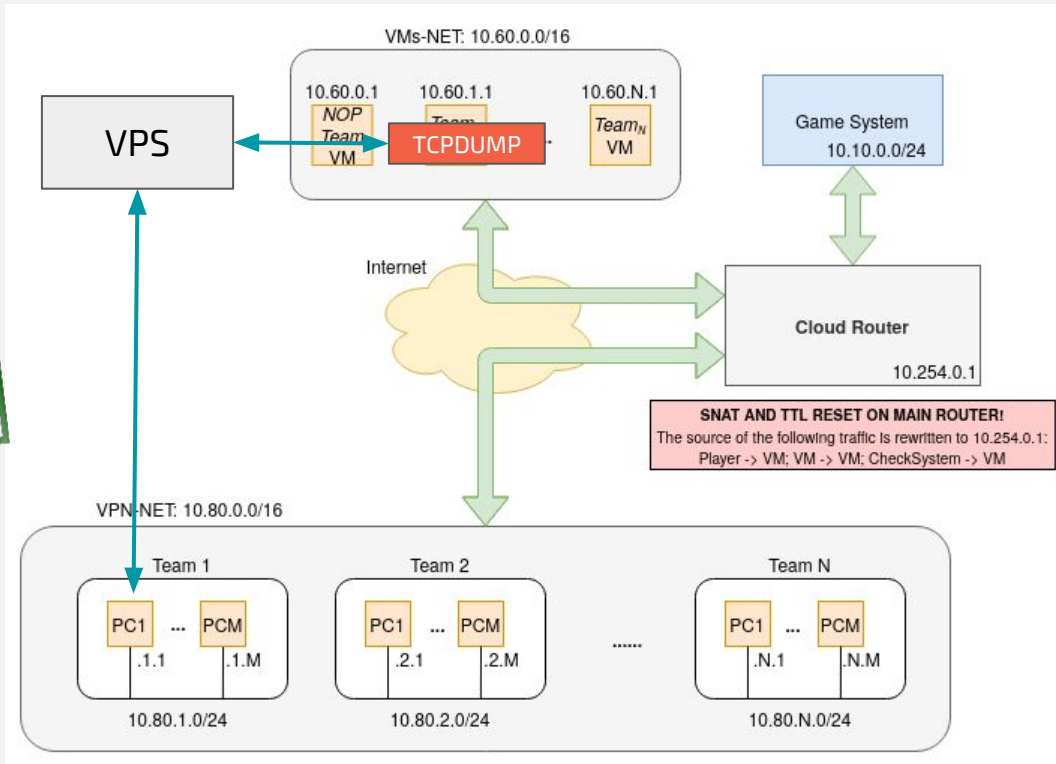
Image taken from rules.ad.cyberchallenge.it. All images are copyright to their respective owners.

Appendix A: L'admin con poca banda

```
ssh -R 3333:localhost:3333 root@...
```

```
inotifywait -m "$PCAP_DIR" -e close_write -e moved_to |  
  while read dir action file; do  
    echo "you're using too much bandwidth!"  
    # curl -F "file=@$file" "http://localhost:3333/api/pcap/upload"  
  done
```

Appendix A: L'admin con poca banda

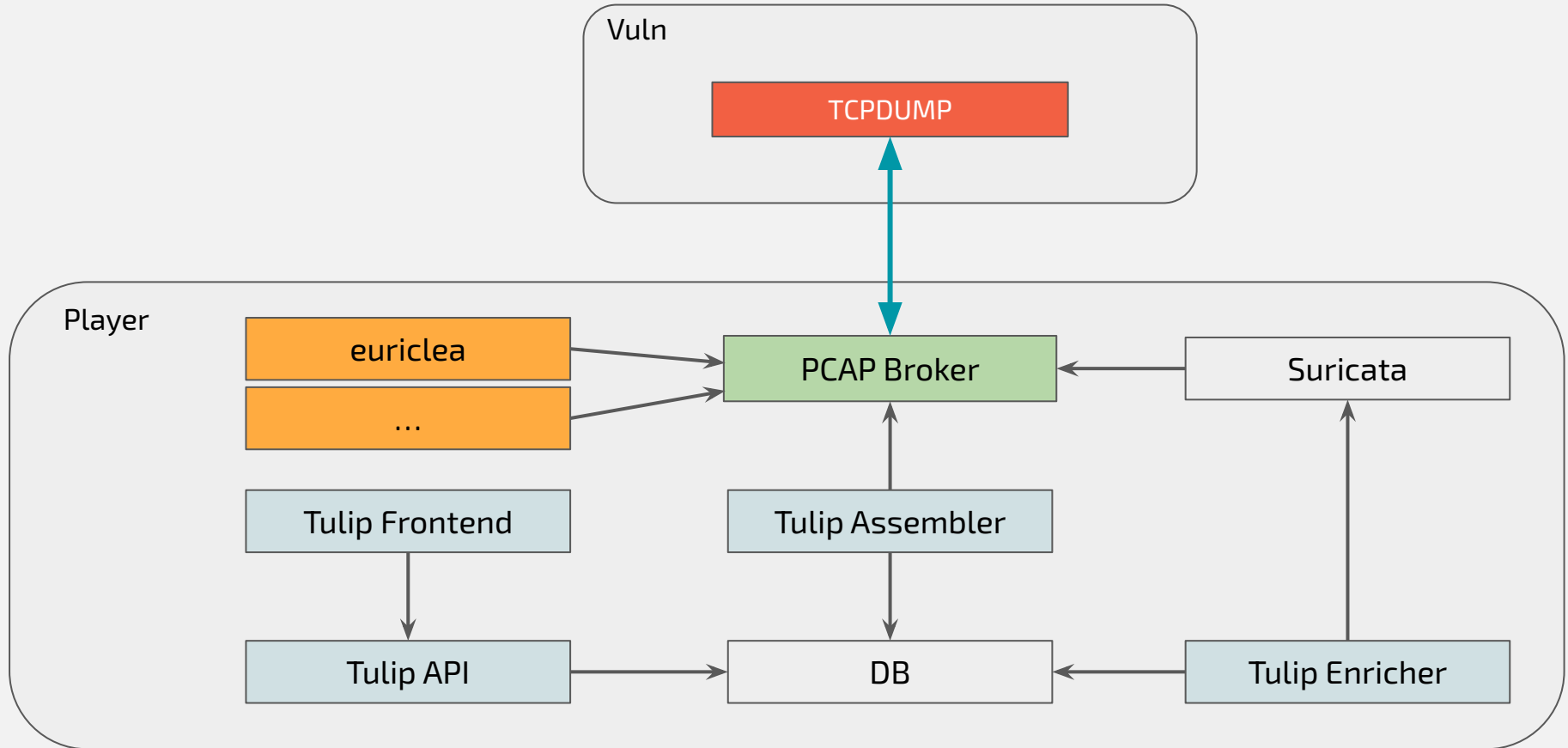


APPROVATO

APPROVATO

Image taken from rules.ad.cyberchallenge.it. All images are copyright to their respective owners.

Architettura di difesa: Ulisse



Sviluppi futuri

Sviluppi futuri

- Creazione di tool che lavorano su un flusso PCAP
 - *Fingerprinting TCP/IP* 🦴🦴🦴
 - *Abbiamo un tool WIP...*
 - Integrazione con tulip via "enrichers"
- Utilizzo di software "à la" [Malcolm](#) in competizioni A/D
- ...

Grazie! Domande?