

Challenge 2 : ICMP Exfiltration

Contexte : Nous avons remarqué des échanges curieux avec le serveur dash.pasfastoche.lan. Il semblerait que de nombreuses requêtes ping soient envoyées par là-bas. Trouvez ce qu'il s'y trame..

Hint : Les trames intéressantes possède un ttl modifié.

=> Les trames modifiées ont un ttl à 20. Pour filtrer ceci avec wireshark, ip.ttl==20 pour passer à la Step 2.

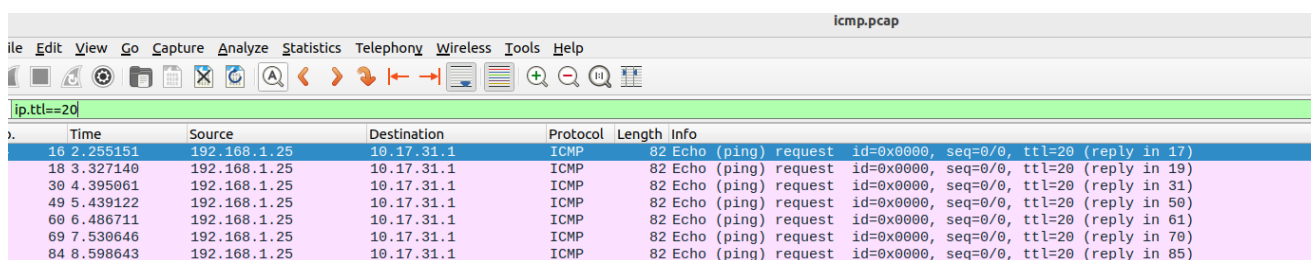
Step 1 : Retrouver les trames intéressantes

Il faut tout d'abord retrouver l'IP de la machine distante :

dns.qry.name=="dash.pasfastoche.lan", nous retrouvons 10.17.31.1, on va pouvoir s'en servir pour filtrer un peu plus avec ip.addr==10.13.31.1.

Parmi toutes les trames envoyées, seules les trames ICMP à destination de la machine 10.17.31.254 nous intéressent.

Seulement, il faut aussi faire attention, seules celles avec un TTL à 20 sont vraiment intéressantes ^:)



The image shows a Wireshark packet capture of ICMP Echo (ping) requests. The filter bar at the top is set to 'ip.ttl==20'. The packet list shows several ICMP Echo (ping) requests from source 192.168.1.25 to destination 10.17.31.1, all with TTL=20. The packet details pane shows the structure of an ICMP Echo request: Type 8, Code 0, Identifier 0x0000, and Sequence Number 0.

No.	Time	Source	Destination	Protocol	Length	Info
16	2.255151	192.168.1.25	10.17.31.1	ICMP	82	Echo (ping) request id=0x0000, seq=0/0, ttl=20 (reply in 17)
18	3.327140	192.168.1.25	10.17.31.1	ICMP	82	Echo (ping) request id=0x0000, seq=0/0, ttl=20 (reply in 19)
30	4.395061	192.168.1.25	10.17.31.1	ICMP	82	Echo (ping) request id=0x0000, seq=0/0, ttl=20 (reply in 31)
49	5.439122	192.168.1.25	10.17.31.1	ICMP	82	Echo (ping) request id=0x0000, seq=0/0, ttl=20 (reply in 50)
60	6.486711	192.168.1.25	10.17.31.1	ICMP	82	Echo (ping) request id=0x0000, seq=0/0, ttl=20 (reply in 61)
69	7.539646	192.168.1.25	10.17.31.1	ICMP	82	Echo (ping) request id=0x0000, seq=0/0, ttl=20 (reply in 70)
84	8.598643	192.168.1.25	10.17.31.1	ICMP	82	Echo (ping) request id=0x0000, seq=0/0, ttl=20 (reply in 85)

Step 2 : Se rendre compte de ce qui est **vraiment** intéressant

Maintenant qu'on a les bonnes trames, il faut également se rendre compte que le champ data est modifié et ce qui est envoyé est en base64.

On peut voir que la dernière trame se termine par '==', c'est caractéristique d'un base64.

Step 3 : Reconstruire le base64

Ecrire un script permettant de reconstruire le base64 en entier, par exemple :

```
from scapy.all import *

pcap_flow = rdpcap('icmp.pcap')

solv = list()
for packet in pcap_flow:

    if 'ICMP' in packet and packet.ttl == 20:
        payload = packet[ICMP].payload
        solv.append(payload.load.decode("utf-8"))
```

```
print("Base64 : ", end = '')
print(''.join(solv))
```

Step 4 : Retrouver l'information utile

Premier reflexe : <https://base64decode.org>

Decode from Base64 format

Simply enter your data then push the decode button.

[illegible]

i For encoded binaries (like images, documents, etc.) use the file upload form a little further down on this page.









































ASCII Source character set.

☐ Decode each line separately (useful for when you have multiple entries).

 Live mode OFF Decodes in real-time as you type or paste (supports only the UTF-8 character set).

< DECODE > Decodes your data into the area below.

PNG

Intéressant : On observe qu'il s'agit d'une image PNG !

Deuxième reflexe : google : base64 to PNG
Base64 to Image

Convert Base64 to image online using a free decoding tool which allows you to decode Base64 as image and preview it directly in the browser. In addition, you will receive some basic information about this image (resolution, MIME type, extension, size). And, of course, you will have a special link to download the image to your device. If you are looking for the reverse process, check [Image to Base64](#).

Base64* [copy](#) [clear](#) [download](#)

[illegible]

Decode Base64 to Image

Preview Image | [Toggle Background Color](#)

```
flag = "24IUT{IcmpExfiltrationIsEasy}"
```

Bingo !