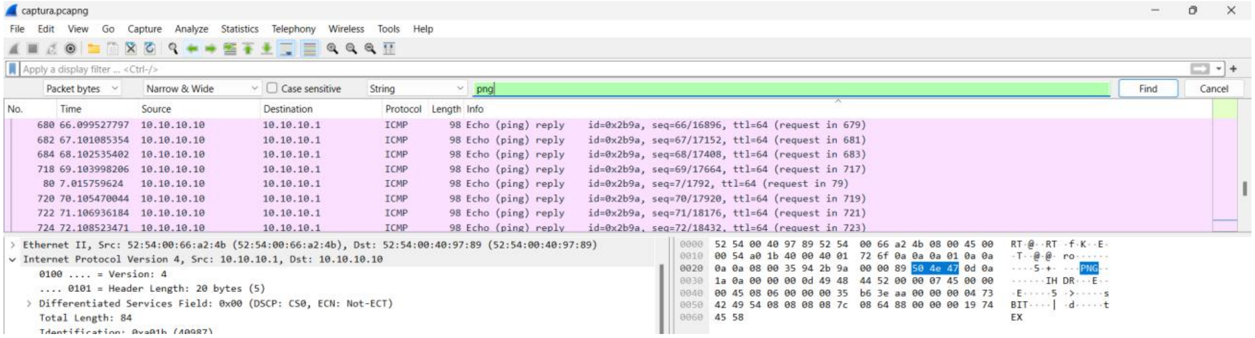
Since we know we are looking for a screenshot, which are usually png's, we can assume that we can find the header of the file by searching for the string PNG, as from the challenge description it seems the transfer wasn't made safely.


captura.pcapng
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
Packet bytes
Narrow & Wide
□ Case sensitive
String
png
Find
Cancel

No.	Time	Source	Destination	Protocol	Length	Info
680	66.099527797	10.10.10.10	10.10.10.1	ICMP	98	Echo (ping) reply id=0x2b9a, seq=66/16896, ttl=64 (request in 679)
682	67.101085354	10.10.10.10	10.10.10.1	ICMP	98	Echo (ping) reply id=0x2b9a, seq=67/17152, ttl=64 (request in 681)
684	68.102535402	10.10.10.10	10.10.10.1	ICMP	98	Echo (ping) reply id=0x2b9a, seq=68/17408, ttl=64 (request in 683)
718	69.103998206	10.10.10.10	10.10.10.1	ICMP	98	Echo (ping) reply id=0x2b9a, seq=69/17664, ttl=64 (request in 717)
80	7.015759624	10.10.10.10	10.10.10.1	ICMP	98	Echo (ping) reply id=0x2b9a, seq=7/1792, ttl=64 (request in 79)
720	70.105470044	10.10.10.10	10.10.10.1	ICMP	98	Echo (ping) reply id=0x2b9a, seq=70/17920, ttl=64 (request in 719)
722	71.106936184	10.10.10.10	10.10.10.1	ICMP	98	Echo (ping) reply id=0x2b9a, seq=71/18176, ttl=64 (request in 721)
724	72.108523471	10.10.10.10	10.10.10.1	ICMP	98	Echo (ping) reply id=0x2b9a, seq=72/18432, ttl=64 (request in 723)

```
> Ethernet II, Src: 52:54:00:66:a2:4b (52:54:00:66:a2:4b), Dst: 52:54:00:40:97:89 (52:54:00:40:97:89)
v Internet Protocol Version 4, Src: 10.10.10.1, Dst: 10.10.10.10
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 84
    Identification: 0xa01b (40987)
```

```
0000  52 54 00 40 97 89 52 54  00 66 a2 4b 08 00 45 00   RT·@··RT ·f·K··E·
0010  00 54 a0 1b 40 00 40 01  72 6f 0a 0a 0a 01 0a 0a   ·T··@·@· ro······
0020  0a 0a 08 00 35 94 2b 9a  00 00 89 50 4e 47 0d 0a   ····5·+· ···PNG··
0030  1a 0a 00 00 00 0d 49 48  44 52 00 00 07 45 00 00   ······IH DR···E··
0040  00 45 08 06 00 00 00 35  b6 3e aa 00 00 00 04 73   ·E·····5 ·>·····s
0050  42 49 54 08 08 08 08 7c  08 64 88 00 00 00 19 74   BIT····| ·d·····t
0060  45 58                                              EX
```

ICMP? That's weird. Turns out, ICMP allows the user to send arbitrary data when pinging. Therefore, we need to extract, hex decode and concatenate all the "data" fields of the ICMP packets (only request or only response):

tshark -r captura.pcapng -Y "icmp.type == 8" -T fields -e data | xxd -r -p > image.png

After this, we have a valid png which contains the password….?

In this case, we are dealing with a type 7 cisco password, which is not encrypted, rather just obfuscated, since it is easily reversible, no bruteforce needed. A website like https://ccnax.com/cisco-type-7-password-decryption/ easily reveals the flag.

Made with love by: AndreiCat