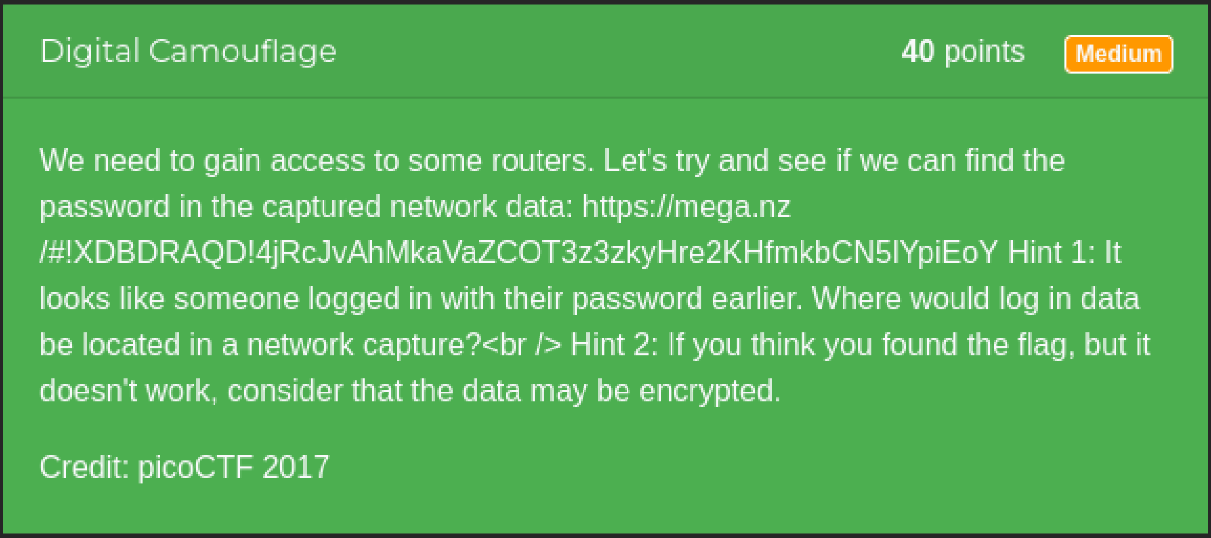
# Digital Camouflage CTF Writeup

This document is a walkthrough on one way to solve the **Digital Camouflage CTF** on **CTFLearn**. The objective is to explain how I was able to solve this CTF to my future self.

## General Information

- *Difficulty:* Easy / Medium
- *Category:* Forensics
- *Link:* [Challenge - Digital Camouflage - CTFlearn - CTF Practice]()

## Introduction

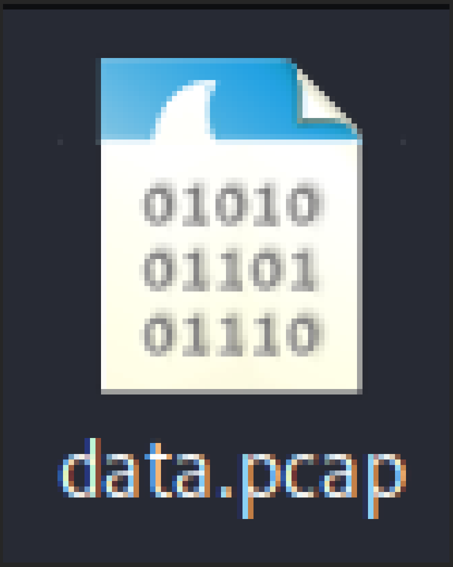


Digital Camouflage 40 points Medium

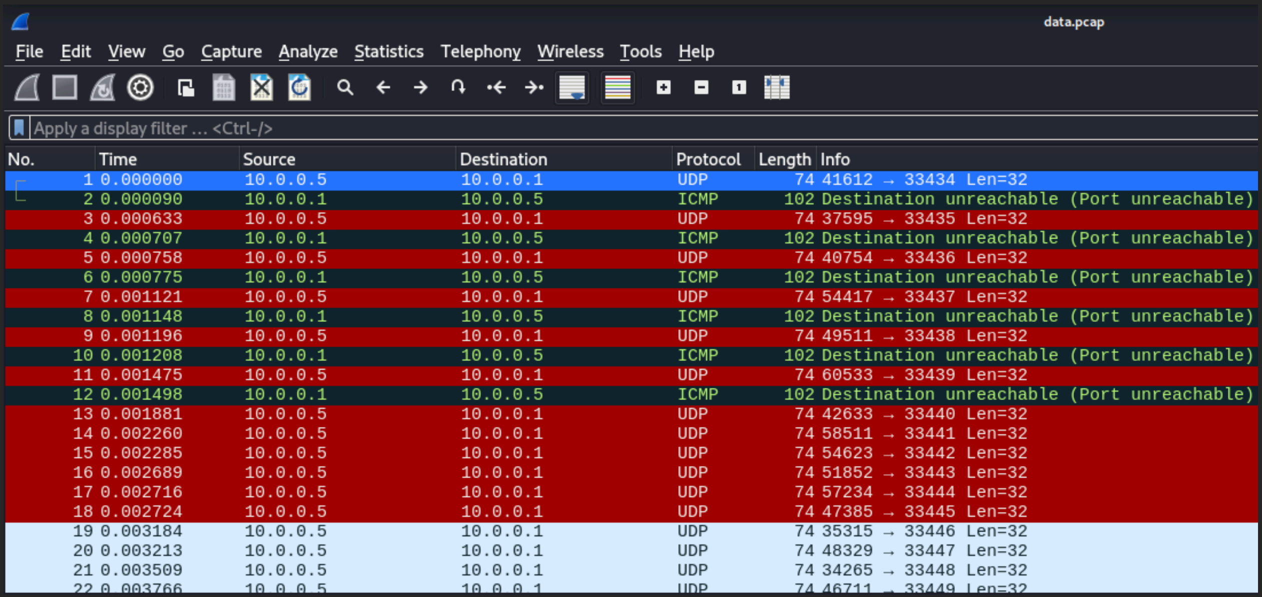
We need to gain access to some routers. Let's try and see if we can find the password in the captured network data: https://mega.nz /#!XDBDRAQD!4jRcJvAhMkaVaZCOT3z3zkyHre2KHfmkbCN5lYpiEoY Hint 1: It looks like someone logged in with their password earlier. Where would log in data be located in a network capture?<br /> Hint 2: If you think you found the flag, but it doesn't work, consider that the data may be encrypted.

Credit: picoCTF 2017

The link sends us to the download of a PCAP file, which contains packet capture data, which stems from network traffic.



We're going to use Wireshark, a network-analysing tool to view the contents of this file

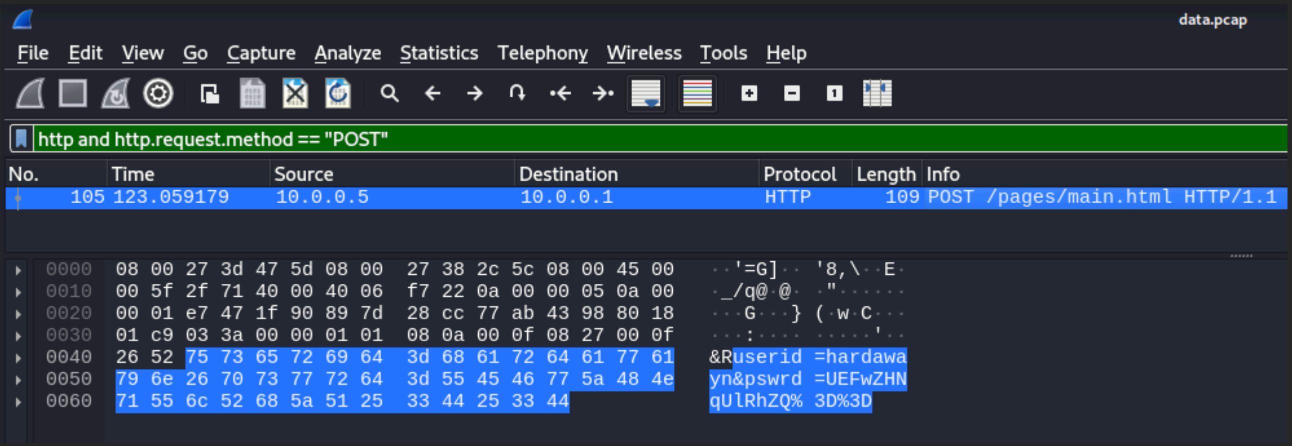

data.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>

```
No.  Time      Source    Destination Protocol Length Info
   1 0.000000  10.0.0.5  10.0.0.1    UDP       74 41612 → 33434 Len=32
   2 0.000090  10.0.0.1  10.0.0.5    ICMP     102 Destination unreachable (Port unreachable)
   3 0.000633  10.0.0.5  10.0.0.1    UDP       74 37595 → 33435 Len=32
   4 0.000707  10.0.0.1  10.0.0.5    ICMP     102 Destination unreachable (Port unreachable)
   5 0.000758  10.0.0.5  10.0.0.1    UDP       74 40754 → 33436 Len=32
   6 0.000775  10.0.0.1  10.0.0.5    ICMP     102 Destination unreachable (Port unreachable)
   7 0.001121  10.0.0.5  10.0.0.1    UDP       74 54417 → 33437 Len=32
   8 0.001148  10.0.0.1  10.0.0.5    ICMP     102 Destination unreachable (Port unreachable)
   9 0.001196  10.0.0.5  10.0.0.1    UDP       74 49511 → 33438 Len=32
  10 0.001208  10.0.0.1  10.0.0.5    ICMP     102 Destination unreachable (Port unreachable)
  11 0.001475  10.0.0.5  10.0.0.1    UDP       74 60533 → 33439 Len=32
  12 0.001498  10.0.0.1  10.0.0.5    ICMP     102 Destination unreachable (Port unreachable)
  13 0.001881  10.0.0.5  10.0.0.1    UDP       74 42633 → 33440 Len=32
  14 0.002260  10.0.0.5  10.0.0.1    UDP       74 58511 → 33441 Len=32
  15 0.002285  10.0.0.5  10.0.0.1    UDP       74 54623 → 33442 Len=32
  16 0.002689  10.0.0.5  10.0.0.1    UDP       74 51852 → 33443 Len=32
  17 0.002716  10.0.0.5  10.0.0.1    UDP       74 57234 → 33444 Len=32
  18 0.002724  10.0.0.5  10.0.0.1    UDP       74 47385 → 33445 Len=32
  19 0.003184  10.0.0.5  10.0.0.1    UDP       74 35315 → 33446 Len=32
  20 0.003213  10.0.0.5  10.0.0.1    UDP       74 48329 → 33447 Len=32
  21 0.003509  10.0.0.5  10.0.0.1    UDP       74 34265 → 33448 Len=32
  22 0.003766  10.0.0.5  10.0.0.1    UDP       74 46711 → 33449 Len=32
```

The challenge indicates that the packet data contains information about someone logging in their password, and received by the router.

Now, this is very commonly done via the HTTP protocol, using the POST method, which basically requests the server (in this case the router) to accept its data.

It is under this context that we will filter the packet to only HTTP via POST method requests

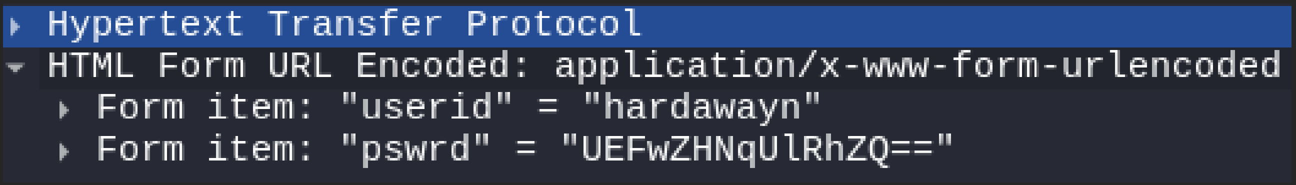

data.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

```
http and http.request.method == "POST"
```

```
No.  Time        Source    Destination Protocol Length Info
 105 123.059179  10.0.0.5  10.0.0.1    HTTP      109 POST /pages/main.html HTTP/1.1
```

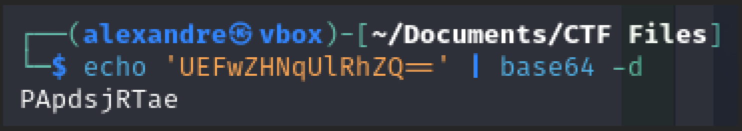
```
0000  08 00 27 3d 47 5d 08 00  27 38 2c 5c 08 00 45 00   ··'=G]·· '8,\··E·
0010  00 5f 2f 71 40 00 40 06  f7 22 0a 00 00 05 0a 00   ·_/q@·@· ·"······
0020  00 01 e7 47 1f 90 89 7d  28 cc 77 ab 43 98 80 18   ···G···} (·w·C···
0030  01 c9 03 3a 00 00 01 01  08 0a 00 0f 08 27 00 0f   ···:···· ·····'··
0040  26 52 75 73 65 72 69 64  3d 68 61 72 64 61 77 61   &Ruserid =hardawa
0050  79 6e 26 70 73 77 72 64  3d 55 45 46 77 5a 48 4e   yn&pswrd =UEFwZHN
0060  71 55 6c 52 68 5a 51 25  33 44 25 33 44            qUlRhZQ% 3D%3D
```

Taking a closer look:



```
▸ Hypertext Transfer Protocol
▾ HTML Form URL Encoded: application/x-www-form-urlencoded
    ▸ Form item: "userid" = "hardawayn"
    ▸ Form item: "pswrd" = "UEFwZHNqUlRhZQ=="
```

*UEFwZHNqUlRhZQ==* looks like a base64-encrypted string, let's decode it:



```
┌──(alexandre㉿vbox)-[~/Documents/CTF Files]
└─$ echo 'UEFwZHNqUlRhZQ==' | base64 -d
PApdsjRTae
```

And there you have it, the flag is **PApdsjRTae**.