

# If we upload the image on **Cyberchef**, we can see that **0xffd9** appears 2 times. More specifically, at the bottom of the file : **ff d9 88 9f 8d a7 ae aa b9 a5 b0 9e a9 be a5 bf be 94 b9 fb a8 a0 fe b6 ff d9**. For the previous hex code, I used this recipe on Cyberchef :

The screenshot shows the CyberChef web interface. On the left, the 'From Hex' tool is active with 'Delimiter' set to 'Auto'. Below it, the 'XOR Brute Force' tool is configured with 'Key length' 2, 'Sample length' 100, and 'Sample offset' 0. The 'Scheme' is 'Standard', 'Null preserving' is unchecked, 'Print key' is checked, and 'Output as hex' is unchecked. A 'Crib (known plaintext string)' field is empty. On the right, the hex input is displayed as '88 9f 8d a7 ae aa b9 a5 b0 9e a9 be a5 bf be 94 b9 fb a8 a0 fe b6 ff'. The 'Output' section shows a list of keys, with the key 'CTFlearn{Ubuntu\_r0ck5}' highlighted in yellow.

From Hex

Delimiter  
Auto

XOR Brute Force

Key length  
2

Sample length  
100

Sample offset  
0

Scheme  
Standard

☐ Null preserving ☒ Print key

☐ Output as hex

Crib (known plaintext string)

88 9f 8d a7 ae aa b9 a5 b0 9e a9 be a5 bf be 94 b9 fb a8 a0 fe b6 ff

Output

learn next previous all ☐ match case ☐ regexp ☐ by word

Key = c0c4: C{rcenra{z0zri{urrc05r4  
Key = cbc5: CZFbeor`{[b{nzuQr>ce5s4  
Key = cbc6: CYFaelrc{XbxnyuRr=cf5p4  
Key = cbc7: CXF`emrb{YbynxuSr<cg5q4  
Key = cbc8: CWFoebrrm{Vbvnwu\r3ch5~4  
Key = cbc9: CVFnecr1{Wbwnvu]r2ci5•4  
Key = cbca: CUFme`ro{Tbtuu^r1cj5|4  
Key = cbc b: **CTFlearn{Ubuntu\_r0ck5}4**  
Key = cbcc: CSFkefri{RbrnsuXr7cl5z4  
Key = cbcd: CRFjegrh{SbsnruYr6cm5{4  
Key = cbce: CQFiedrk{PbpnquZr5cn5x4

THE FLAG : CTFlearn{Ubuntu\_r0ck5}  
~Z4que