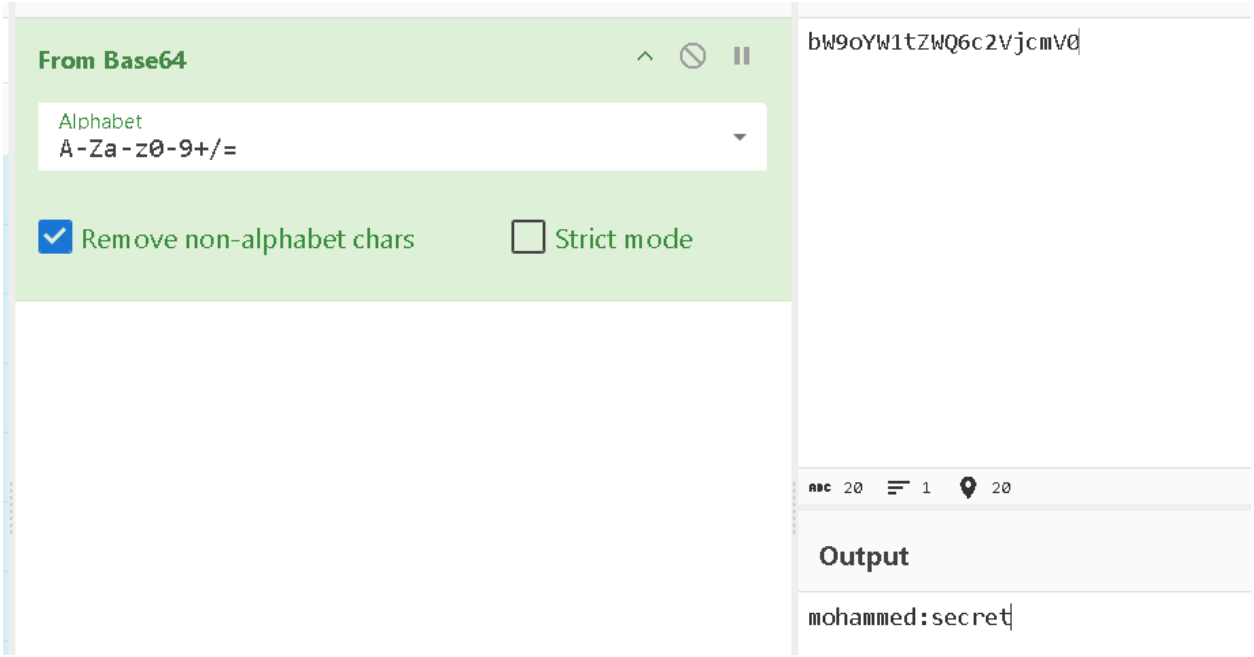
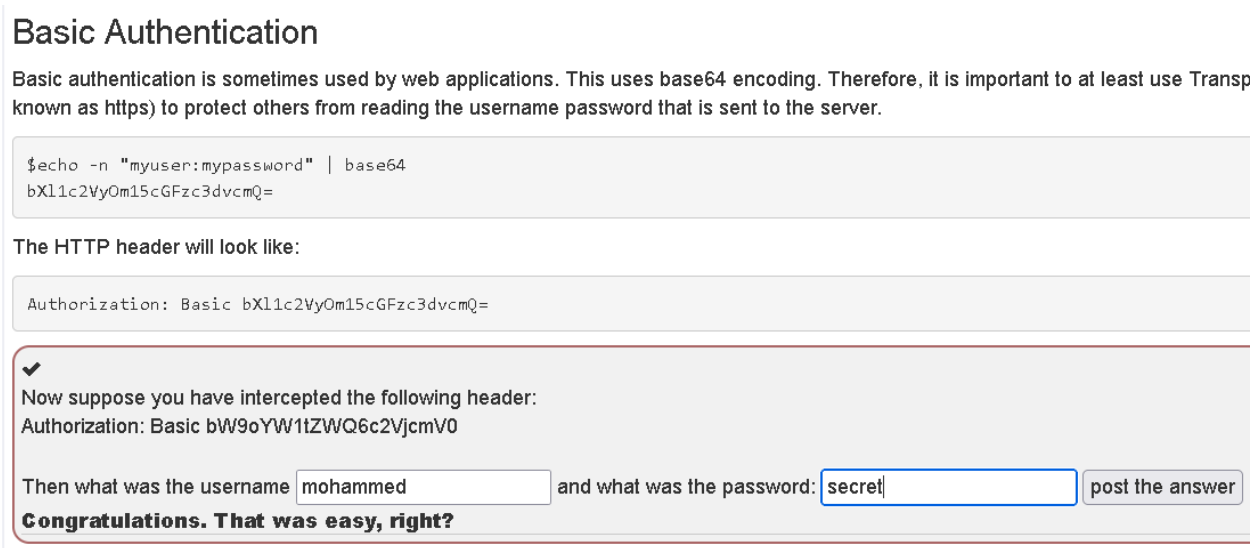
(A2) Cryptographic Failure :

- ### Challenge 1 : Base64 Encoding

Easy challenge give u encoded words with base64
Use cyberchef to decode it



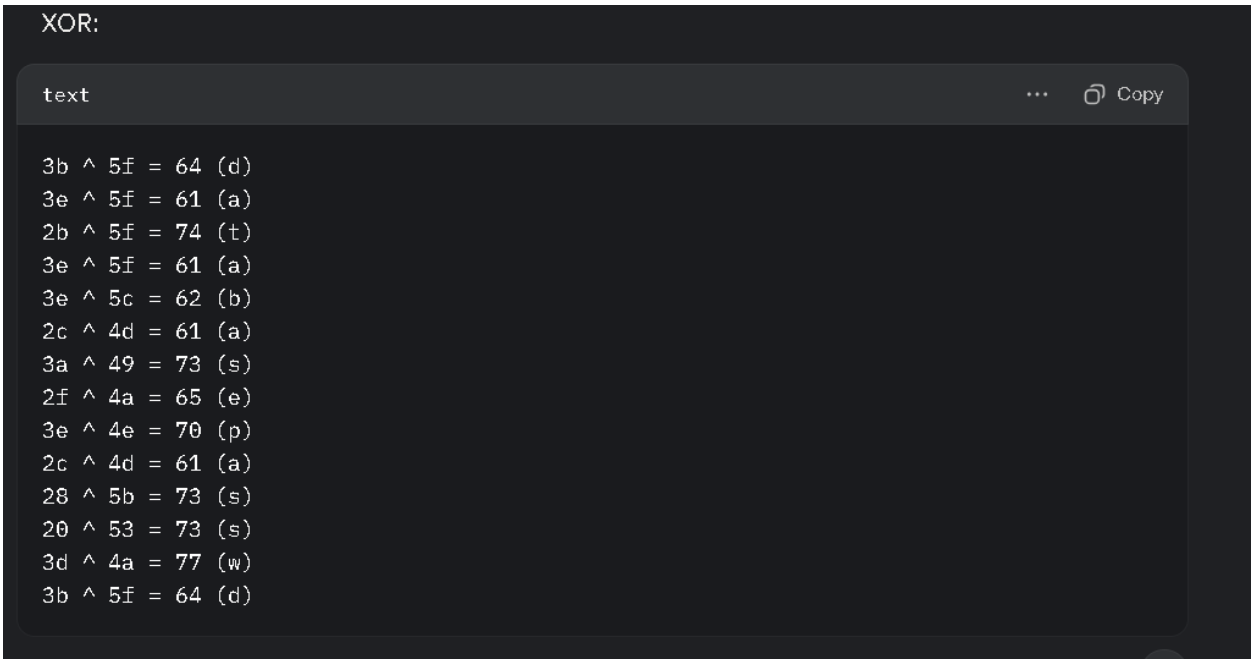
Username is mohammed, password is secret



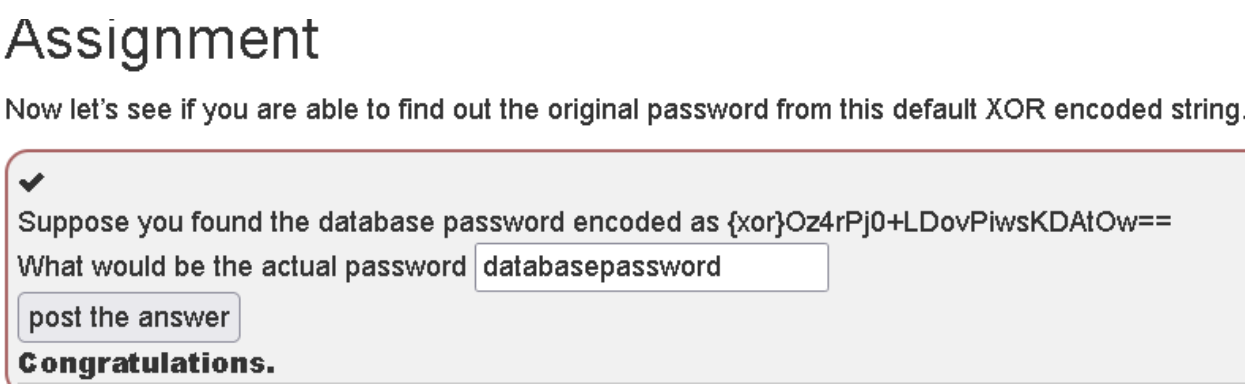
-Challenge 2 : Other encoding :

Encoded password with XOR is : Oz4rPj0+LDovPiwsKDAtOw==

Use grok3 to decode it



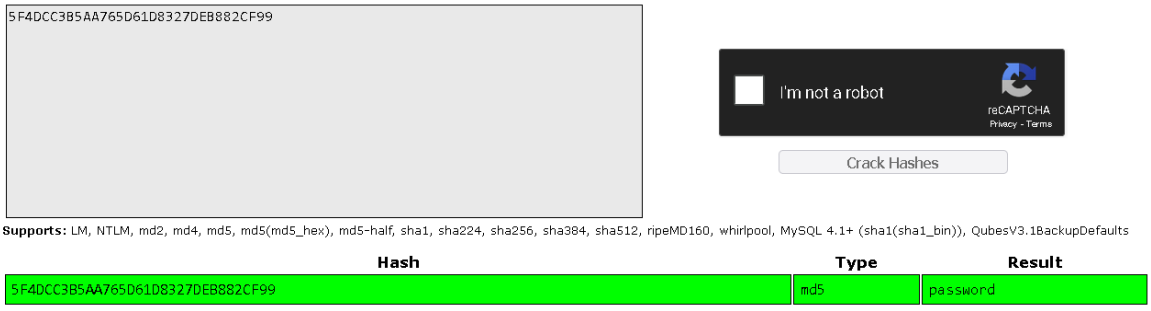
Password is databasepassword



Challenge 3 : Plain Hashing :

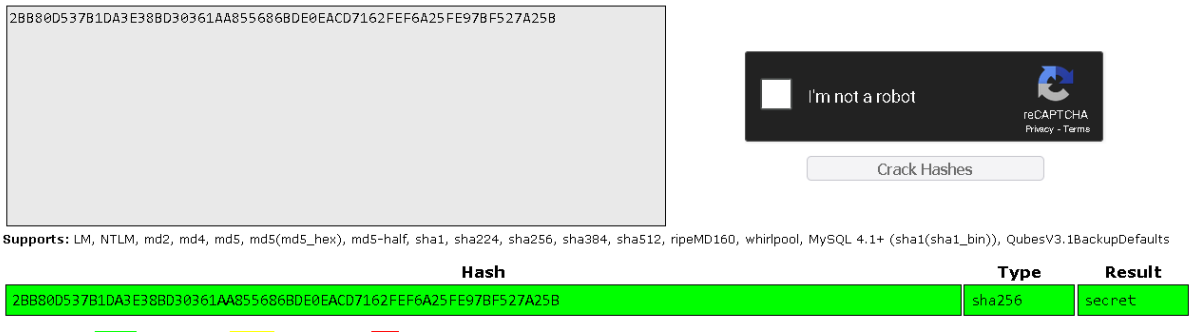
1- Use crack station website to crack hashes

5F4DCC3B5AA765D61D8327DEB882CF99
Is password and type is md5



2- Password is secret and type is sha256

3- 2BB80D537B1DA3E38BD30361AA855686BDE0EACD7162FEF6A25FE97BF527A25B



- **Challenge 3 : RSA Signatures :**

Determine the modulus of the RSA key as a hex string, and calculate a signature for that hex string using the key.

1- Take private key and store in file (c.pem)

2- Run openssl command to extract modulus from private key

```
PS C:\Users\mh855\Downloads> openssl rsa -in c.pem -noout -modulus
Modulus=A2DE545E5A8E68049AB8590494D96570188C56BE78E11DAC2EB9F57285234D537576CECEB242D8B957F471C4833FE55D7E0CFDE3B989DCBC
6A0882DEC8A4A7957F9BF2CB3D3FB140C6461488A5279813D8F205E2A6780F939543C3743392673523B37B1239ED2BCA3B97991E4F6EEB462D641BEF
8225835F40E5B2D1735B9BC26E5AF11029ABC2A1DAEB8FFA031589FD7EECDBB12000FA8978F45F19337EF12620A0C7C56576C18C088317D12AC8501D
FD1BF86D1992FE2CDCD16BD08FD903C9752F7D506285982AF4F5792A80A08FE55E7ADC0BBEB9B25555BAD44EA775C617F31AA8D2DE47F4265D695A27
49BF051FF70561A6841B35E3EBE0825A1B58AE39
PS C:\Users\mh855\Downloads>
```

3- Get the modulus then calculate signature :

-Hashing the input data (e.g., using SHA-256, a common default for RSA signatures).

- Signing the hash with the private key to produce a signature.

4- Convert hex to binary

```
A2DE545E5A8E68049AB8590494D96570188C56BE78E11DAC2EB9F57285234D537576CECEB242D8B957F471C4833FE5
5D7E0CFDE3B989DCBC6A0882DEC8A4A7957F9BF2CB3D3FB140C6461488A5279813D8F205E2A6780F939543C37433B2
364B733B1239ED2BCA3B97991E4F6EEB462D641BEF8225835F40E5B2D1735B9BC26E5AF11029ABC2A1DAEB8FFA0315
89FD7EECCDBB12000FAA5E3D17C64CDFBC49882831F1595DB0630220C5F44AB214077F46FE1B4664BF8B37345AF423
F640F25D4BDF5418A1660ABD3D5E4AA02B0FF9579EB702EBB6B25555BAD44EA775C617F31AA8D2DE47F4265D695A27
49BF051FF70561A6841B35E3EBC08825A1B58AE39
```

abc 511 1 Raw Bytes LF

Output

```
01000001 00110010 01000100 01000101 00110101 00110100 00110101 01000101 00110101 01000001
00111000 01000101 00110110 00111000 00110000 00110100 00111001 01000001 01000010 00111000
00110101 00111001 00110000 00110100 00111001 00110100 01000100 00111001 00110110 00110101
00110111 00110000 00110001 00111000 00111000 01000011 00110101 00110110 01000010 01000101
00110111 00111000 01000101 00110001 00110001 01000100 01000001 01000011 00110010 01000101
01000010 00111001 01000110 00110101 00110111 00110010 00111000 00110101 00110010 00110011
00110100 01000100 00110101 00110011 00110111 00110101 00110111 00110110 01000011 01000101
01000011 01000101 01000010 00110010 00110100 00110010 01000100 00111000 01000010 00111001
00110101 00110111 01000110 00110100 00110111 00110001 01000011 00110100 00111000 00110011
00110011 01000110 01000101 00110101 00110101 01000100 00110111 01000101 00110000 01000011
01000110 01000100 01000101 00110011 01000010 00111001 00111000 00111001 01000100 01000011
01000010 01000011 00110110 01000001 00110000 00111000 00111000 00110010 01000100 01000101
01000011 00111000 01000001 00110100 01000001 00110111 00111001 00110101 00110111 01000110
```

5- The binary in file modulus.bin

6- create an RSA signature of the binary modulus file using the private key. By default, openssl dgst -sign uses SHA-256

```
PS C:\Users\mh855\Downloads> openssl dgst -sha256 -sign c.pem -out signature.bin modulus.bin
PS C:\Users\mh855\Downloads> cat signature.bin
        Å,5Ê0jã¹8æ†ÅE¶7È€SD<mÀøY²6–¿÷:'B¬ïÃMØˆt
                                                C¥›A¾‡D?$¸„ìî¯.è, ´˜C_ÄBqô}N=j¾|ƒ*d(†Ë_¾É#å8Ç»ýÓþÆÉöÔó]a¥H$cpuÖÒX6%\íä7€›
.pÇQN"u¤hûR9íïs©æ¯&Q“8Øz–ðe$ôë wïdp„82ÿR'ŸÊ
                                        Ìã6uÚíÃ¡œÇÍú^¾Ÿ~¤i¹z‡i¶ÞéÜKíø‹³§ß£ÌÇ–x•~³1Iªe0Âak´ª»D®KÏqfÞ>÷í£¸Ì¤¸}
PS C:\Users\mh855\Downloads>
```

7- use this command to convert signature to hex

certutil -encode signature.bin signature.b64

```
PS C:\Users\mh855\Downloads> certutil -encode signature.bin signaturee.b64
Input Length = 256
Output Length = 412
CertUtil: -encode command completed successfully.
PS C:\Users\mh855\Downloads> cat .\signaturee.b64
-----BEGIN CERTIFICATE-----
CcWCNcowauO5OOaGxUW2N8iAU0Q8bcAP+Fm5Npe/9zoAkkKs78NN2Ih0C0Olm0G+
h0Q/JCyE7J3ury7oghQFoLSYBENfFBHEQnH0fU49arx8gypkKIbLX73JI+UAOMe7
/dP+xsn2G+Ef1PNdYaVIJGNwFwZ11tJYNolcBe3kHzeAApsucAfHURJOInUcD6Ro
+1I57e9zqeavJlGTONh6lvBlJPTroBJ372RwhDgy/1KSn8oLHQfM4zZ12+3DoRyd
hMfN+l6+n36kabl6h+8a3uncS+74i7mn36PMx5bXlZ0YfrMxSapl1cJhBmu0qrtE
tkvPcWbePo/37aO4zKS4fQ==
-----END CERTIFICATE-----
PS C:\Users\mh855\Downloads>
```

Finally

The signature is computed by:

1. Converting the modulus hex string to its binary representation (raw bytes).
2. Computing the SHA-256 hash of those bytes.
3. Signing the hash with the private key using RSA.
4. Encoding the resulting 256-byte signature in **Base64** (WebGoat's required format, as per,,,).