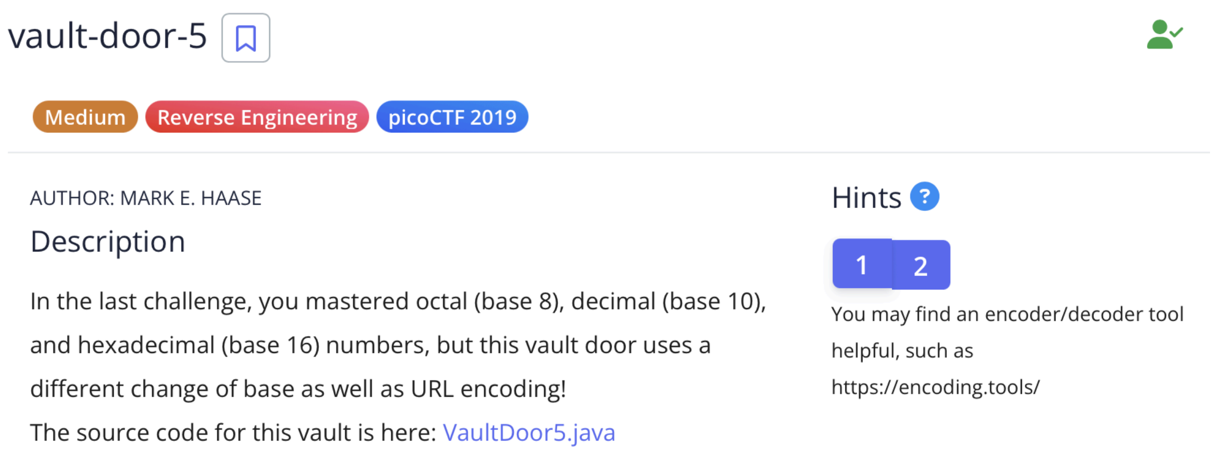


vault-door-5

AUTHOR: MARK E. HAASE

Description

In the last challenge, you mastered octal (base 8), decimal (base 10), and hexadecimal (base 16) numbers, but this vault door uses a different change of base as well as URL encoding!

The source code for this vault is here: VaultDoor5.java

Hints ?

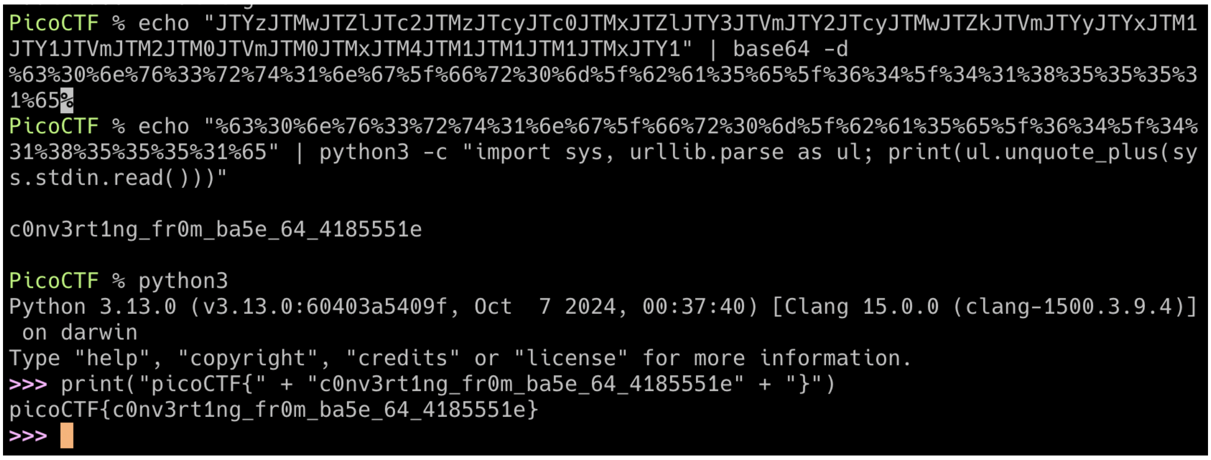
You may find an encoder/decoder tool helpful, such as https://encoding.tools/

Isi filenya :

```java
import java.net.URLDecoder;
import java.util.*;

class VaultDoor5 {
    public static void main(String args[]) {
        VaultDoor5 vaultDoor = new VaultDoor5();
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter vault password: ");
        String userInput = scanner.next();
    String input =
userInput.substring("picoCTF{".length(),userInput.length()-1);
    if (vaultDoor.checkPassword(input)) {
        System.out.println("Access granted.");
    } else {
        System.out.println("Access denied!");
        }
    }

    // Minion #7781 used base 8 and base 16, but this is base 64, which
is
    // like... eight times stronger, right? Riiigghtt? Well that's what
my twin
    // brother Minion #2415 says, anyway.
    //
    // -Minion #2414
    public String base64Encode(byte[] input) {
        return Base64.getEncoder().encodeToString(input);
    }

    // URL encoding is meant for web pages, so any double agent spies
who steal
    // our source code will think this is a web site or something,
defintely not
    // vault door! Oh wait, should I have not said that in a source code
    // comment?
    //
    // -Minion #2415
    public String urlEncode(byte[] input) {
        StringBuffer buf = new StringBuffer();
        for (int i=0; i<input.length; i++) {
            buf.append(String.format("%%%2x", input[i]));
        }
        return buf.toString();
    }

    public boolean checkPassword(String password) {
        String urlEncoded = urlEncode(password.getBytes());
        String base64Encoded = base64Encode(urlEncoded.getBytes());
        String expected = "JTYzJTMwJTZlJTc2JTMzJTcyJTc0JTMxJTZlJTY3JTVm"
                        + "JTY2JTcyJTMwJTZkJTVmJTYyJTYxJTM1JTY1JTVmJTM2"
                        + "JTM0JTVmJTM0JTMxJTM4JTM1JTM1JTM1JTMxJTY1";
        return base64Encoded.equals(expected);
    }
}
```

Jika kita lihat, disitu terdapat fungsi cek password yang didalamnya ada sebuah string expected untuk verifikasi apakah password sama dengan string expected.

Kita bisa melakukan decode dengan base64 lalu decode URL seperti berikut untuk mendapatkan flag



```
PicoCTF % echo "JTYzJTMwJTZlJTc2JTMzJTcyJTc0JTMxJTZlJTY3JTVmJTY2JTcyJTMwJTZkJTVmJTYyJTYxJTM1
JTY1JTVmJTM2JTM0JTVmJTM0JTMxJTM4JTM1JTM1JTM1JTMxJTY1" | base64 -d
%63%30%6e%76%33%72%74%31%6e%67%5f%66%72%30%6d%5f%62%61%35%65%5f%36%34%5f%34%31%38%35%35%35%3
1%65%
PicoCTF % echo "%63%30%6e%76%33%72%74%31%6e%67%5f%66%72%30%6d%5f%62%61%35%65%5f%36%34%5f%34%
31%38%35%35%35%31%65" | python3 -c "import sys, urllib.parse as ul; print(ul.unquote_plus(sy
s.stdin.read()))"

c0nv3rt1ng_fr0m_ba5e_64_4185551e

PicoCTF % python3
Python 3.13.0 (v3.13.0:60403a5409f, Oct  7 2024, 00:37:40) [Clang 15.0.0 (clang-1500.3.9.4)]
 on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> print("picoCTF{" + "c0nv3rt1ng_fr0m_ba5e_64_4185551e" + "}")
picoCTF{c0nv3rt1ng_fr0m_ba5e_64_4185551e}
>>> 
```

Flag : picoCTF{c0nv3rt1ng_fr0m_ba5e_64_4185551e}