Enumeration and Exploitation

Python 2 (Easy)

This challenge evaluates the contestant's ability understand and analyze vulnerabilities in a compiled Python program. The uncompyle program can be used to convert the compiled program back into Python code. The result of running uncompyle can be seen below:

```
import sys

def main():
    if len(sys.argv) != 2:
        print 'Invalid args'
        return
    password = sys.argv[1]
    counter = 0
    vals = list('tfzbwlyzljylawhzzdvyk')
    if len(password) != len(vals):
        print 'incorrect'
        return
    while counter < len(password):
        x = ord(password[counter]) + 7
        if x > ord('z'):
            x -= 26
        if chr(x) != vals[counter]:
            print 'incorrect'
            return
        counter += 1

    print 'correct'


if __name__ == '__main__':
    main()
```

An analysis of this code reveals that the password is comprised of each character in "vals" shifted 7 positions down the alphabet, with a wraparound. This is an implementation of a shift cipher. To solve the challenge, shift each character in "vals" up 7 positions up the alphabet.

Question	Answer
What is a secret key that will pass validation?	`mysupersecretpassword`