Module 6.3: Password integrity-entropy

- Entropy is a measure of disorder.
- In our case, high entropy is good, as it makes it harder for hackers to guess your password.
- OSINT (open source intelligence) is a different matter-these are based on human things like pet names, birthdays, etc.

How good are your passwords?

Create a password entropy checker:

In your terminal, enter the following:

```
sudo nano entropy_calc.py
```

paste the text below into the window, then ctrl-o and ctrl-x:

```
# entropy_calc.py
import math,sys
def entropy(password):
    L = len(password)
    N = 0
    if any(c.islower() for c in password): N += 26
    if any(c.isupper() for c in password): N += 26
    if any(c.isdigit() for c in password): N += 10
    if any(not c.isalnum() for c in password): N += 33   # rough symbol set
    return 0.0 if N==0 else L * math.log2(N)
pw = sys.argv[1]
print(f"{pw}: length={len(pw)} entropy≈{entropy(pw):.1f} bits")
```

How to use:

```
python3 entropy_calc.py 'Password123'
python3 entropy_calc.py 'correcthorsebatterystaple'
```

Look at the results

Try with the passwords you generated before

create a password checker program:

```
sudo nano zxcvbn_demo.py
```

paste this into the window, then ctrl-o and ctrl-x:

```
#!/usr/bin/env python3
# zxcvbn_demo.py
# Usage:
python3 zxcvbn_demo.py password1 "another password"
# Or:
echo -e "pw1\npw2" | python3 zxcvbn_demo.py
import sys
try:
    from zxcvbn import zxcvbn
except Exception as e:
    print("Error importing zxcvbn. Install via: pip3 install zxcvbn
(or `npm i -g zxcvbn` and use node fallback).")
    raise
def analyze(pw):
    r = zxcvbn(pw)
    print("="*60)
    print("Password:", pw)
    print(" score (0-4):", r.get("score"))
    print(" guesses (est):", r.get("guesses"))
    print(" guesses_log10:", r.get("guesses_log10"))
    fb = r.get("feedback") or {}
    if fb.get("warning"):
        print(" warning:", fb.get("warning"))
    if fb.get("suggestions"):
        print(" suggestions:")
        for s in fb.get("suggestions"):
            print("   -", s)
    # show match breakdown (if present)
    matches = r.get("sequence", [])
    if matches:
        print(" pattern matches (top few):")
        for m in matches[:5]:
            print("  -", m.get("pattern"), m.get("token"), f"(i:{m.get('i')}, j:{m.get('j')})")
    print()
def main():
```

```
    inputs = []
    if not sys.stdin.isatty():
        inputs += [l.strip() for l in sys.stdin if l.strip()]
    if len(sys.argv) > 1:
        inputs += sys.argv[1:]
    if not inputs:
        print("Usage: echo 'pass1\npass2' | python3 zxcvbn_demo.py
OR python3 zxcvbn_demo.py pass1 pass2")
        sys.exit(1)
    for pw in inputs:
        analyze(pw)
if __name__ == "__main__":
    main()
```

Try it out:

```
python3 zxcvbn_demo.py Password123 "correct horse battery staple"
```

or

```
echo -e "Password123\nTr0ub4dor&3\ncorrecthorsebatterystaple" | python3 zxcvbn_demo.py
```

```
scripts to use: download the file below
```

```
unzip
sudo chmod +x setup_commands.sh
./setup_commands.sh
python3 entropy_demo.py
```

Download raspi_cyber_labs.zip (application/zip, 5.8 KB)

Migrated from Apple Server Wiki. Original page ID: S9X8T7

Revision #3
Created 2026-09-29 16:00:01 UTC by Admin

Updated 2026-09-29 16:49:24 UTC by Admin