OWASP Top 10 - In the Wild

> Common pitfalls using hypothetical scenarios and pseudo-code snippets.

1. Broken Access Control

Insecure direct object reference, incorrect authorization checks.

```
curl vuln.target/invoice?id=1234
# Here is the invoice document for user John Doe: ...

curl vuln.target/management-panel?admin=true
# Hello, admin!
```

2. Cryptographic Failures

Cryptographically insecure randomness, weak signing key, using MD5 for password hashes, unsalted hash, hard-coded secret.

```
from string import lower, hexdigits
from random import choices
from flask import Flask

app = Flask(__name__)
app.secret_key = ''.join(lower(choices(hexdigits, k=4)))

@app.post('/login')
def login(username, password):
    if username == 'admin' and md5(password) == '345301f693deda839f7ed280a4b30763':
        return 200, "Login Succeeded"
    return 304, "Login Failed"
```

3. Injection

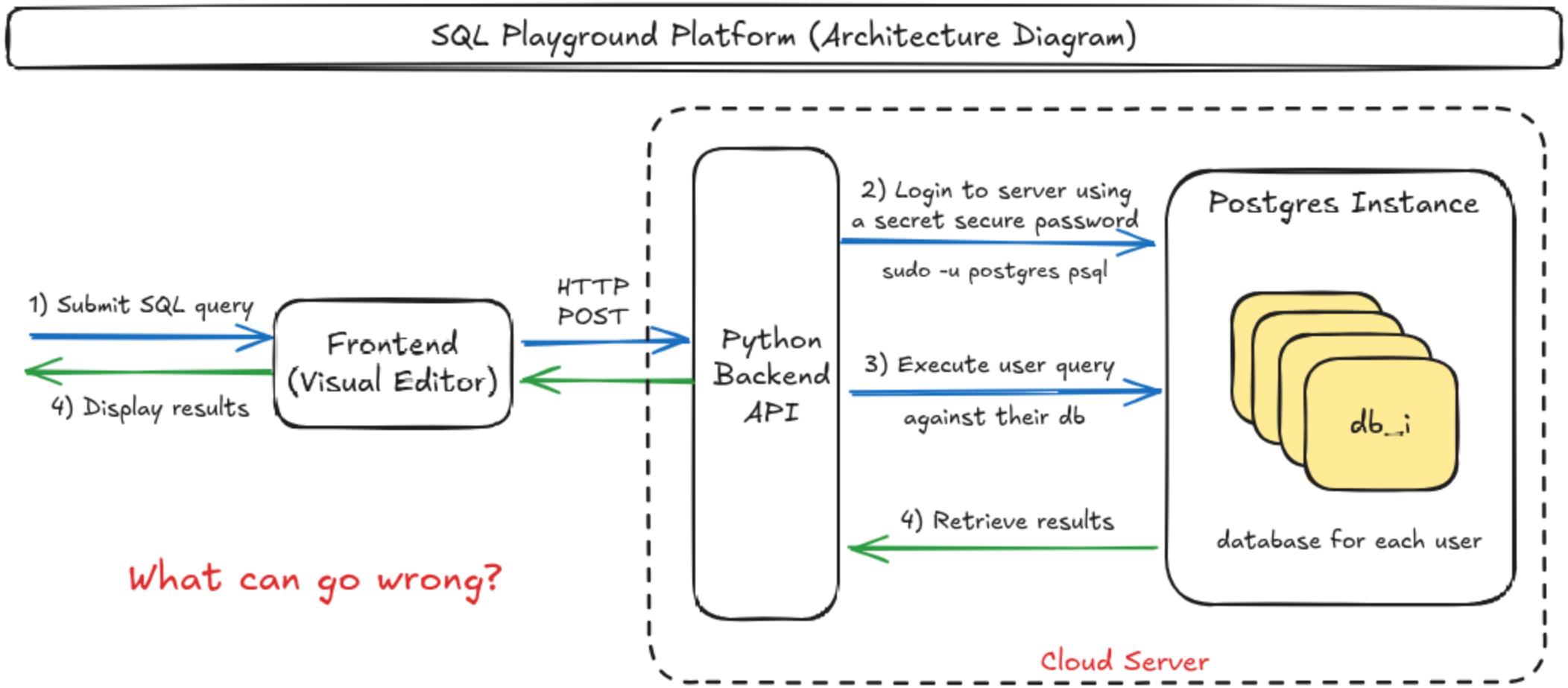
SQL Injection, OS command injection, HTTP CRLF injection

```
def getUser(user_id):
    res = cur.execute("SELECT * FROM Users WHERE id = " + user_id)
    return res.fetchall()

def ping(address):
    return os.system("ping " + address + " -c 1")
```

```
HTTP/1.1 200 OK
Content-Type: text/html
Content-Length: 36
Some-Header: [INPUT]

<body>This is the normal body</body>
```

4. Insecure Design

Initial architecture diagram for a SQL training platform. What can go wrong?



5. Security Misconfiguration

Non-strict permissions, default credentials, HTTPS server allowing HTTP requests.

```
$ ls -la /etc/shadow
-rw-rw-rw- 1 ubuntu ubuntu 1294 Jan  1 00:00 /etc/shadow

$ cat docker-compose.yaml
  ...
  image: postgres
  environment:
      POSTGRES_PASSWORD: example
  ...

$ cat nginx.conf
server {
    listen   80, 443;
    ...
}
```

6. Vulnerable and Outdated Components

A web application may rely on packages that have publicly-disclosed vulnerabilities. Same problem with os-level packages.

```
$ npm audit
...
webpack-dev-middleware <=5.3.3
Severity: high
Path traversal in webpack-dev-middleware - https://github.com/advisories/...
...
21 vulnerabilities (10 low, 4 moderate, 7 high)

$ sshd -V
OpenSSH_4.3p2, OpenSSL 0.9.8e 23 Feb 2007

# CVE-2008-5161 - Pre-authentication X11 forwarding vulnerability
# CVE-2008-3259 - X11 cookie theft vulnerability
# CVE-2007-4752 - CRC32 compensation attack detector vulnerability
```

7. Identification and Authentication Failures

Allowing weak passwords, not blocking automated attacks

```
$ curl -X POST http://vuln.target/register \
  -d '{"username": "someone", "password": "pass123"}
Registered successfuly

$ hydra -l someone -P wordlist.txt http-post://vuln.target/login
...
login: someone   password: pass123
...
```

Password reuse across services, not using 2FA, long-lived session tokens.

```
$ curl -X POST https://another.target/login \
  -d '{"username": "someone", "password": "pass123"}
Login succeeded.

$ curl https://another.target/session-token | jwt-decode
{
  "sub": "1234567890",
  "name": "John Doe",
  "iat": 1678886400,
  "exp": 1688890000,
  "role": "user"
}
```

8. Software and Data Integrity Failures

Compromised artifact repo lead to mismatching hash of dependency.

```
$ wget https://vuln.target/artifact-repo/dep-v1.0.0.pkg
$ sha1sum dep-v1.0.0.pkg
6d24c593a0a56cec4f361d18b5d22f55a9c0a4ed dep-v1.0.0.pkg

$ curl https://dependecy-official.com/download/dep-v1.0.0.pkg.sha1
488f987b777f03a1293cba89d39f8c79c179061e dep-v1.0.0.pkg
```

9. Security Logging and Monitoring Failure

Failed authentication attempted not properly logged, or logs were manipulated.

```
user@ubuntu$ su root
Password:
su: authentication failure

...

root@ubuntu$ grep "authentication failure" /var/log/auth.log
root@ubuntu$ # What?
```

10. Server-Side Request Forgery

Retriever service leaked data from an internal resource, and the firewall allowed outbound traffic.

```
$ curl https://vuln.target/retrieve/?url=http://example.com | grep title
<title>Example domain</title>

$ curl https://vuln.target/retrieve/?url=http://localhost/admin | grep title
<title>Admin Panel</title>
```