Tutorial: Squat Scanner

Foundations of Cybersecurity

Giacomo Benedetti

Institute for Applied Mathematics and Information Technologies

National Research Council of Italy

giacomo.benedetti@ge.imati.cnr.it

University of Pavia – Department of Electrical, Computer and Biomedical Engineering

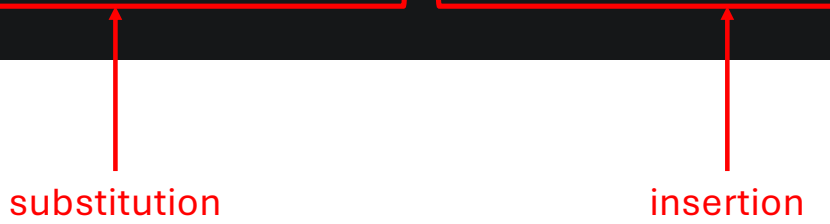
Package Names Similarity: Is it Common?

- This example contains a small fuzzer for testing the presence of typosquatted packages.
- Idea:
  - generating package names
  - try to retrieve them to check whether they exist
  - if yes, inspect collected output.

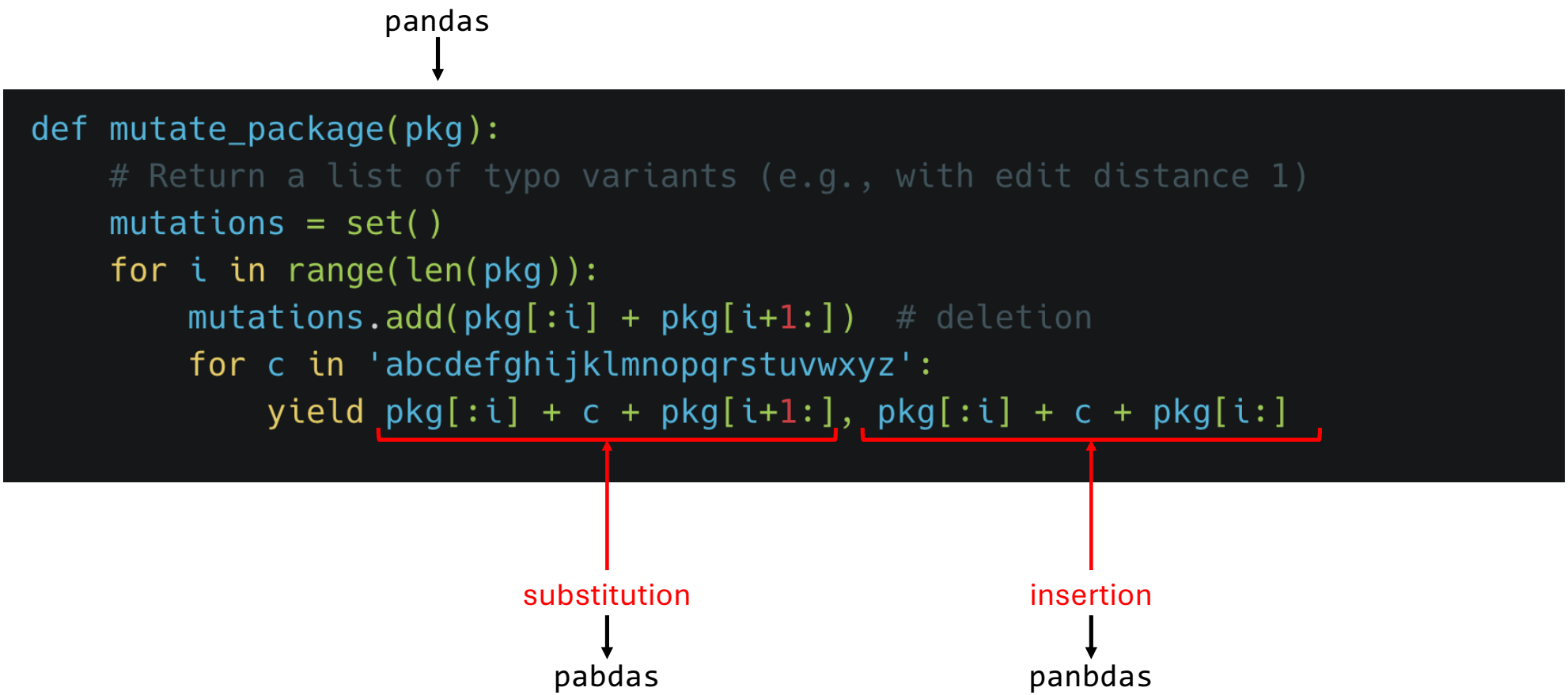
```python
from rapidfuzz.distance import Levenshtein
import subprocess

def mutate_package(pkg):
    # Return a list of typo variants (e.g., with edit distance 1)
    mutations = set()
    for i in range(len(pkg)):
        mutations.add(pkg[:i] + pkg[i+1:])  # deletion
        for c in 'abcdefghijklmnopqrstuvwxyz':
            yield pkg[:i] + c + pkg[i+1:], pkg[:i] + c + pkg[i:]


def try_install(package_name):
    try:
        result = subprocess.run(
            ["pip", "install", package_name, "--no-deps", "--no-cache-dir"],
            stdout=subprocess.PIPE,
            stderr=subprocess.PIPE,
            timeout=15  # prevent hanging
        )
        success = result.returncode == 0
        output = result.stdout.decode("utf-8")
        error = result.stderr.decode("utf-8")
        return success, output, error
    except subprocess.TimeoutExpired:
        return False, "", "Timeout"


if __name__ == "__main__":
    for subst, inser in mutate_package('pandas'):
        success, out, err = try_install(subst)
        if success:
            print(subst)
        success, out, err = try_install(inser)
        if success:
            print(inser)
```

Package Names Similarity: Is it Common?

```python
def mutate_package(pkg):
    # Return a list of typo variants (e.g., with edit distance 1)
    mutations = set()
    for i in range(len(pkg)):
        mutations.add(pkg[:i] + pkg[i+1:])  # deletion
        for c in 'abcdefghijklmnopqrstuvwxyz':
            yield pkg[:i] + c + pkg[i+1:], pkg[:i] + c + pkg[i:]
```



Package Names Similarity: Is it Common?



Package Names Similarity: Is it Common?

pandas →

```python
from rapidfuzz.distance import Levenshtein
import subprocess

def mutate_package(pkg):
    # Return a list of typo variants (e.g., with edit distance 1)
    mutations = set()
    for i in range(len(pkg)):
        mutations.add(pkg[:i] + pkg[i+1:])  # deletion
        for c in 'abcdefghijklmnopqrstuvwxyz':
            yield pkg[:i] + c + pkg[i+1:], pkg[:i] + c + pkg[i:]


def try_install(package_name):
    try:
        result = subprocess.run(
            ["pip", "install", package_name, "--no-deps", "--no-cache-dir"],
            stdout=subprocess.PIPE,
            stderr=subprocess.PIPE,
            timeout=15  # prevent hanging
        )
        success = result.returncode == 0
        output = result.stdout.decode("utf-8")
        error = result.stderr.decode("utf-8")
        return success, output, error
    except subprocess.TimeoutExpired:
        return False, "", "Timeout"


if __name__ == "__main__":
    for subst, inser in mutate_package('pandas'):
        success, out, err = try_install(subst)
        if success:
            print(subst)
        success, out, err = try_install(inser)
        if success:
            print(inser)
```

→

```
(venv) [~/Desktop/Fuzzing/pypi]$ python3 test.py
bandas
candas
fandas
fpandas
ipandas
kpandas
mpandas
pandas
spandas
upandas
wandas
xpandas
pandas
paandas
psndas
pundas
pyndas
paandas
pamdas
pandas
panndas
panadas
pandas
panndas
panvas
pandas
pandis
pandaa
pandag
pandan
pandaq
pandas
pandavs
pandaz
```