

Packet Sniffing with Python

Objectives:

At the end of this episode, I will be able to:

1. Capture and save network packets using Python
-

Snippets

main.py

```
from scapy import all as s
from PyInquirer import prompt
from netifaces import interfaces

def main():
    questions = [
        dict(
            type="checkbox",
            name="interfaces",
            message="Which of the following interfaces do you want to sniff?",
            choices=[{"name": k} for k in interfaces()],
        ),
        dict(type="input", name="timeout", message="How long do you want to sniff?"),
        dict(
            type="confirm", name="save", message="Do you want to save these to files?"
        ),
    ]
    answers = prompt(questions)
    results = s.sniff(iface=answers["interfaces"], timeout=int(answers["timeout"]))
    if answers["save"]:
        s.wrpcap("capture.pcap", results)

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

pyproject.toml

```
[tool.poetry]
name = "packet-sniffer"
version = "0.1.0"
description = ""
authors = ["<Your name>"]

[tool.poetry.dependencies]
python = "^3.7"
scapy = "^2.4"
netifaces = "^0.10.9"
PyInquirer = "^1.0"

[tool.poetry.dev-dependencies]
bpython = "^0.19.0"

[build-system]
requires = ["poetry>=0.12"]
build-backend = "poetry.masonry.api"
```

External Resources:

During this episode, you can reference the following external resources for supplementary tools and information:

- (S) Scapy Documentation
- (S) PyInquirer Documentation
- (S) Netifaces Documentation