47 lines - 2 Removals

```
import argparse
import logging
import pathlib

import pipeline_framework
import pipeline_steps


def main(pipeline_yaml_path: pathlib.Path, run_name: str, smoke_test: bool):
    run_base_path = pathlib.Path(f"../intermediate/run-{run_name}")
    run_base_path.mkdir(exist_ok=True)

    logging_format = (
        "%(asctime)s | %(relativeCreated)d | %(levelname)s (%(name)s): %(message)s"
    )

    logging.basicConfig(
        level=logging.DEBUG,
        handlers=[
            logging.FileHandler(str(run_base_path / "execution.log"), mode="a"),
            logging.StreamHandler(),
        ],
        format=logging_format,
    )

    logger = logging.getLogger("run_pipeline")

    pipeline = pipeline_framework.Pipeline(
        pipeline_steps.pipeline_steps,
        pipeline_yaml_path,
        logger,
        run_base_path,
        smoke_test,
    )

    pipeline.run()


if __name__ == "__main__":
    parser = argparse.ArgumentParser("run_pipeline")
    parser.add_argument("pipeline_yaml", type=str)
    parser.add_argument("run_name", type=str)
    parser.add_argument("--smoke-test", action="store_true")

    args = parser.parse_args()
    main(pathlib.Path(args.pipeline_yaml), args.run_name, args.smoke_test)
```

49 lines + 4 Additions

```
import argparse
import logging
import pathlib

import pipeline_framework
import pipeline_steps


def main(pipeline_yaml_path: pathlib.Path, run_name: str, smoke_test: bool, malware:bool):
    run_base_path = pathlib.Path(f"../intermediate/run-{run_name}")
    run_base_path.mkdir(exist_ok=True)

    logging_format = (
        "%(asctime)s | %(relativeCreated)d | %(levelname)s (%(name)s): %(message)s"
    )

    logging.basicConfig(
        level=logging.DEBUG,
        handlers=[
            logging.FileHandler(str(run_base_path / "execution.log"), mode="a"),
            logging.StreamHandler(),
        ],
        format=logging_format,
    )

    logger = logging.getLogger("run_pipeline")

    pipeline = pipeline_framework.Pipeline(
        pipeline_steps.pipeline_steps,
        pipeline_yaml_path,
        logger,
        run_base_path,
        smoke_test,
        malware
    )

    pipeline.run()


if __name__ == "__main__":
    parser = argparse.ArgumentParser("run_pipeline")
    parser.add_argument("pipeline_yaml", type=str)
    parser.add_argument("run_name", type=str)
    parser.add_argument("--smoke-test", action="store_true")
    parser.add_argument("--malware", action="store_true")

    args = parser.parse_args()
    main(pathlib.Path(args.pipeline_yaml), args.run_name, args.smoke_test, args.malware)
```