# GothX: a generator of customizable, legitimate and malicious IoT network traffic

Authors: **POISSON Manuel**
CARNIER Rodrigo, FUKUDA Kensuke

Full paper: https://inria.hal.science/hal-04629350

*August 13, 2024, CSET'24*

# Introduction

NII   AMOSSYS   COLLEGE DOCTORAL BRETAGNE

**Internet of Things (IoT)
increasing usage**

**MQTT** and **Kafka**
IoT data
collection/processing

## Defend IoT

Increase of attacks against IoT [1]

⇒ Development of Intrusion Detection Systems (IDS) performing anomaly detection using machine learning[2]

⇒ Need of datasets for training models

[1] Kolias et al. **"DDoS in the IoT: Mirai and Other Botnets"**. In: *Computer* 50.7 (2017), pp. 80−84
[2] Lahesoo et al. **"SIURU: A Framework for Machine Learning Based Anomaly Detection in IoT Network Traffic"**. In: AINTEC '23. Dec. 2023, pp. 87−95

Implementation and architecture
○○○

Use-cases
○○○○○○

evaluation: scalability and reproductibility
○

Conclusion
○

# Introduction

## Expected properties of datasets

- Mix legitimate and malicious traffic
- Supervised training and validation ⇒ labels
- Robustness of IDS ⇒ diversity
  - Detection of different attacks
  - Avoid alerts when legitimate traffic varies

## Get desired datasets

- Use publicly available dataset → single snapshot
- Generate own dataset:
  - Develop own traffic generator → requires time and expertise
  - Use existing traffic generator → difficult to find and not very flexible

# Our contribution

## GothX traffic generator

- IoT network traffic: **MQTT** and **Kafka**
- Generates **labeled dataset**
- **Open-source and modifiable**

## Automatically executing a customizable scenario

- **Legitimate actions**
- Attacker complete **kill chain from initial compromission to DDoS**
- **Customizable**: study impact on IDS of various parameters (eXplainable AI)

## Ready-to-use new datasets

- Provide datasets generated using GothX

Implementation and architecture
●○○

Use-cases
○○○○○○

evaluation: scalability and reproducibility
○

Conclusion
○

# Related works

## GothX: a fork of Gotham[3]

Gotham uses **GNS3°** to **emulate virtual networks**

| Features | Gotham | GothX |
|---|---|---|
| Open-source | ✓ | ✓ |
| Legitimate + malicious traffic | ✓ | ✓ |
| Virtualization (Docker + VM) | ✓ | ✓ |
| Automatic network initialization | ✓ | ✓ |
| Reproducible results | ✓ | ✓ |
| Labeled data | | ✓ |
| Customizable node behavior | | ✓ |
| MQTT service | ✓ | ✓ |
| MQTT-Kafka service | | ✓ |
| Accompanying ready-made datasets | | ✓ |

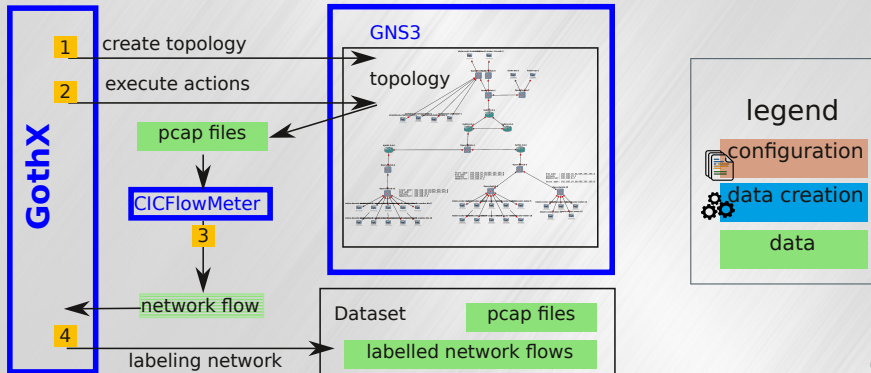**GothX extends Gotham's features and add new ones**

[3] Saez-de-Camara et al. **"Gotham Testbed: A Reproducible IoT Testbed for Security Experiments and Dataset Generation".** In: *IEEE Transactions on Dependable and Secure Computing* PP (Jan. 2023), pp. 1–18

# GothX architecture and workflow

NII  AMOSSYS  COLLEGE DOCTORAL BRETAGNE

## GothX's workflow
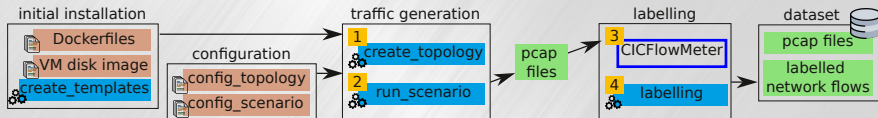


## GothX's interaction with other tools

Implementation and architecture
○○●
Use-cases
○○○○○○
evaluation: scalability and reproducibility
○
Conclusion
○

# Customization

NII AMOSSYS

COLLEGE
DOCTORAL
BRETAGNE

## Benefits of customization

- Settings combinations ⇒ **diversity of the network traffic**
- Analyze the efficiency of anomaly detection when
  **legitimate traffic varies but the attack is the same, or vice-versa**
- Variation of settings independently ⇒
  study the **impact of a specific parameter** on a machine learning model (XAI)

### Customizable topology and scenario parameters

| Legitimate traffic | Malicious traffic |
|---|---|
| Sensors count | Parameters of attack tools |
| Messages rate* (periodic/random) | Intensity of DDoS attack (e.g. payload size) |
| (In)activity duration* | % of compromised sensors |
| Which data, from a dataset of real sensors, is sent* | Sleep time between attack steps |
| Traffic volume (MQTT/Kafka)* | |

*customizable for each sensor independently

Implementation and architecture
○○○

Use-cases
●○○○○○○

evaluation: scalability and reproductibility
○

Conclusion
○

# 2 case examples

## Case 1: MQTTSet reproduction

- Multiple MQTT behavior patterns
- 5 types of denial of service (DoS)

## Case 2: Full, multi-step, attack scenario

- Legitimate MQTT and Kafka traffic
- Attacker spread in the network
  (different techniques to take control of multiple nodes)
- DDoS

# Case 1: MQTTSet reproduction

## The MQTTSet dataset [4]

**Legitimate traffic**
10 sensors publishing periodically or randomly

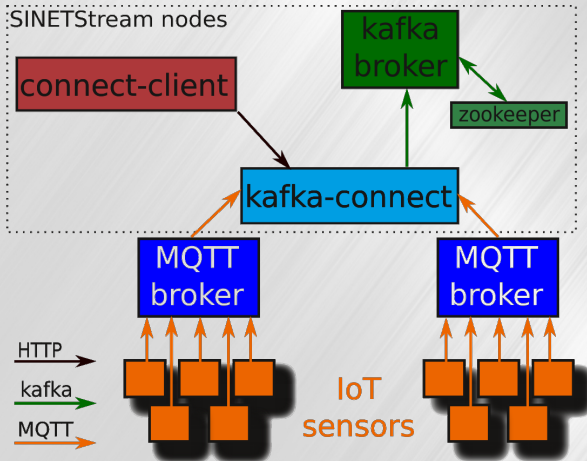**5 types of denial of service in MQTTSet**

- 1 file with legitimate traffic only, 1 file per attack type
- Synthetic legitimate traffic (no real broker) ⇒ **impossible to visualize DoS impact**

## Our contribution

Reproduction of MQTTSet: similar characteristics of legitimate and attack traffic.
GothX is more realistic: **mix legitimate/malicious traffic**

---

[4]Vaccari et al. **"MQTTset, a New Dataset for Machine Learning Techniques on MQTT".** en. In: *Sensors* 20.22 (Jan. 2020). Number: 22 Publisher: Multidisciplinary Digital Publishing Institute, p. 6578
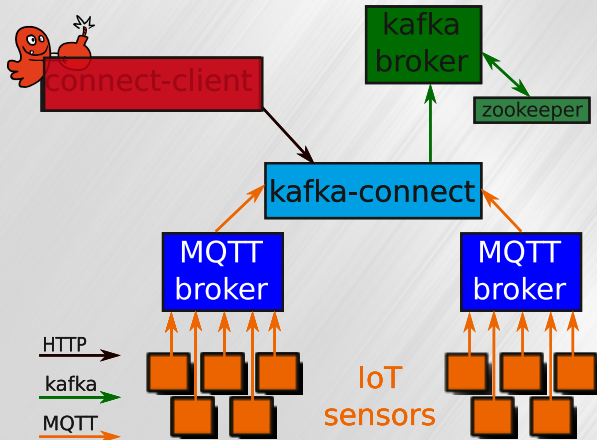
Implementation and architecture
○○○

Use-cases
○○●○○○

evaluation: scalability and reproducibility
○

Conclusion
○

# Case 2: Full scenario: topology

NII    AMOSSYS    COLLEGE DOCTORAL BRETAGNE



## Assumptions

- Some IoT sensors with SSH open ports
- *kafka-connect*
  - version 7.3.1 (December 2022)
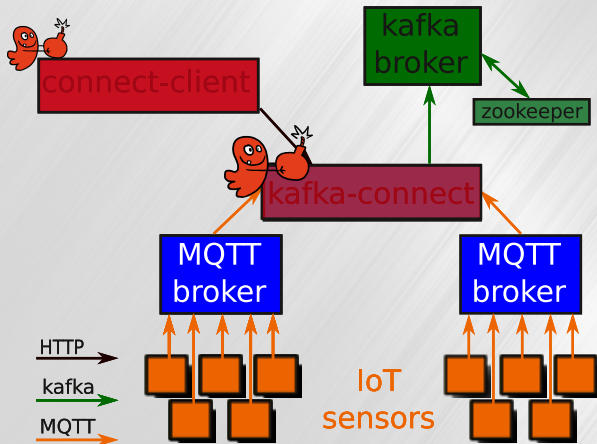  - `enableUnsafe Serialization=true`
  - ⇒ CVE-2023-25194

# Case 2: Full scenario: attack steps
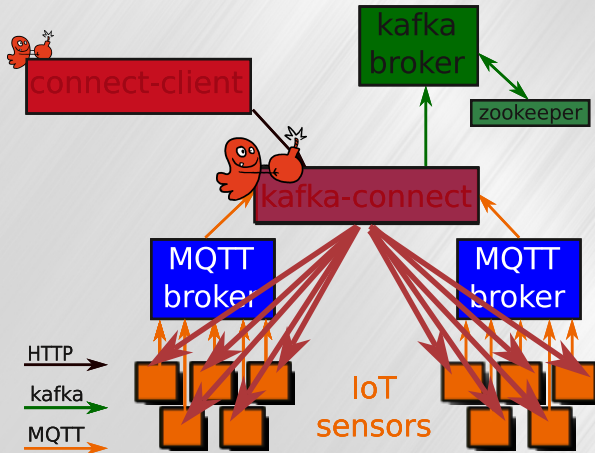


1. Attacker controls *connect-client*

**Internal attack**: device *connect-client* sent legitimate requests. It starts to be malicious.
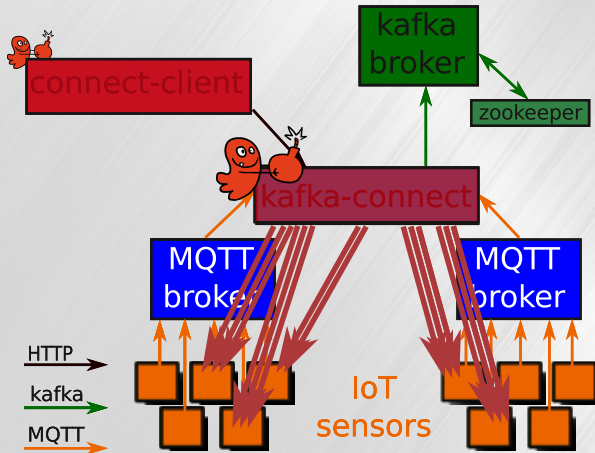
# Case 2: Full scenario: attack steps



1. Attacker controls *connect-client*
2. Exploit CVE-2023-25194 on *kafka-connect* ⇒ RCE

Implementation and architecture  ○○○
Use-cases  ○○○●○○
evaluation: scalability and reproducibility  ○
Conclusion  ○

# Case 2: Full scenario: attack steps



1. Attacker controls *connect-client*
2. Exploit CVE-2023-25194 on *kafka-connect* ⇒ RCE
3. Discover of devices responding to SSH

Implementation and architecture
○○○

**Use-cases**
○○○●○○

evaluation: scalability and reproducibility
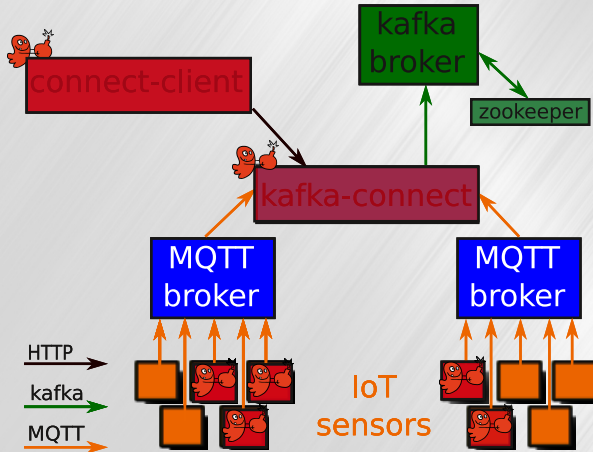○

Conclusion
○

# Case 2: Full scenario: attack steps



1. Attacker controls *connect-client*
2. Exploit CVE-2023-25194 on *kafka-connect* $\Rightarrow$ RCE
3. Discover of devices responding to SSH
4. Bruteforce SSH credentials

Implementation and architecture
○○○

Use-cases
○○○●○○

evaluation: scalability and reproducibility
○

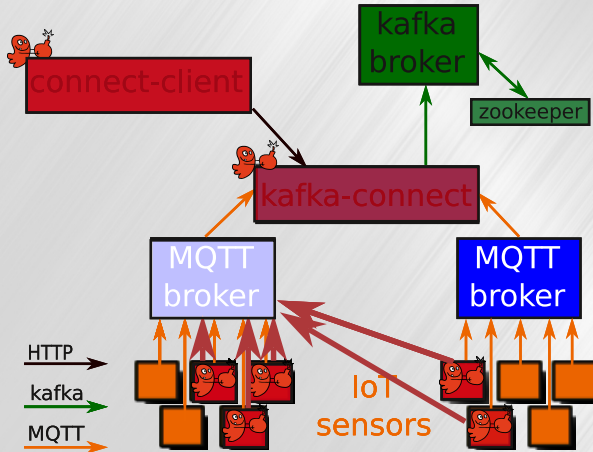Conclusion
○

# Case 2: Full scenario: attack steps



1. Attacker controls *connect-client*
2. Exploit CVE-2023-25194 on *kafka-connect* ⇒ RCE
3. Discover of devices responding to SSH
4. Bruteforce SSH credentials
5. Transfer payload (via SSH)

Implementation and architecture
○○○

Use-cases
○○○●○○

evaluation: scalability and reproducibility
○

Conclusion
○

# Case 2: Full scenario: attack steps



1. Attacker controls *connect-client*
2. Exploit CVE-2023-25194 on *kafka-connect* ⇒ RCE
3. Discover of devices responding to SSH
4. Bruteforce SSH credentials
5. Transfer payload (via SSH)
6. Simultaneous payload execution ⇒ DDoS
7. Target (MQTT Broker) crash

Implementation and architecture
○○○

**Use-cases**
○○○○●○

evaluation: scalability and reproductibility
○

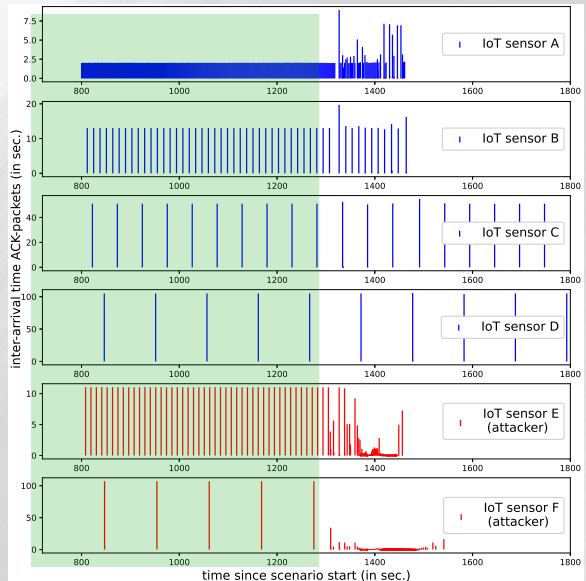Conclusion
○

# Case 2: Full scenario: DDoS analysis

DDoS impact:
Inter-arrival time
of ACK-packets
during scenario
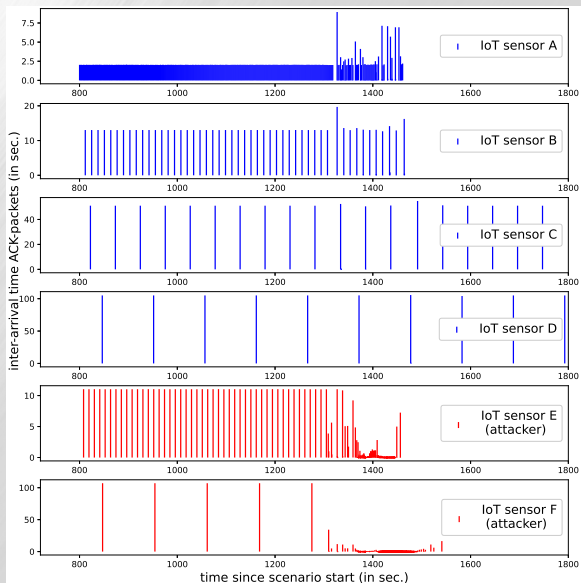
## Case 2: Full scenario: DDoS analysis

DDoS impact:
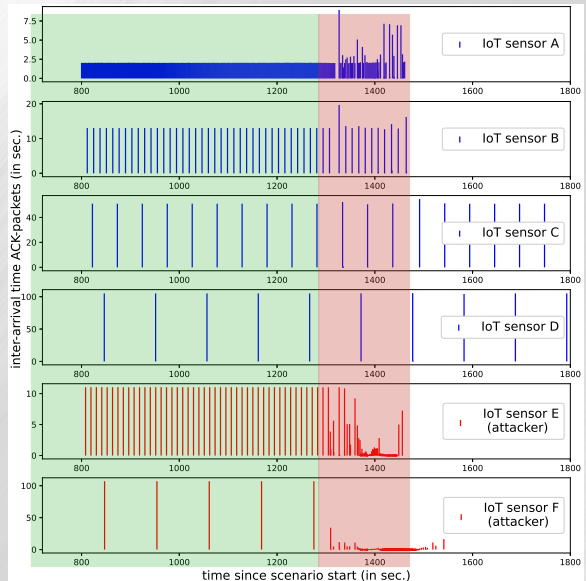Inter-arrival time
of ACK-packets
during scenario

**Before DDoS**

# Case 2: Full scenario: DDoS analysis

DDoS impact:
Inter-arrival time
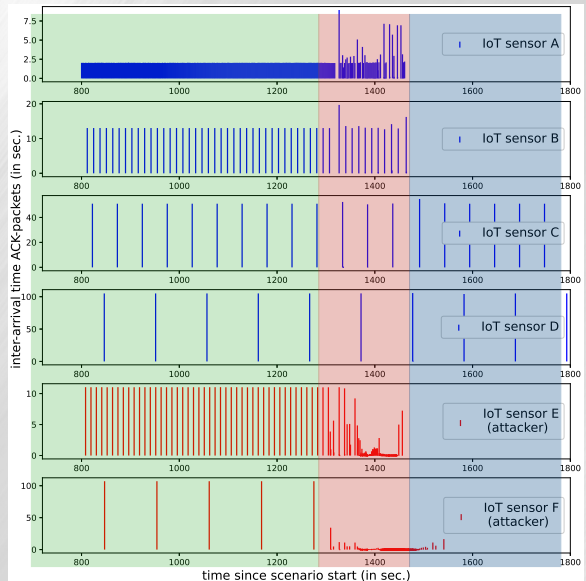of ACK-packets
during scenario

**Before DDoS**
**During DDoS**

Implementation and architecture
○○○

Use-cases
○○○○●○

evaluation: scalability and reproducibility
○

Conclusion
○

# Case 2: Full scenario: DDoS analysis

DDoS impact:
Inter-arrival time
of ACK-packets
during scenario

**Before DDoS**
**During DDoS**
**After DDoS**

# Case 2: Full scenario: provided dataset

## pcap files and details

- **All generated traffic captured**
  3 pcap files mixing legitimate and malicious actions (like in real world network traffic)
- Text file **describing configuration** and **command line** executed

## TCP flows labelling

1. Automatic flows' features extraction with *CICFlowMeter*[5]
2. **Attack steps labelling adaptative to customization** with provided custom script

## Usage of variations of this dataset

Atsuya et al. **"Dynamic Fixed-point Values in eBPF: a Case for Fully In-kernel Anomaly Detection".** In: AINTEC '24. Aug. 2024, p. 8

---

[5]https://github.com/GintsEngelen/CICFlowMeter

# GothX scalability and replication

## Scalability

Definition: more IoT devices running simultaneously

- **Hardware ressources**:
  - RAM: 20GB for 450 sensors
  - CPU: depends on DDoS intensity
- **Realism**:
  - do not simply duplicate sensors with exactly the same behavior
  - use customization to send different data for each sensor
- **Execution time**:
  - data generation: fully customizable, depends on scenario duration
  - topology deployment: 4 VM and 498 Docker containers $\rightarrow \approx$ 26 minutes

## Replication

✓ public source code and documentation
✓ GothX's installation and usage on different computers using documentation
✓ Executions with the same configuration $\Rightarrow$ Generation of similar datasets

# Conclusion

Delivery of the **traffic generator GothX**
- Open-source[6]
- Customizable

Producing **IoT network datasets**
- Labeled
- Legitimate and malicious traffic

Delivery of **2 datasets**
1. MQTTSet reproduced
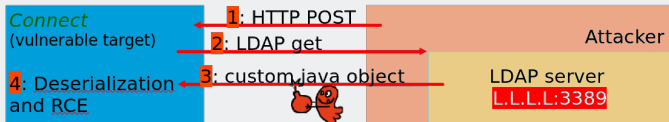2. New dataset on $\approx$ 14h from our customizable multi−steps scenario

**Customizable full (attack) scenario**
- Legitimate MQTT and Kafka messages
- Exploitation of recent, highly critical vulnerability (CVE-2023-25194)
- Ports scan and credentials bruteforce
- DDoS

[6] Software and datasets available at https://github.com/fukuda-lab/GothX

# Details on the attack

NII   AMOSSYS   COLLEGE DOCTORAL BRETAGNE

## CVE-2023-25194



**Connect**
(vulnerable target)

**1**: HTTP POST

**2**: LDAP get

**4**: Deserialization and RCE

**3**: custom java object

Attacker

LDAP server
L.L.L.L:3389

**CVE-2023-25194 exploitation**

## Type of DDoS

SlowITe[7]: exhaust the number of simultaneous connections to the broker
Using tool mqttsa

---

[7]Ivan Vaccari, Maurizio Aiello, and Enrico Cambiaso. **"SlowITe, a Novel Denial of Service Attack Affecting MQTT".** In: *Sensors* 20 (May 2020), p. 2932

# Attacks in MQTTSet

NII   AMOSSYS   COLLEGE DOCTORAL BRETAGNE

| Attack type | Tool |
|---|---|
| Flood DoS | MQTT-malaria |
| MQTT publish flood (CVE-2018-1684) | IoT-Flock |
| SlowITe | SlowTT |
| Maleformed data | MQTTSA |
| Authentication bruteforce | MQTTSA |

[7]Ivan Vaccari, Maurizio Aiello, and Enrico Cambiaso.  **"SlowITe, a Novel Denial of Service Attack Affecting MQTT".** In: *Sensors* 20 (May 2020), p. 2932

Implementation and architecture
○○○

Use-cases
○○○○○○

evaluation: scalability and reproducibility
○

Conclusion
●

# Example of a configuration

NII

AMOSSYS

COLLEGE
DOCTORAL
BRETAGNE

```python
iot_devices = {
    "iotsim-domotic-monitor-bis-1": {
        "SLEEP_TIME": "10",
        "SLEEP_TIME_SD": "0",
        "DATASET_COLUMNS": "1,2",
        "MQTT_BROKER_ADDR": "broker.neigh.lab",
        "ACTIVE_TIME": "120",
        "INACTIVE_TIME": "60",
    },
    "iotsim-cooler-motor-1": {
        "SLEEP_TIME": "1",
        "DATASET_COLUMNS": "0,1",
        "MQTT_BROKER_ADDR": "broker.steel.lab",
    },
    "iotsim-predictive-maintenance-60": {
        "SLEEP_TIME": "65",
        "SLEEP_TIME_SD": "1",
        "DATASET_COLUMNS": "11,1,9",
        "MQTT_BROKER_ADDR": "secure.mqtt.lab",
        "TLS": True,
    }}
kafka_topic = "kafka-topic"
mqtt_topics_to_connect = {
    "iotsim-mqtt-broker-1.6-1": [
        "iotsim-domotic-monitor-bis-1"],
    "iotsim-mqtt-broker-1.6-auth-1": [
```

```python
DDoS_only = False

proportion_devices_launching_ddos = 20 / 100
shuffled_iot_names = list(iot_devices.keys())
random.shuffle(shuffled_iot_names)
nodes_with_ssh = list(...)


w_time_legitimate_only_before_attack = 60 * 60 * 24
w_time_cve_exploitation_openrevshell = 60 * 60 * 1
w_time_openrevshell_toolstransfert = 60 * 60 * 1
w_time_toolstransfert_nmap = 60 * 60 * 2
w_time_nmap_hydra = 60 * 60 * 2
w_time_hydra_mqttsa_scp = 60 * 30
w_time_scp_coordinated_launch = 60 * 10
w_time_end_ddos_to_end_scenario = 60 * 60


nmap_args = "192.168.18-20.10-150 --max-rate 0.7 -p 22"
hydra_args = "-f -L u.txt -P p.txt -t 2"
mqttsa_args = "-fc 100 -fcsize 10 -sc 2400"
target_mqtt_broker_ip = "192.168.2.1"
```