Date 23/01/2023

Défi 01 "Automatisation de la vérification cybersécurité de systèmes cyber physiques

Groupe de travail défi 01

Philippe Massonet, Coordinateur Scientifique CETIC
Guillaume NGuyen, UNamur
Martin Vivian, UCLouvain
Sébastien Dupont, Guillaume Ginis, CETIC

digital wallonia 4.cyberwal

Wallonie recherche SPW

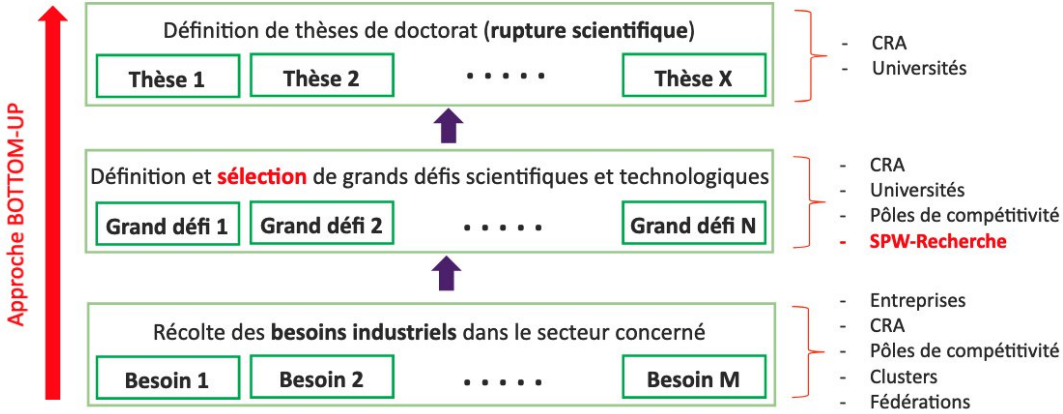
https://cyberwal.be
https://cyberexcellence.be

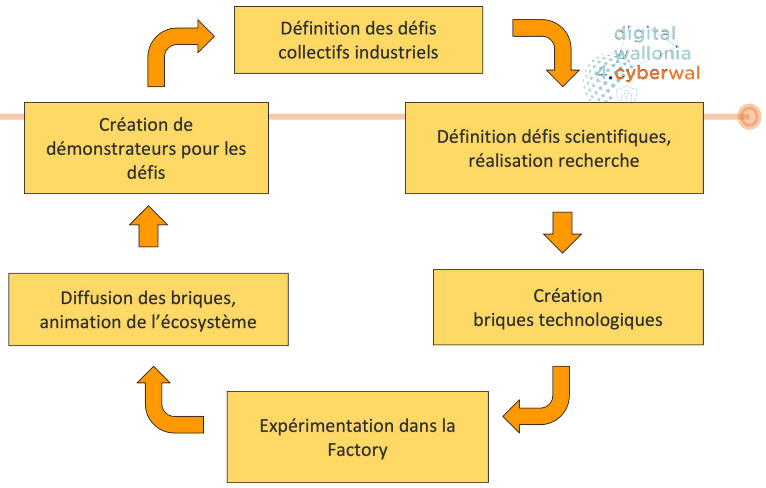
Agenda

10:00-10:05	Rappel projet CyberExcellence, expérimentation dans la factory, et défi 01	Philippe Massonet
10:05-10:20	Présentation des problèmes de recherches liés au défi: •Identification of Cyber Physical System (CPS) & Orchestration of fuzzing testing •Improvements for stateful fuzzing, •Tests generation for cyber security	Guillaume Nguyen (Unamur), Martin Vivian (UCLouvain), Philippe Massonet
10:20-10:40	Présentation de l'étude de cas proposée pour expérimenter les algorithmes de génération de tests de cybersécurité (factory)	Guillaume Ginis, Sébastien Dupont
10:40-11:00	Discussion sur des vulnérabilités, défauts de conception à tester	Tous

Projet CyberExcellence et Défis Collectifs Industriels

- Projet CyberExcellence
  - Projet de recherche en cybersécurité, 01/01/2022, 18,9 millions de budget)
  - Partenaires : 5 universités + 2 CRA
  - Recherche fondamentale mais **au bénéfice du tissu industriel**: réponds aux besoins des entreprises/administrations
- Défi Collectif Industriel
  - Récolte des besoins industriels dans le secteur concerné
  - Identification des défis Collectif Industrie
- Factory
  - Production de briques technologiques





WP

WP1 : Rendre les systèmes résilients aux cyberattaques : phase de conception.

WP2 : Détection, Réponse, Réaction : Phase Dynamique

WP3 : RGPD et Open data : sécurité à la conception

WP4 : La protection et le partage des données au cœur des préoccupations

WP5 : Laboratoires d'expérimentation, de validation, et d'entraînement

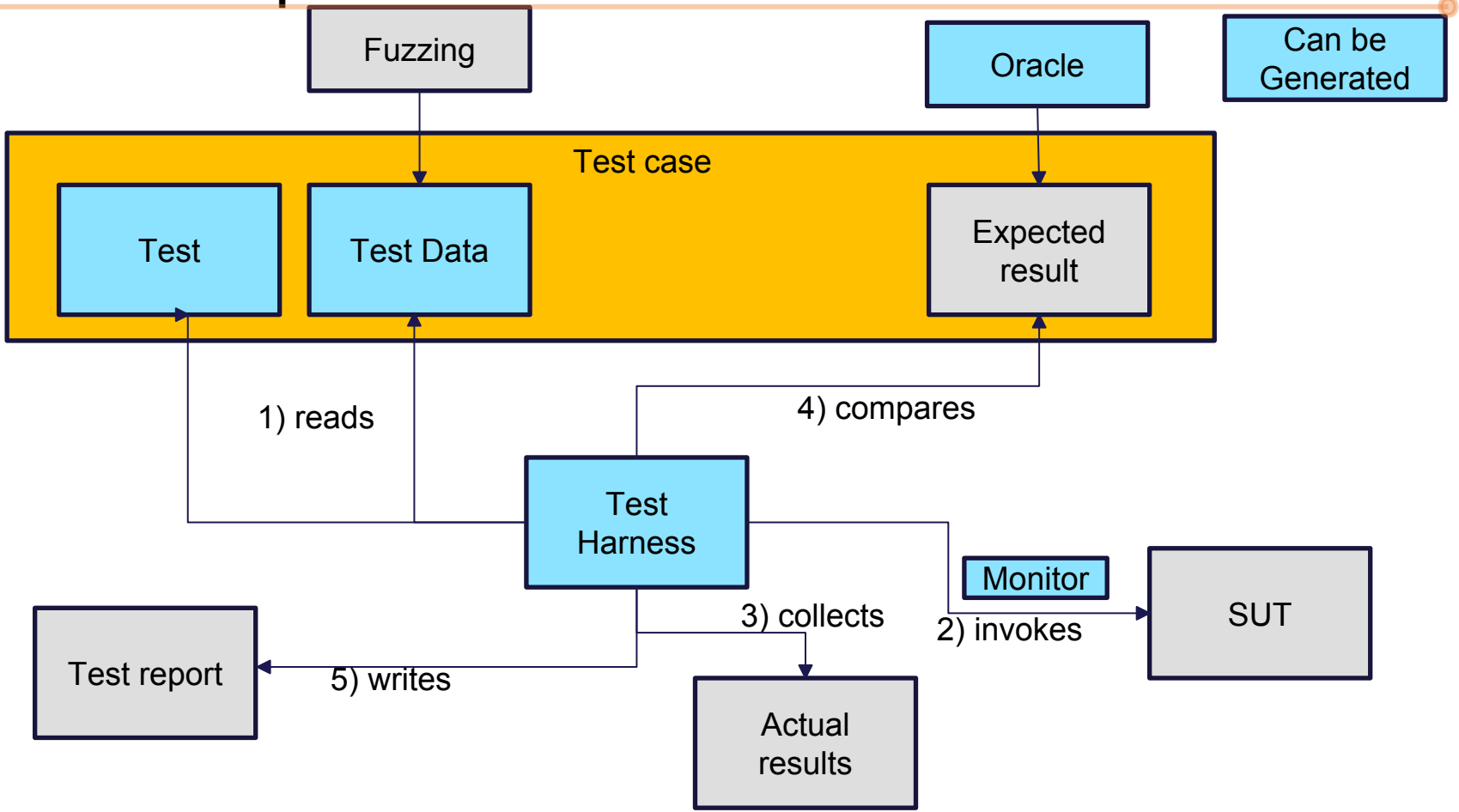
WP6 : Factory et grands défis

Défi 01 : Automatisation de la vérification cybersécurité de systèmes cyber physiques - consultation – Commentaires reçus

- **Résumé du défi:**
  - Tests de pénétration: processus encore très manuel, requiert des experts en cybersécurité
  - Ambition: automatiser (en partie) la création des tests de pénétration pour rendre les tests de pénétration plus accessible par les entreprises (PME, grandes entreprises)

- **Challenges de recherche**:
  - Génération automatique des tests de cybersécurité fonctionnels (architecture de sécurité), utilisation de différentes techniques de génération (à comparer) pour les tests de pénétration:
    - Techniques de fuzzing
    - Génération de tests par mutation génétique
    - Génération de tests à partir de modèles
    - ...
  - Automatisation partielle sous forme d'assistance du processus de création et de la définition des tests de pénétration.
  - « Risk-based testing » pour trouver un meilleur ROI (vulnérabilités/attaques trouvées/budget de test)

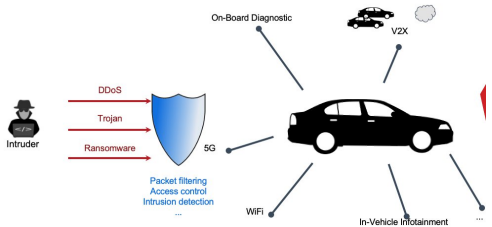
“Test Harness pattern” - Génération

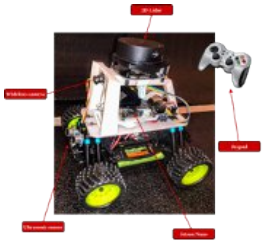


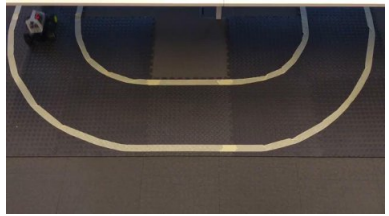
Défi 01 : Automatisation de la vérification cybersécurité de systèmes cyber physiques - consultation – Problèmes de recherche

- Problèmes de recherche
  - UNamur : fuzzing guidé par des algorithmes génétiques (Prof. Xavier Devroey, 1 chercheur)
  - UCLouvain : fuzzing – découverte de protocoles de communication par apprentissage, analyse de malware (Prof. Axel Legay, 2 chercheurs)
  - CETIC : génération de jeux de tests, proposition d'une étude de cas (Philippe Massonet, 2 chercheurs pour 1ETP)

- Expérimentation dans la factory
  - Déploiement d'un système à tester dans une sandbox de la factory
  - Etude de cas :
    - véhicule connecté (V2X)+ centre de gestion de traffic (Cloud)
    - Introduction de vulnérabilités
  - Challenge : tests générés découvrent-ils les vulnérabilités/malware







Rover - Donkey Car

Identification of Cyber Physical System (CPS) & Orchestration of fuzzing testing (UNamur)

Plan d'action:

1. Enquête (clients)
2. Trouver les limitations des tests
3. Enquête (fournisseurs)
4. Trouver les limitations des tests
5. Quand et comment faire du fuzzing?
6. Quels fuzzers utiliser?
7. Lister les fuzzers manquants
8. Lister les tests de fuzzer manquants (performance)

Identification of Cyber Physical System (CPS) & Orchestration of fuzzing testing (UNamur)

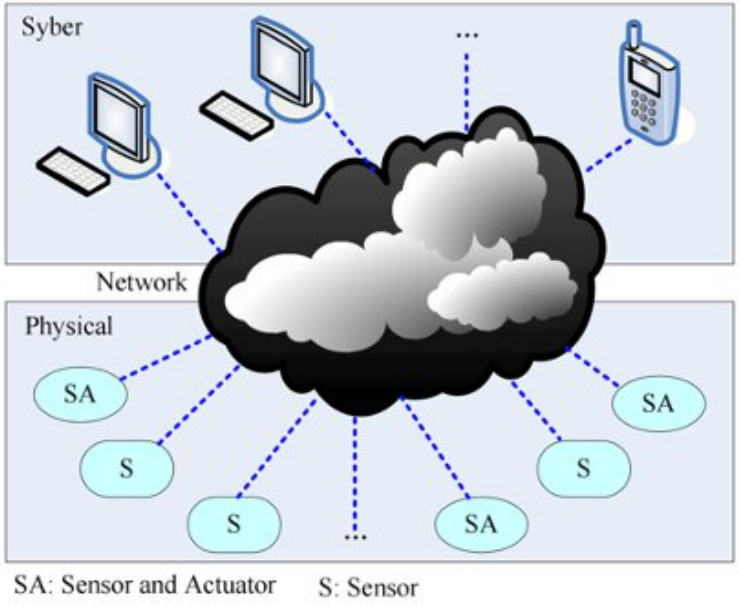


Figure 1. Diagrammatic layout for CPSs

- Approche très large;
- Touche énormément de sous-systèmes;
- Ensemble de composantes testables individuellement et en groupe (physique, cyber, réseau);
- Pas spécifique à une industrie (médecine, usine, automobile, etc.);
- Réel nécessité d'un Framework de test

Source: Shi et al., A survey of CPS (2011)

Identification of Cyber Physical System (CPS) & Orchestration of fuzzing testing (UNamur)

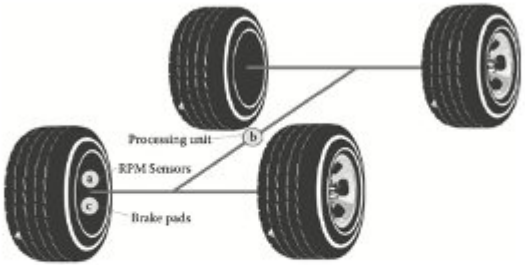


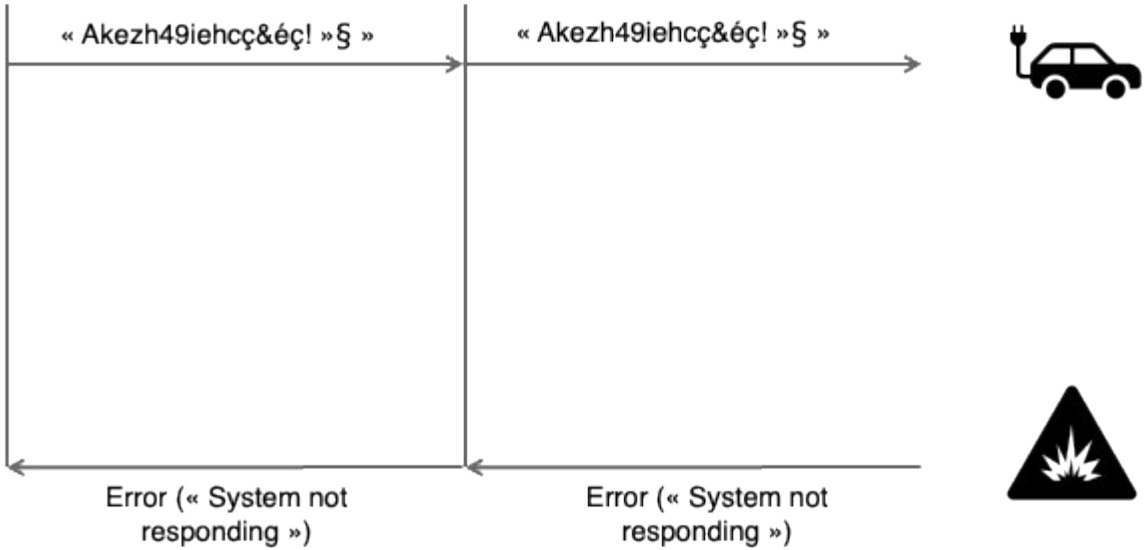
Figure 1.1 CPS example: antilock brake system. When the driver holds down the brake pedal, the CPS in the car pumps the brakes automatically. (a) Sensors near the tires collect information about the rate of rotation of each tire and send these data back to the (b) processing unit. This information is then used to determine the system's status. A series of commands to carry out the response is sent to the (c) brake pads, which activate or deactivate appropriately.

- Approche très réduite;
- Limite la définition de l'implémentation;
- Peu de composantes à tester;
- Composantes facilement identifiables;
- Très spécifique;

Source: Fei hu et al., CPS integrated computing and engineering design (2014)

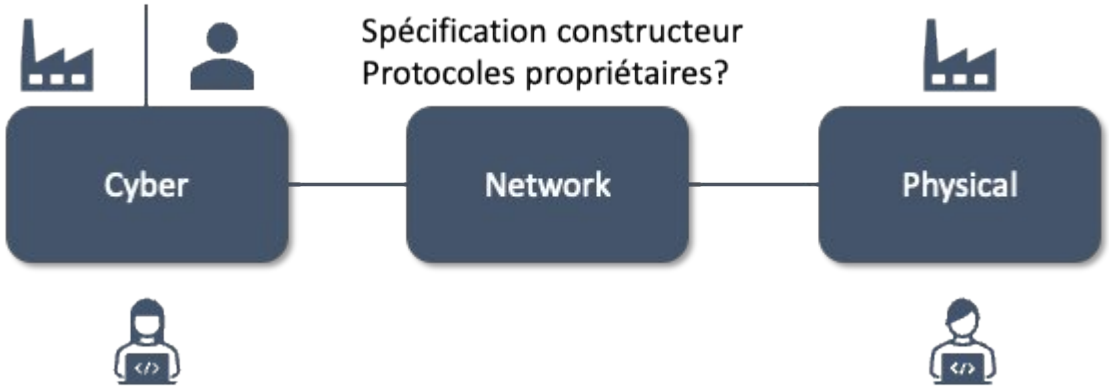
Identification of Cyber Physical System (CPS) & Orchestration of fuzzing testing (UNamur)

Fuzzing : Envoyer des inputs aléatoires ou pseudo aléatoires vers un système pour voir comment il y répond. Le système peut casser, cesser de fonctionner, se comporter étrangement, etc.



Identification of Cyber Physical System (CPS) & Orchestration of fuzzing testing (UNamur)

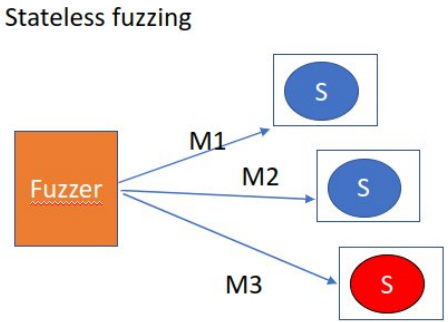
But:
En utilisant la définition d'un CPS, lister les CPS utilisés en Belgique/Europe/monde afin d'en établir le paysage.
Cette liste sera agrémentée des exigences/défis en termes de maintenance et de test.

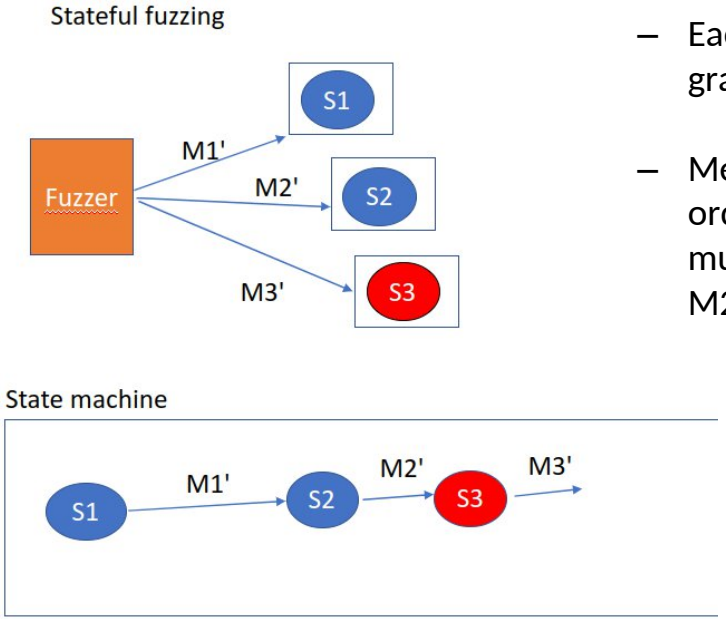


Où faire du fuzzing?
- Trouver le code
- Côté client
- Côté fournisseur

Improvements for stateful fuzzing (UCLouvain)

Stateless and Stateful





- Each message has their grammar
- Messages must follow an order. In this example, we must send M1' before M2' to reach S3

Solutions for black-box stateful systems

Some approaches

Random fuzzer : *Radamsa* (https://gitlab.com/akihe/radamsa) …

Mannualy config fuzzer : *Boofuzz* (https://github.com/jtpereyda/boofuzz)

Mitm fuzzing : *ProxyFuzzer*

Problem

These approaches are inefficient, or it takes time to be configured,needs to reverse the protocol to configure the fuzzer...

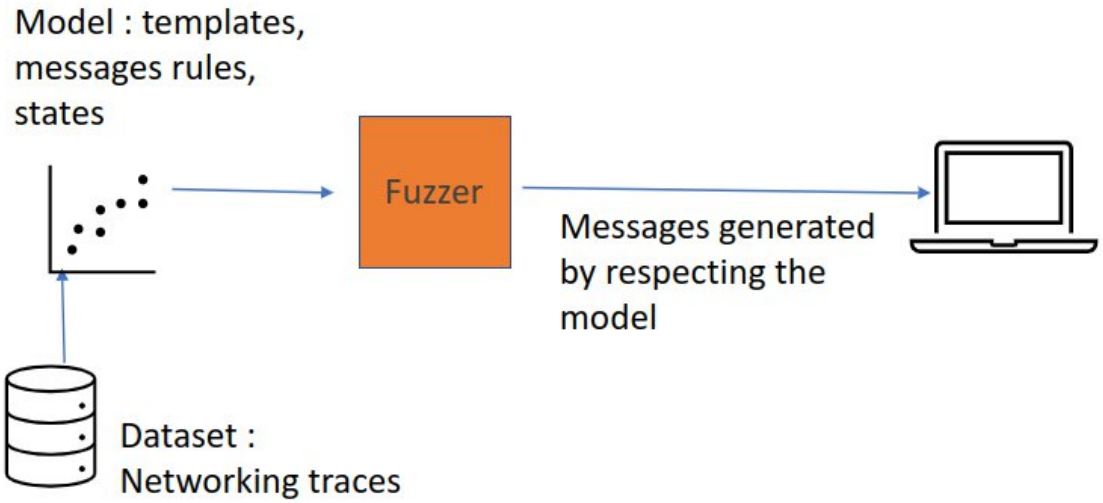
Solution for black-box stateful systems

Research solutions

Some fuzzer in the research field can understand the system based on the network traffic, and find a model to get efficient fuzzing and a few configuration.

Example : *Pulsar* (https://github.com/hgascon/pulsar)
Autofuzz (https://autofuzz.sourceforge.net/)

Example Pulsar



Find a template for the message : constant part in the frames, rules (incremented value)
Ex : START_ _ _ _ END (In this frame constant at the beginning and the end and 4 variables)

Link the templates messages to the state machine to know the order

Fuzzing on the variable parts of each frame with the template that corresponds to the state in state machine

Problem these tools are also limited

It works good on simple systems like a FTP server but less when it is complex .

- Some parts could be interpreted like variables and this fuzzer become useless. For example a bad DateTime, a bad length of payload... If these parameters are incorrect the frames will be directly rejected by the system
- Some protocols use more complex rules than the rules implemented in Pulsar.
- If an ID is reset at each session

=> It has few chance to work correctly on CPS

Excepted results & Future works

Excepted Results

An efficient fuzzer compatible with the maximum of the stateful program and then the less loss of time to configure it
Able to find new bugs or vulnerabilities in systems
Possibilities to improve the initial model during the execution

Future Works

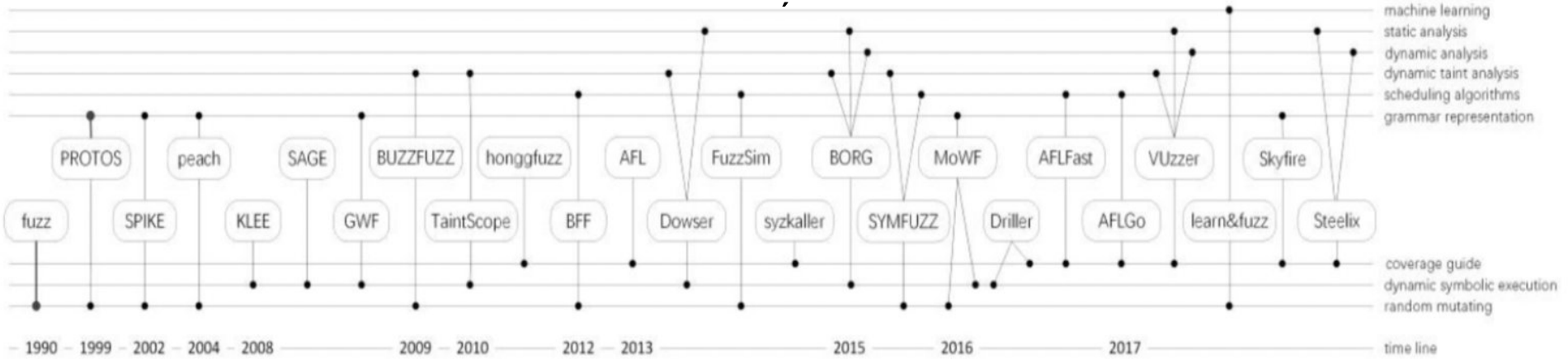
Improve the Fuzzing strategy based on the system reply
Work on differents IP layer
Use information from code sources when available
Improvement of the quality of the models
better recognition of rules, recognition of crc…
Work with encrypted traffic

Etat de l’art en generation de tests de cyber securité (CETIC)

3 classes de techniques pour la génération de tests:

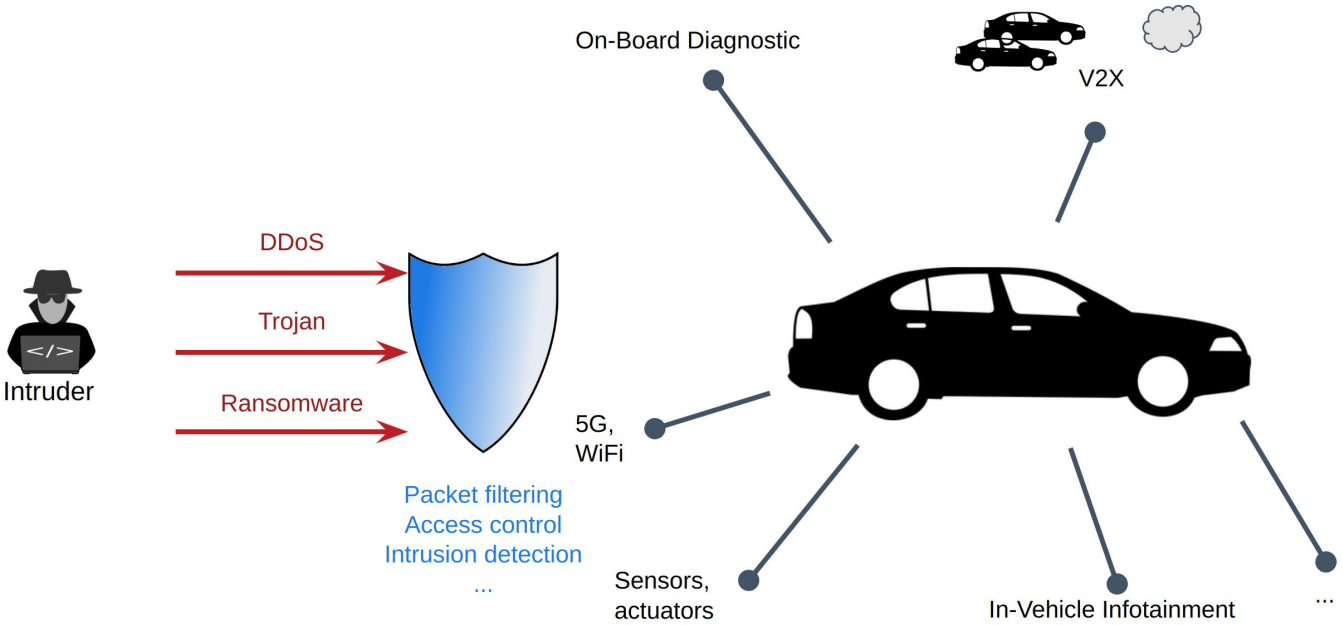
- Sample generation technique
- Dynamic analysis techniques
- Static analysis techniques

Outils (by Chen et al.)



Présentation de l'étude de cas - Platooning

Connected vehicles security



- vehicles are getting more and more connected, which increases their attack surface, making them more vulnerable
- increased computing capabilities in cars allow better protection strategies to counter those new threats

V2X - vehicle to vehicle, vehicle to infrastructure, vehicle to manufacturer, ...

Case Study - Platooning

Platooning with connected vehicles

One **leader** vehicle is followed by N other vehicles (« **followers** »).

The vehicles can exchange information on a **V2V** (vehicule to vehicule) interface and/or the follower can uses sensors to keep distance and direction.

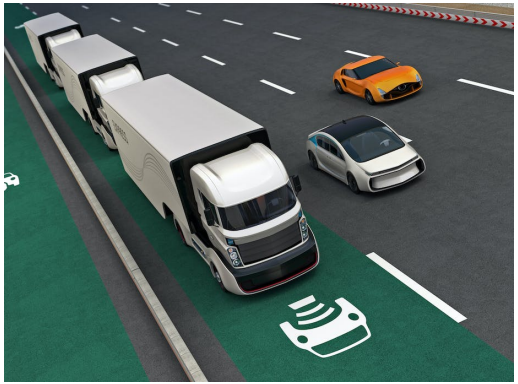
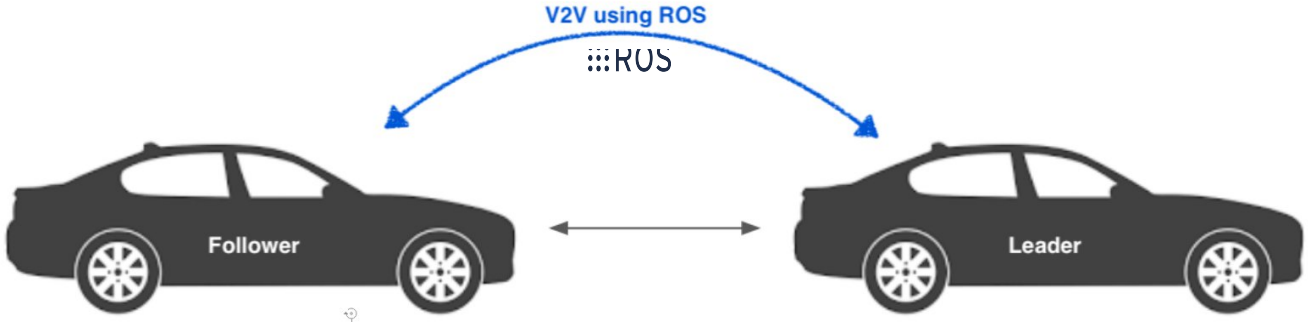
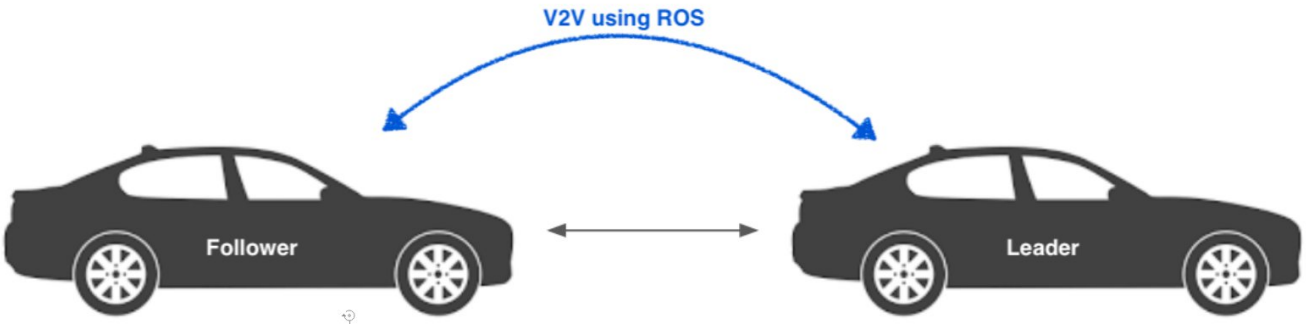


Image source: https://theconversation.com/coming-soon-to-a-highway-near-you-truck-platooning-87748



Case Study - Platooning

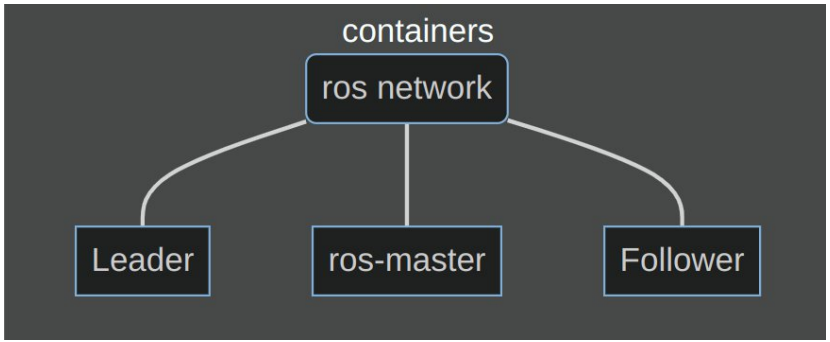
Platooning security vulnerabilities



Any intrusion in the communication can have serious consequences

ROS Protocol v1 :

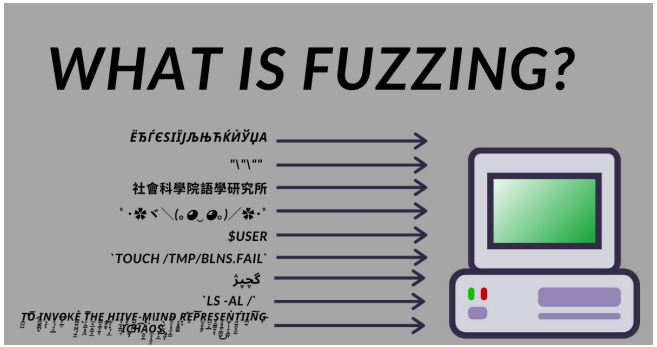
- Publish/Subscribe mechanism with topics
- Use a master to manage communication
- **No encryption**
- **No authentication**
- Basically **No security**

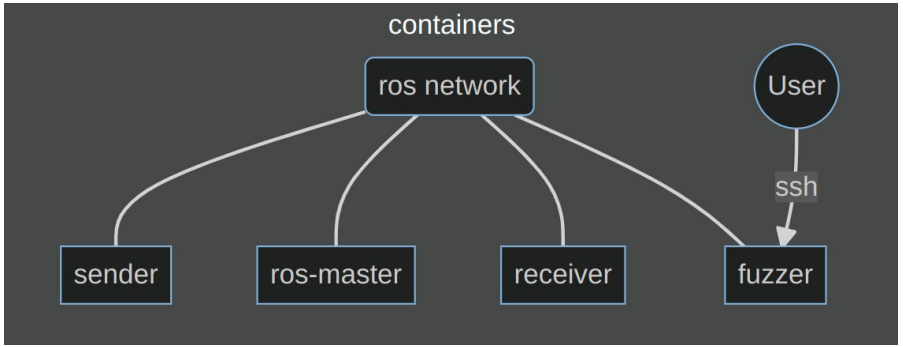


Case Study - Platooning

Testing an unprotected system for vulnerabilities and defects

- Fuzzing tests can be performed on the communications between the vehicles.
- Even if the protocol is unprotected, this can reveal defects or vulnerabilities in the follower software
- Example :
  - Random fuzzing and mutation fuzzing on json formatted communications won't show any effect
  - Grammar Based fuzzing can reveal more vulnerabilities or defects



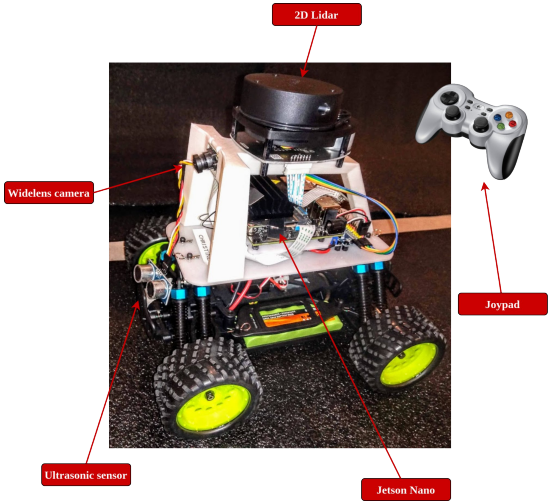


Case Study - Platooning

Exploiting Vulnerabilities

The attacker uses this vulnerability to modify the car behavior and create an accident:

- *forces acceleration, brake, steering*
- *poison camera sensor input*
- *impersonate leader or infrastructure*

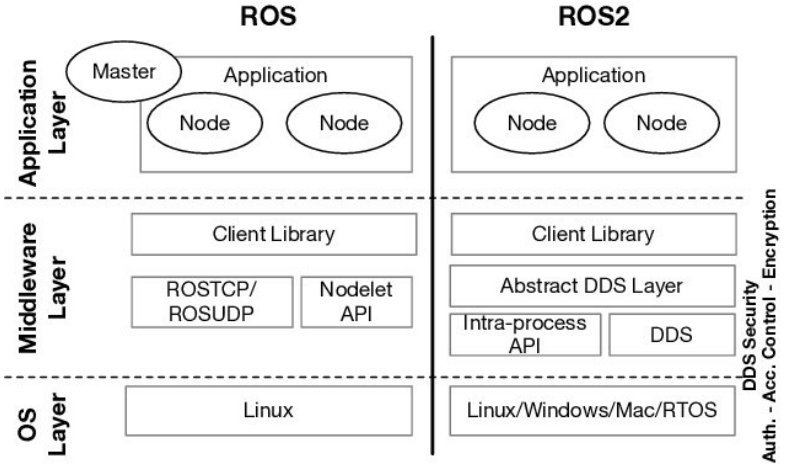


https://youtu.be/fZ8goQkyGUs

Case Study - Platooning

Testing a **protected** system for vulnerabilities and defects

ROS



Mazzeo, Giovanni & Staffa, Mariacarla. (2020). TROS: Protecting Humanoids ROS from Privileged Attackers. International Journal of Social Robotics. 12. 10.1007/s12369-019-00581-4.

ROS2 introduces:

- security - AuthN, AuthZ, communication encryption
- real time
- distributed processing
- resilience & robustness
- ...

Case Study - Platooning

Testing a **protected** system for vulnerabilities and defects

2 interesting papers about vulnerabilities discovered on ROS2 and SROS2 (secured ROS2 variant) :

- PGFUZZ: Policy-Guided Fuzzing for Robotic Vehicles (source)
- On the (In)Security of Secure ROS2 (source)

2 related CVEs :

CVE-2019-19625 Detail

Description

SROS 2 0.8.1 (which provides the tools that generate and distribute keys for Robot Operating System 2 and uses the underlying security plugins of DDS from ROS 2) leaks node information due to a leaky default configuration as indicated in the policy/defaults/dds/governance.xml document.

Severity CVSS Version 3.x | CVSS Version 2.0

CVSS 3.x Severity and Metrics:

NIST: NVD	Base Score: 5.3 MEDIUM	Vector: CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N
CNA: MITRE	Base Score: 7.5 HIGH	Vector: CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:H

QUICK INFO

CVE Dictionary Entry: CVE-2019-19625
NVD Published Date: 12/06/2019
NVD Last Modified: 12/13/2019
Source: MITRE

CVE-2019-19627 Detail

Description

SROS 2 0.8.1 (after CVE-2019-19625 is mitigated) leaks ROS 2 node-related information regardless of the rtps_protection_kind configuration. (SROS2 provides the tools to generate and distribute keys for Robot Operating System 2 and uses the underlying security plugins of DDS from ROS 2.)

Severity CVSS Version 3.x | CVSS Version 2.0

CVSS 3.x Severity and Metrics:

NIST: NVD	Base Score: 5.3 MEDIUM	Vector: CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N
CNA: MITRE	Base Score: 7.5 HIGH	Vector: CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:H

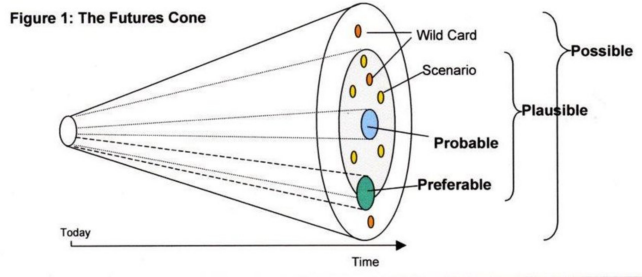
QUICK INFO

CVE Dictionary Entry: CVE-2019-19627
NVD Published Date: 12/06/2019
NVD Last Modified: 12/13/2019
Source: MITRE

Case Study - Platooning

Protecting a system using plausibility analysis

Remote Access Software (MITRE T1219)
Data Manipulation (MITRE T1565)



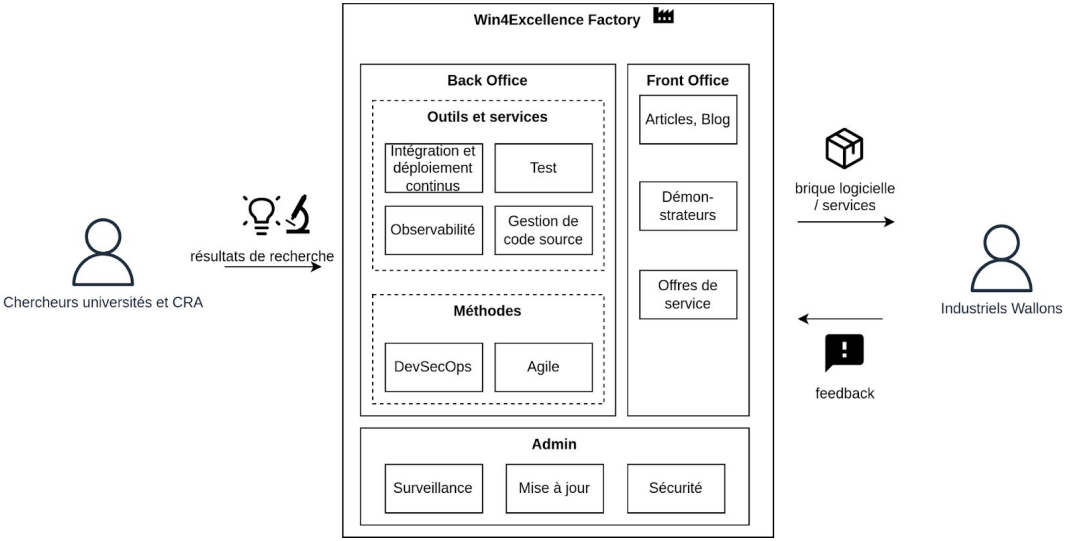
The cone of plausibility (adapted from Taylor, 1993; image retrieved from http://thinkingfutures.net/)

Initial Access	Execution	Persistence	Privilege Escalation	Evasion	Discovery	Lateral Movement	Collection	Command and Control	Inhibit Response Function	Impair Process Control	Impact
12 techniques	9 techniques	6 techniques	2 techniques	6 techniques	5 techniques	7 techniques	10 techniques	3 techniques	13 techniques	5 techniques	12 techniques
Drive-by Compromise	Change Operating Mode	Hardcoded Credentials	Exploitation for Privilege Escalation	Change Operating Mode	Network Connection Enumeration	Default Credentials	Adversary-in-the-Middle	Commonly Used Port	Activate Firmware Update Mode	Brute Force I/O	Damage to Property
Exploit Public-Facing Application	Command-Line Interface	Modify Program	Hooking	Exploitation for Evasion	Network Sniffing	Exploitation of Remote Services	Automated Collection	Connection Proxy	Alarm Suppression	Modify Parameter	Denial of Control
Exploitation of Remote Services	Execution through API	Module Firmware		Indicator Removal on Host	Remote System Discovery	Hardcoded Credentials	Data from Information Repositories	Standard Application Layer Protocol	Block Command Message	Module Firmware	Denial of View
External Remote Services	Graphical User Interface	Project File Infection		Masquerading	Remote System Information Discovery	Lateral Tool Transfer	Detect Operating Mode		Block Reporting Message	Spoof Reporting Message	Loss of Availability
Internet Accessible Device	Hooking	System Firmware		Rootkit	Wireless Sniffing	Program Download	I/O Image		Block Serial COM	Unauthorized Command Message	Loss of Control
Remote Services	Modify Controller Tasking	Valid Accounts		Spoof Reporting Message		Remote Services	Monitor Process State		Data Destruction		Loss of Productivity and Revenue
Replication Through Removable Media	Native API					Valid Accounts	Point & Tag Identification		Denial of Service		Loss of Protection
Rogue Master	Scripting						Program Upload		Device Restart/Shutdown		Loss of Safety
Spearphishing Attachment	User Execution						Screen Capture		Manipulate I/O Image		Loss of View
Supply Chain Compromise							Wireless Sniffing		Modify Alarm Settings		Manipulation of Control
Transient Cyber Asset									Rootkit		Manipulation of View
Wireless Compromise									Service Stop		Theft of Operational Information
									System Firmware		

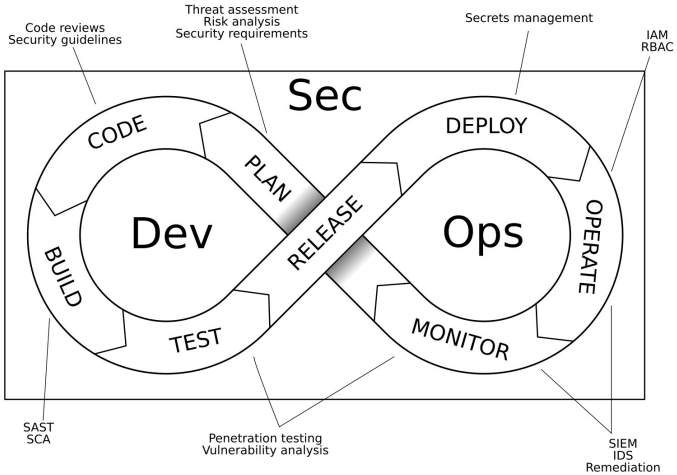
MITRE | ATT&CK®

Knowledge base of adversary tactics and techniques based on real-world observations

Case Study implementation - The Cyber Factory



La Cyber Factory est une usine logicielle (ou *Software Factory* en anglais). Cette plateforme matérielle et logicielle est utilisée sur les projets Win4Excellence pour favoriser la collaboration entre les acteurs de la recherche et de l'industrie wallons, et contribuer à la diffusion des résultats de recherche de ces projets.



DevSecOps - increase quality, speed and security

Discussion sur des vulnérabilités, défauts de conception à tester

- Est-ce que la démarche de défi et de groupes de travail vous permet d’exprimer vos besoins ?
- Quelles sont les difficultés que vous avez avec les tests de pénétration ?
    - Quelles phases de tests souhaitez-vous automatiser ?
    - Quelles vulnérabilités spécifiques faut-il tester (voir CVE) ?
    - ...

Planning réunion de groupe de travail par Défi

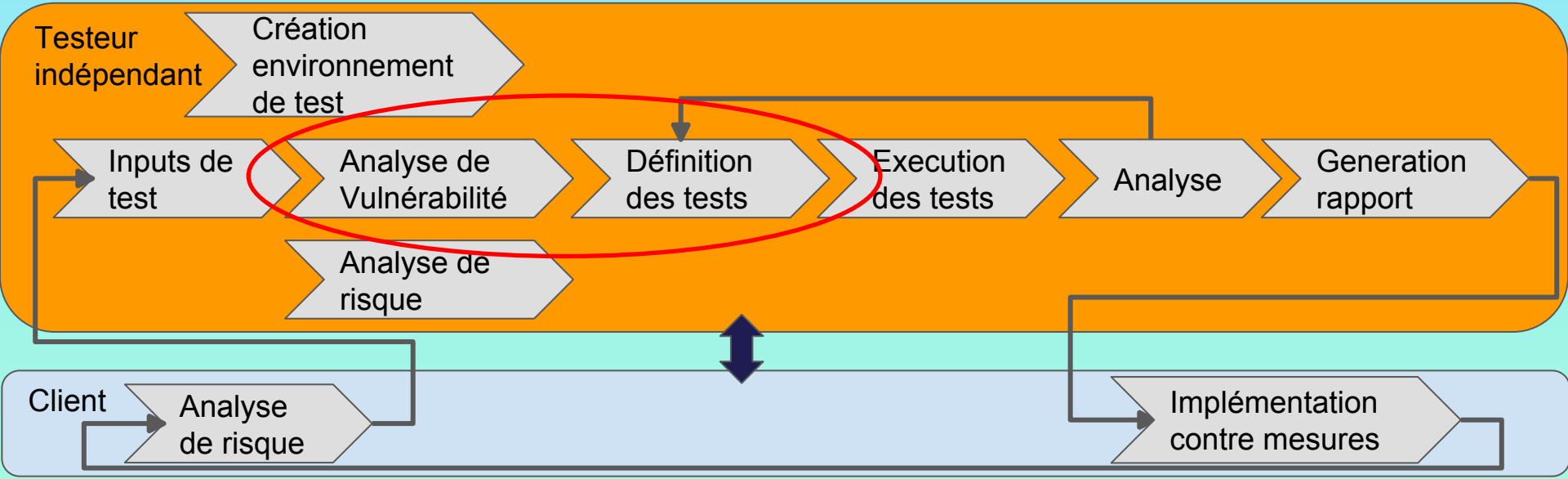
Date	Description
23/01/2023	Première réunion du groupe de travail
*/06/2023 ou */09/2023	Présentation des recherches et discussion sur les démonstrateurs
*/01/2024	Présentation des démonstrateurs
*/06/2024	Présentation des démonstrateurs finaux

Qui participe:

- Entreprises intéressées par le défi
- Responsable de défi
- Chercheurs contribuant au défi
- WSL

Merci de votre attention

Processus de Tests de Pénétration (https://www.cetic.be/CYRUS-EN)



Classes Outils

Network mapping with scanner
Database Scan
SAST
DAST & WAST
Fuzzing Tools
Wireless attacks
Compliance assessment
Sniffing
MITM
Vulnerability analysis
Firmware analysis
Password attacks
Exploitation tools
Hardware
Forensics