Francesco Caturano
Tutor: Simon Pietro Romano
XXXIV Cycle - III year presentation

Automated Offensive Security:
Intelligence is all you need

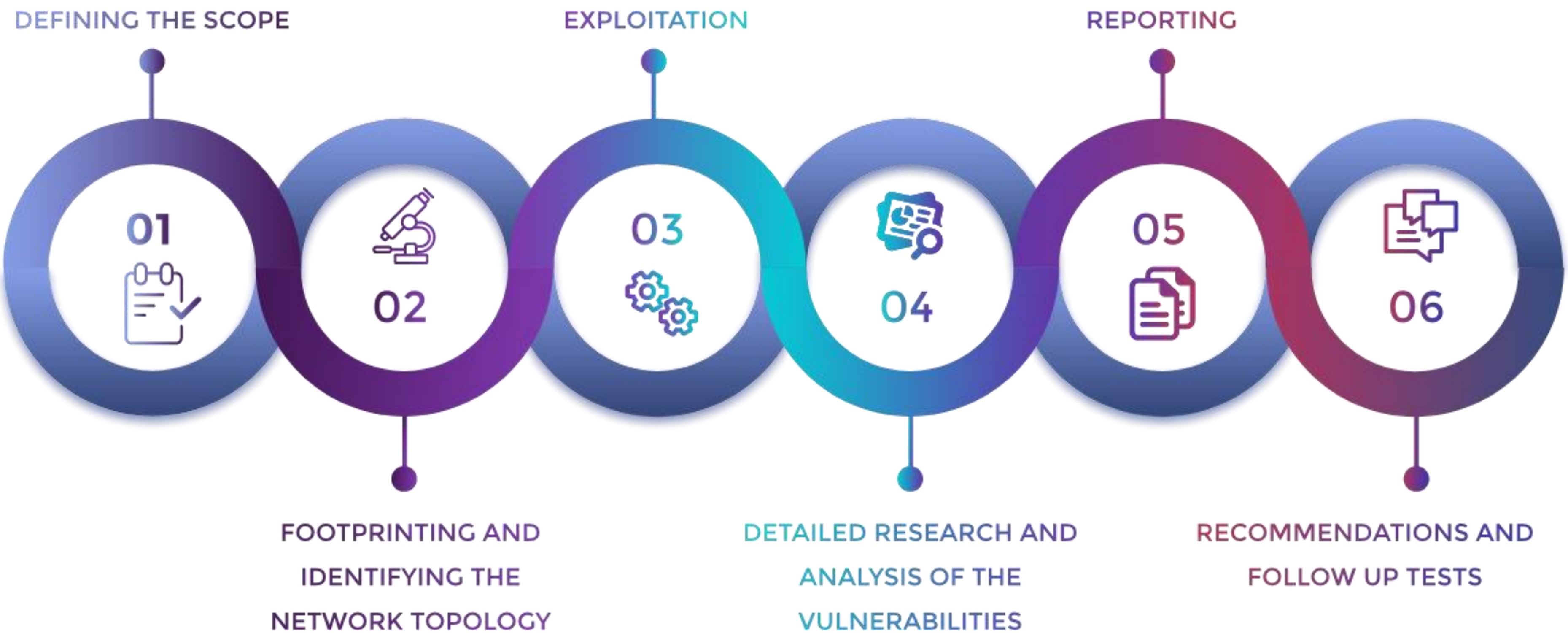
Università degli Studi di Napoli Federico II

Background

- Master's Degree in 2018
  - Automated discovery of CoAP-enabled IoT devices
- GARR scholarship (years 2019-2020)
  - Docker Security Playground
- Sec.S.I. Research Group
  - University spin-off

Context & Contribution

- Context
  - Offensive Security
  - Penetration Testing
    - Web Application Penetration Testing (WAPT)
  - DAST (Dynamic Application Security Testing) tools
- Contribution
  - Intelligent models to improve DAST accuracy and efficiency
    - Intelligent agent for the discovery of Cross-Site Scripting vulnerabilities using Reinforcement Learning
    - Expert system that recommends the best actions to perform in a web penetration test
    - A toolset to enable the collection of dataset for web penetration testing

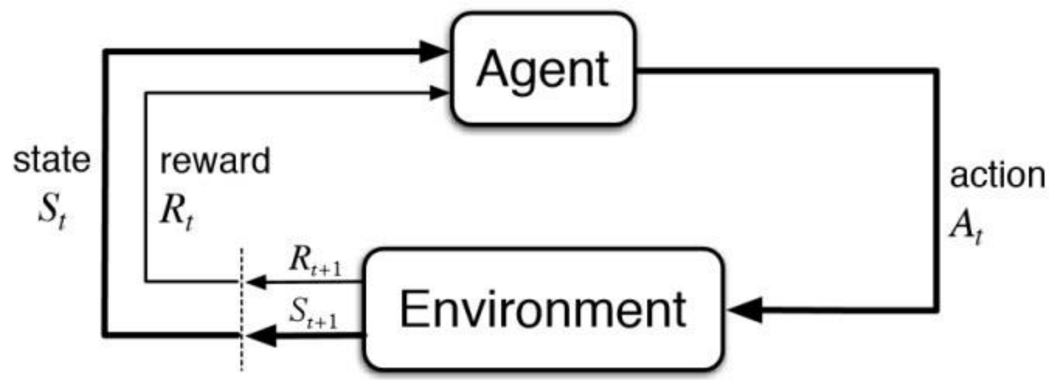


Web Application Penetration Testing

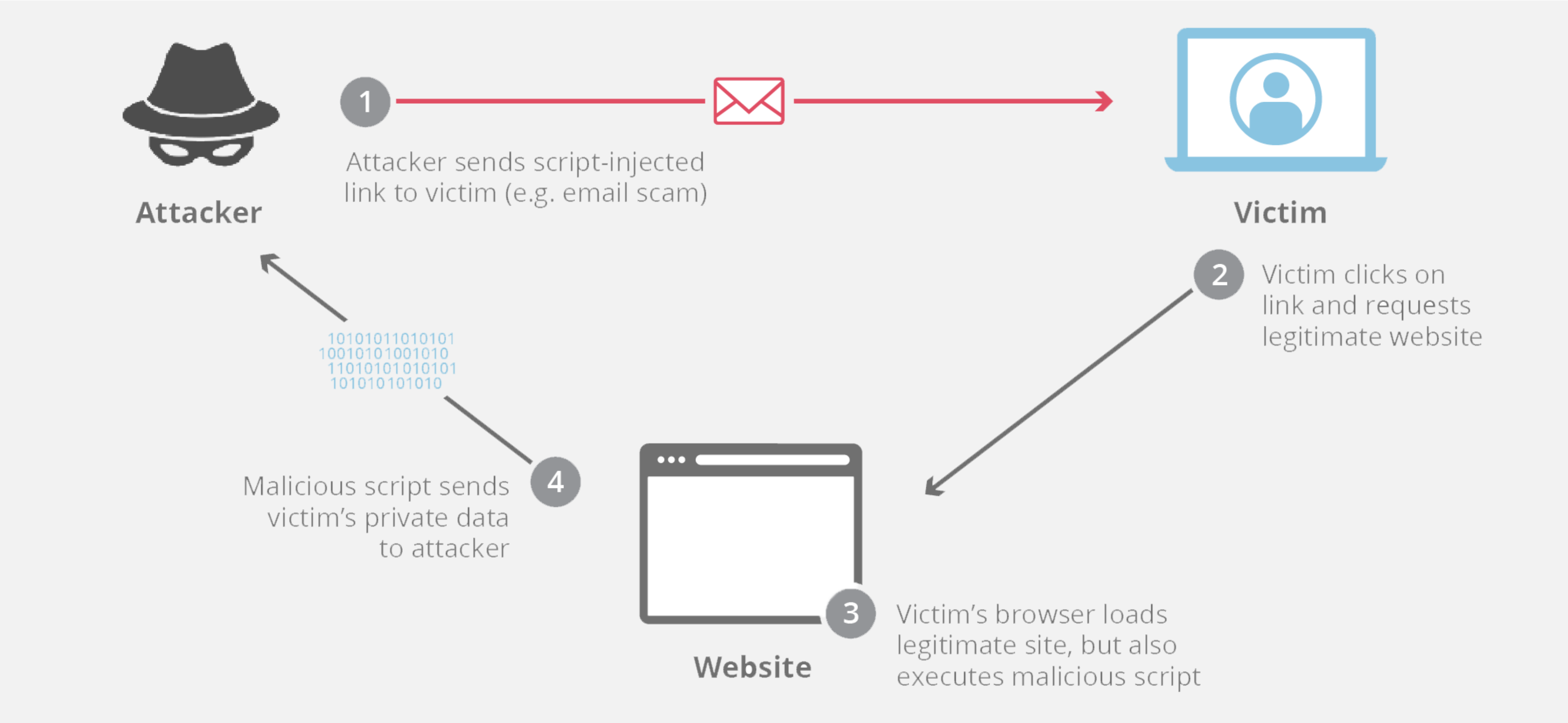
- A sequential decision making process, under uncertainty
- A combination of automated tools and manual inspection
  - Manual testing much more accurate, but tedious
  - Automated tools very inaccurate and inefficient
    - Useful to narrow down the possible tasks to perform
- A different business logic causes the trade-off
  - Experience&Intuition vs. Brute Force

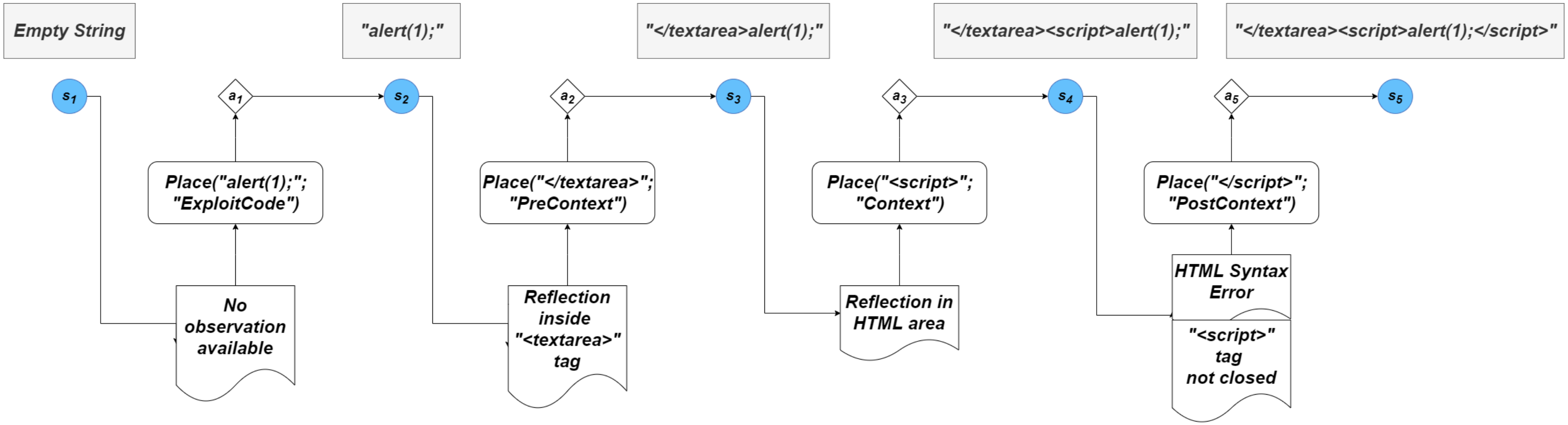
Reinforcement Learning

- An agent
  - learns a policy (a way to perform a task) by interacting with an environment
  - Receives a feedback after every interaction, called reward
  - An algorithm arranges the rewards by assigning a numerical value to each action in any given state
  - The set of best actions for any given state corresponds to the best policy
- Q-Learning



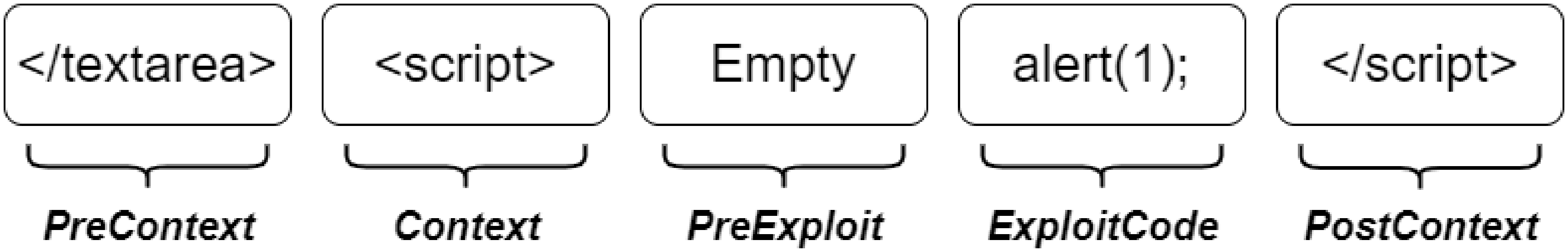
Cross-Site scripting (the attack)





Penetration tester methodology

- Reflection context (<textarea>)
- Escaping (</textarea>)
- New context injection (<script>)
- String well-formedness (</script>)
- Code execution



State-Action space

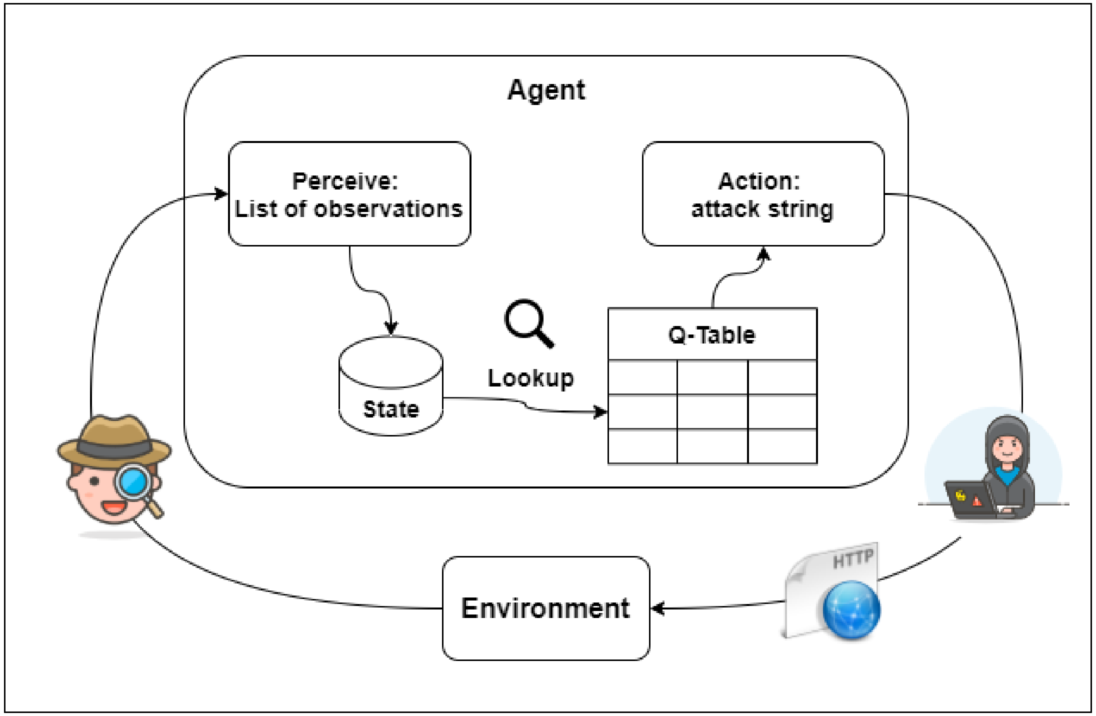
- State
  - Current conditions of the attack string
  - Reflection context
  - Execution context
  - Syntax errors
  - Code execution
- A different action on each attack string section
  - Parameterized action space

Environment Design

- Vulnerable by design applications to emulate attacks and practice hacking techniques
- WAVSEP (Web Application Vulnerability Scanner Evaluation Project)
  - Benchmark
  - Outdated (last commit 2013)
- Enlargement work to bring WAVSEP to the current state of the art
  - Several online training resources considered
    - OWASP vulnerable machines, PortSwigger Academy

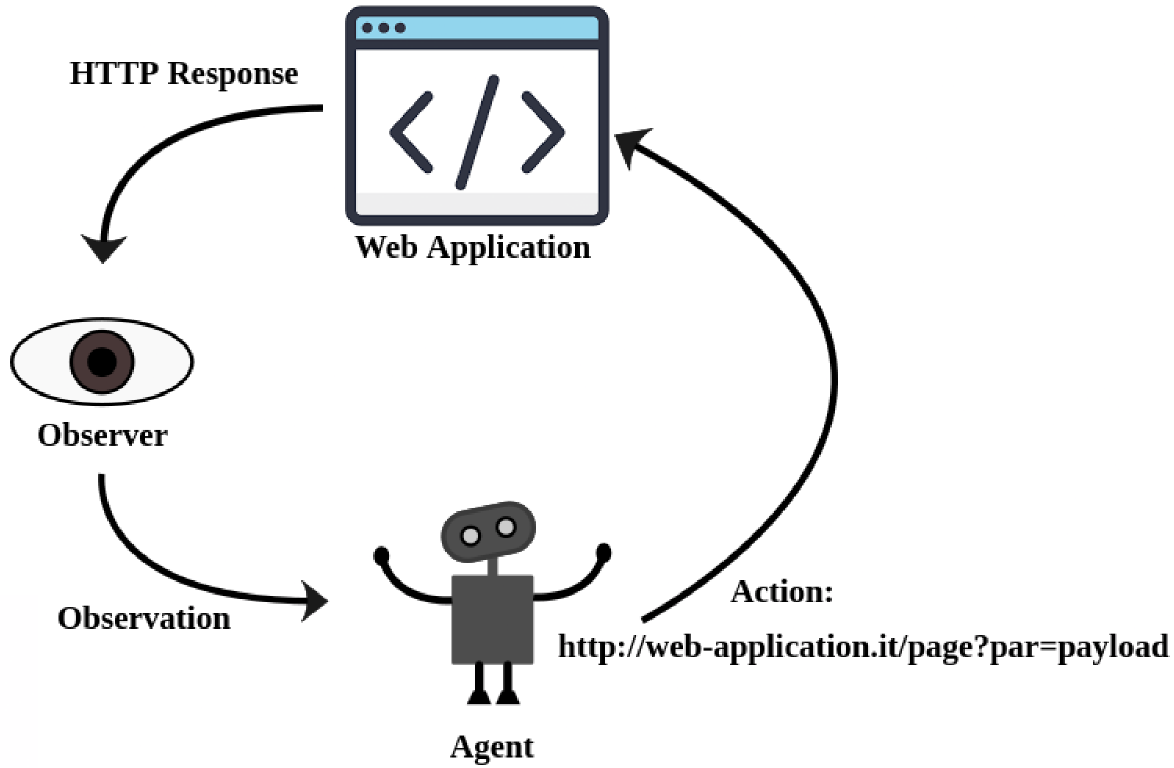
First Iteration

- A semi-automated platform
  - Suggestions to the penetration tester
  - The human in the loop takes care of the interactions with the web application



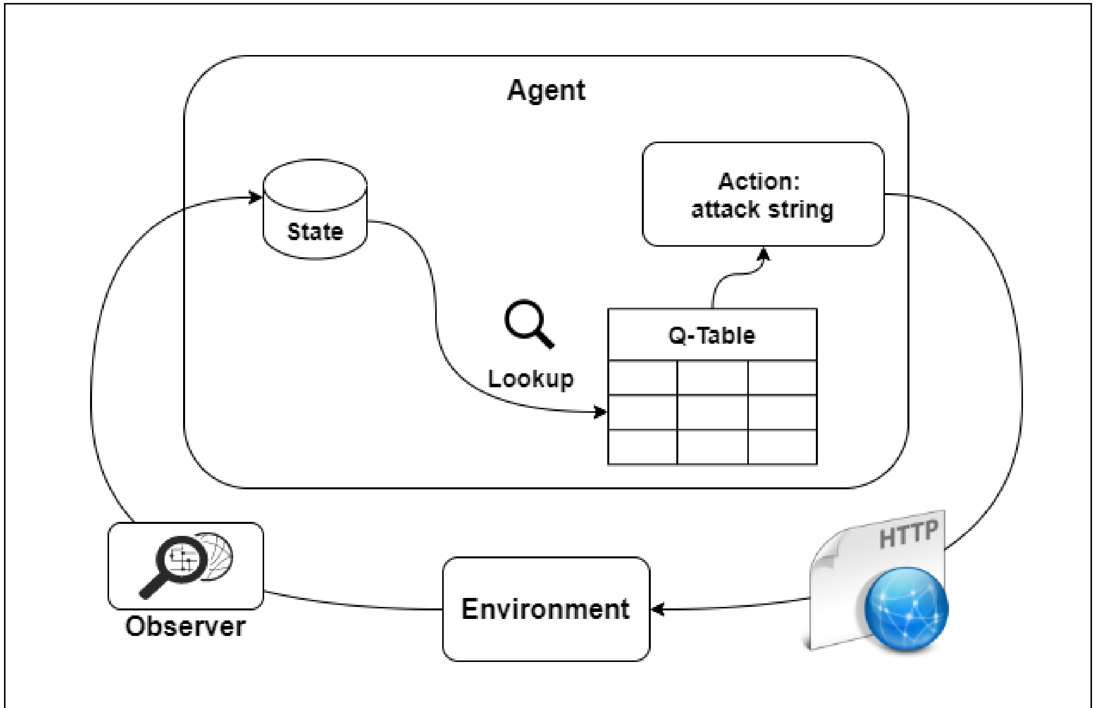
Second iteration

- An automated module, called Observer
  - Sends attack strings in HTTP requests
  - Analyzes the responses looking for behavior representative of Cross-Site scripting vulnerabilities
  - Query Xpath and Selenium headless browser



Third iteration

- Fully automated intelligent agent
  - Reinforcement Learning environment based on Gym OpenAI
  - Integration with Observer module

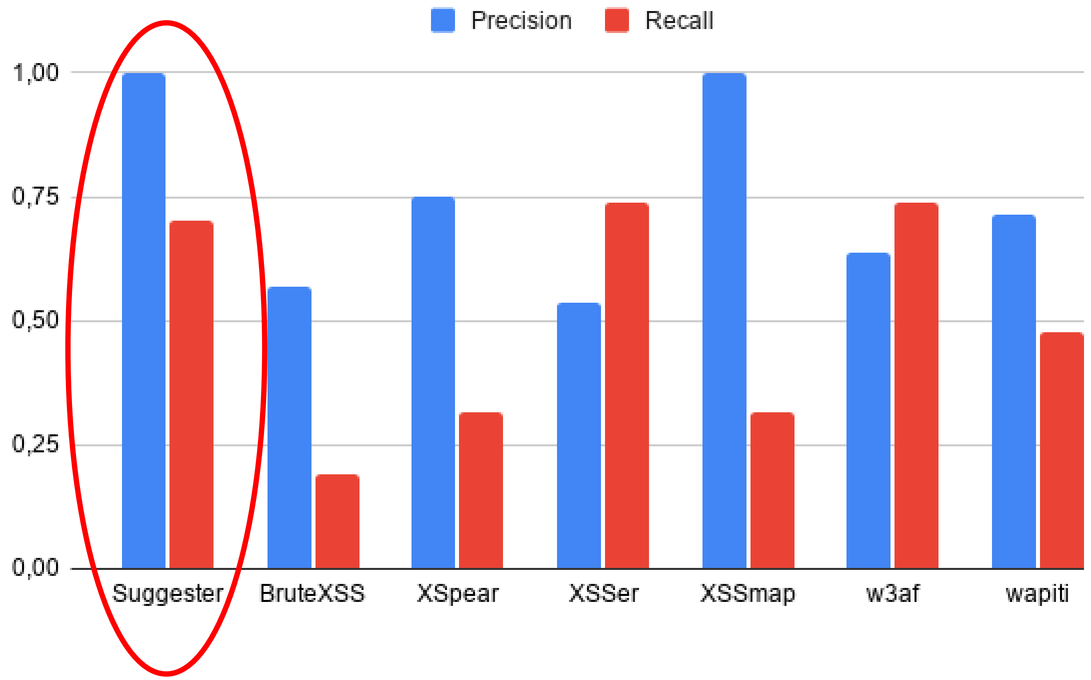


1. Send attack string to the web application
2. Observe the response
3. Identify the state
4. Lookup the corresponding best action
5. Send a new attack string

Performance evaluation (1 of 3) - Accuracy

$$recall = \frac{TP}{TP + FN}$$

$$precision = \frac{TP}{TP + FP}$$



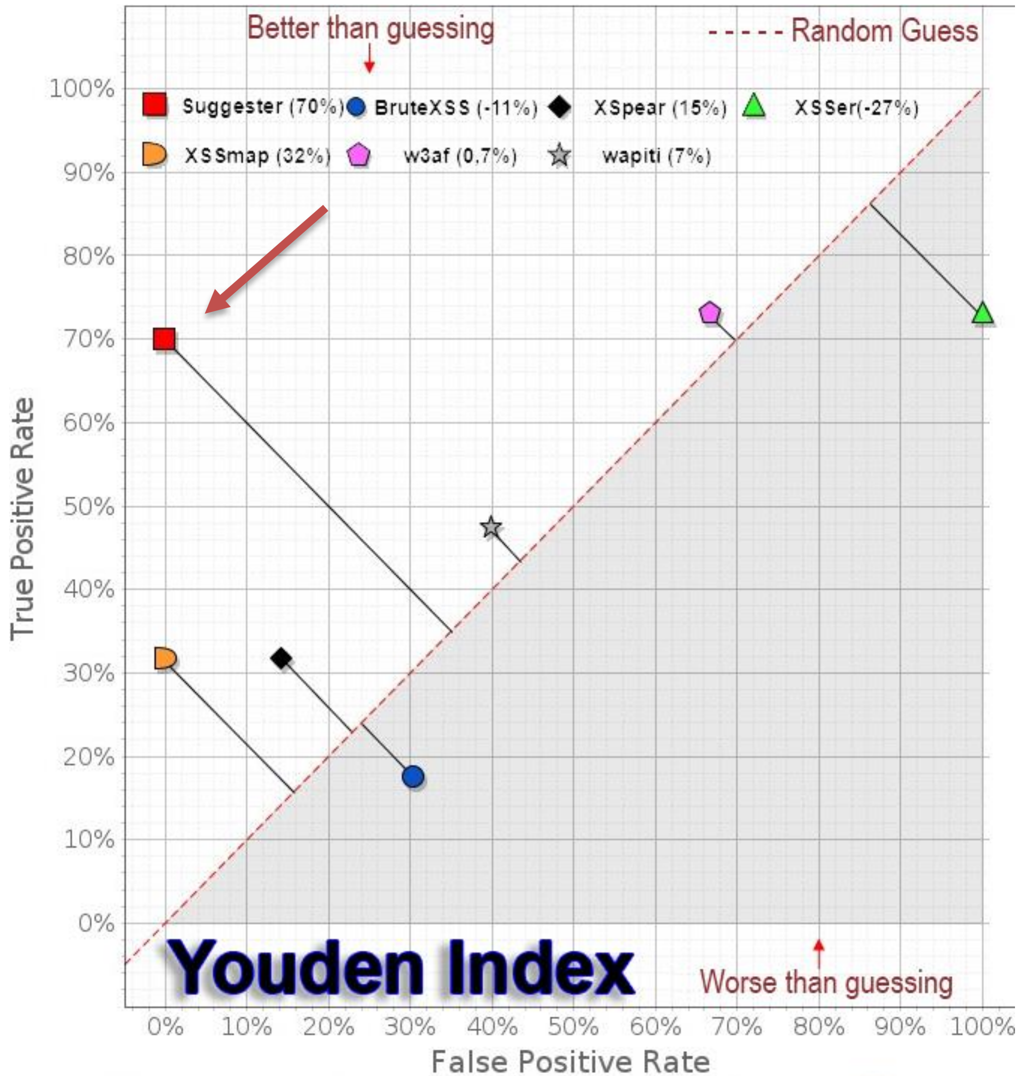
Yahoo Webseclab benchmarking platform

Performance evaluation (2 of 3) - Accuracy

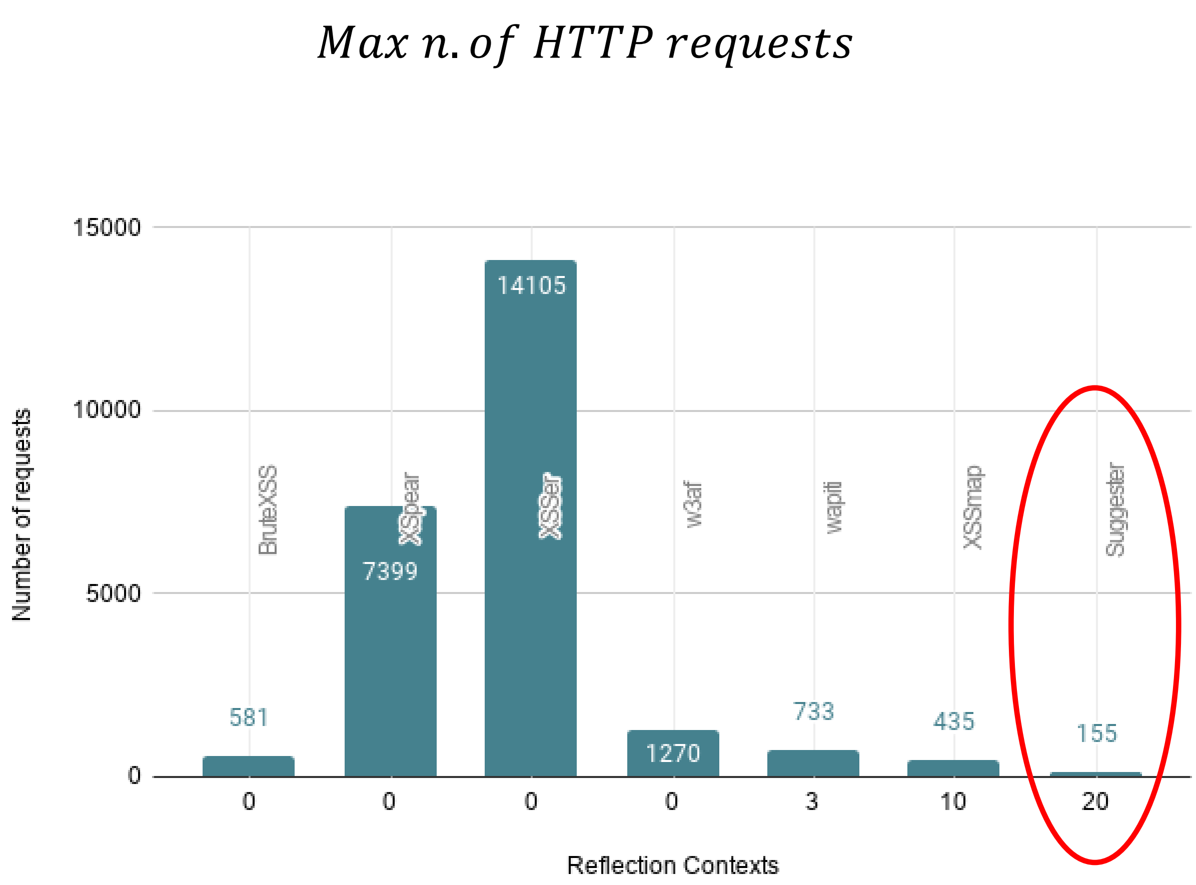
$$J = sensitivity + specificity - 1$$

$$sensitivity = \frac{TP}{TP + FN}$$

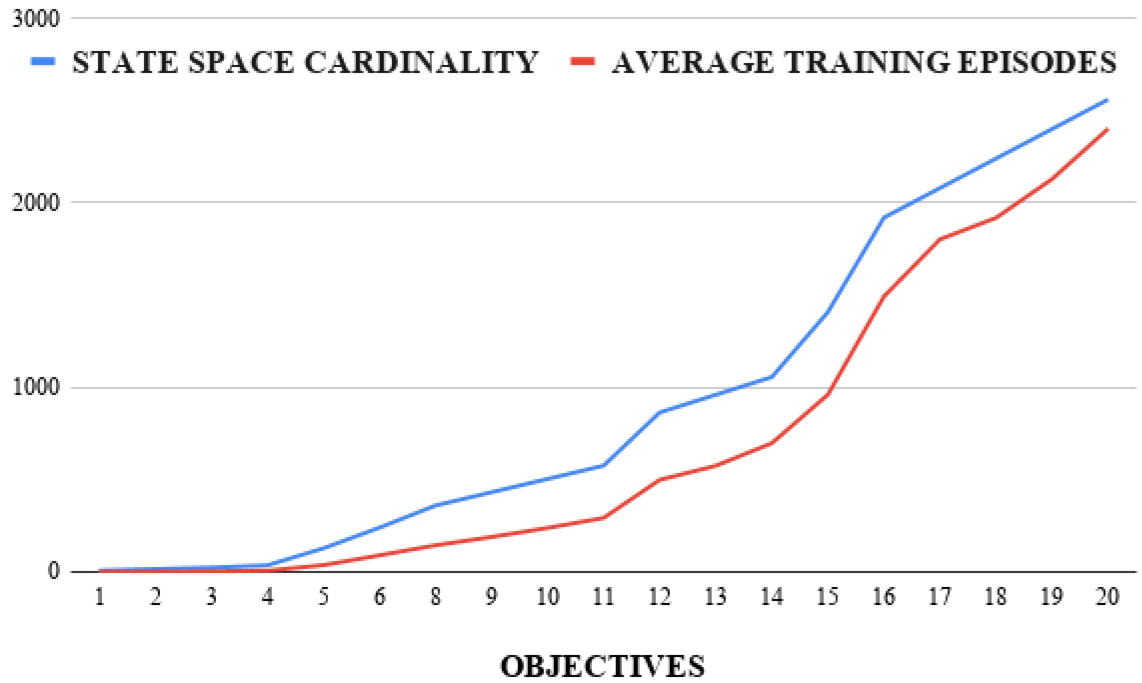
$$specificity = \frac{TN}{TN + FP}$$



Performance evaluation (3 of 3) - Efficiency



Future Work



- Categorical nature of the problem
  - Training increases with larger state-action spaces
  - Environment that encompasses more states than the “real” ones
  - Unable to take advantage of Neural Networks (Deep Reinforcement Learning)
- Solution
  - Use models that capture the dynamics of the system and then apply Reinforcement Learning

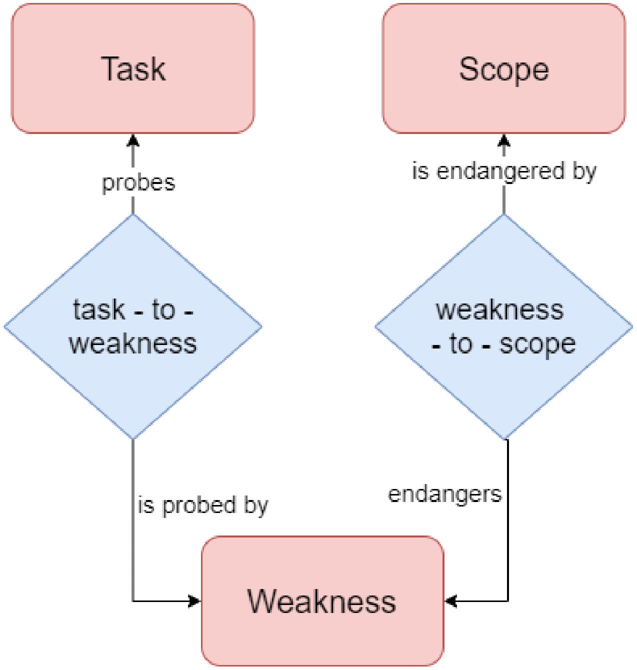
PT Expert system supported by knowledge graphs

- An ontology for web application penetration testing…
- …based on…
  - Web Hacker's handbook
  - OWASP Testing guide
  - CWE
- …represented in the form of a knowledge graph
  - Visualization of attack paths

GRAKN.AI

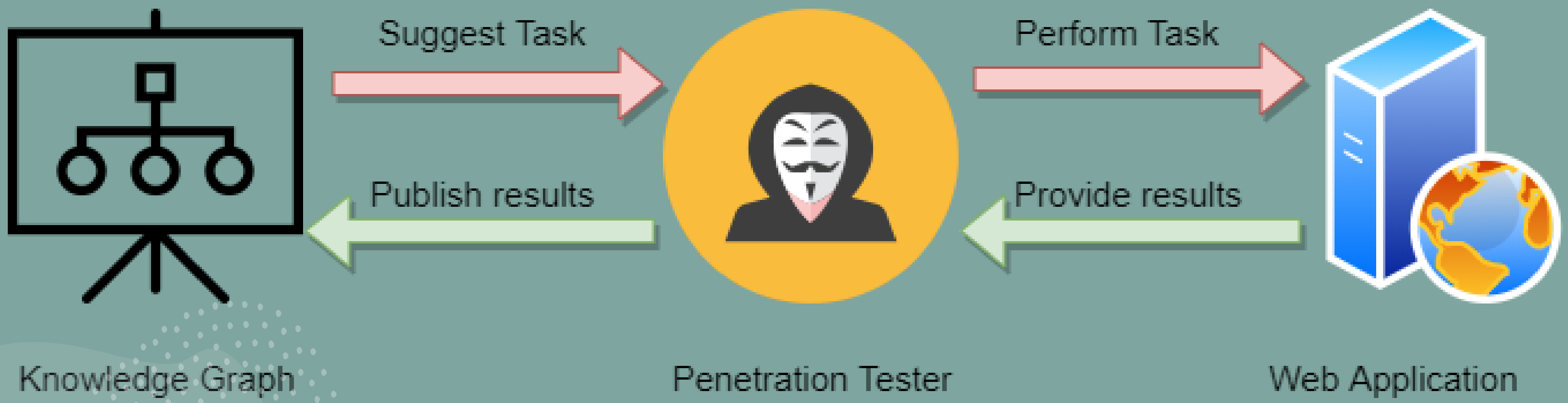
Chain of tasks

- Built around the concept of “Hacking Goal”
  - The objective pursued by the penetration tester
    - E.g. find all SQL Injection vulnerabilities
- A system that outputs the list of tasks to be performed to reach the Hacking Goal
  - Tasks are performed with a combination of manual actions and automated tools
  - An action is an HTTP request
  - Dependencies among tasks
    - Some tasks depend on the results of the previous ones



Recommendation system

1. Users set the goal
2. The system outputs the task to perform next
3. Users perform the recommended actions and review the results
4. Users insert results into the system
5. The system elaborates the results and outputs the next task in the chain



Francesco Caturano

Toolset for web penetration testing dataset

- Proxy architecture to capture:
  - user interactions with the browser
  - generated network traffic
- Dataset storage
- Video playout feature
  - Reproduction of the steps performed during the session.
  - Proves that the collected events are sufficient to recreate the session.

Publications

- CATURANO, Francesco; PERRONE, Gaetano; ROMANO, Simon Pietro. Hacking Goals: A Goal-Centric Attack Classification Framework. In: *IFIP International Conference on Testing Software and Systems*. Springer, Cham, 2020. p. 296-301.
- CATURANO, Francesco; PERRONE, Gaetano; ROMANO, Simon Pietro. Discovering reflected cross-site scripting vulnerabilities using a multiobjective reinforcement learning environment. *Computers & Security*, 2021, 103: 102204.
- CATURANO, Francesco; PERRONE, Gaetano; ROMANO, Simon Pietro. Capturing flags in a dynamically deployed microservices-based heterogeneous environment. In: *2020 Principles, Systems and Applications of IP Telecommunications (IPTComm)*. IEEE, 2020. p. 1-7.
- BRIGNOLI, M. A., et al. A distributed security tomography framework to assess the exposure of ICT infrastructures to network threats. *Journal of Information Security and Applications*, 2021, 59: 102833.
- CATURANO, Francesco; JIMÉNEZ, Jaime; ROMANO, Simon Pietro. Automated discovery of CoAP-enabled IoT devices. In: *2019 Eleventh International Conference on Ubiquitous and Future Networks (ICUFN)*. IEEE, 2019. p. 396-401.

Conclusions & Future Work

- Approaches to provide automation to offensive security practices
  - application of a Reinforcement Learning model to create an intelligent agent that discovers Cross-Site scripting vulnerabilities
  - ontology for web application penetration testing represented in the form of a knowledge graph
- Inspired human penetration testing methodologies
  - Improve tools' detection abilities in terms of accuracy and efficiency
- Future work
  - Application of Artificial Intelligent models to hacking datasets