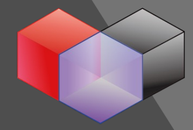


인공지능으로 도전하는 해킹

강화학습으로 웹 해킹 시도해보기

배송현 Otwooo

© Hackingcamp. All Rights Reserved



POC SECURITY

Who Am I

배송현 (Otwooo)

대전대신고등학교 3학년

웹 해킹, 리버싱, 알고리즘, 인공지능

otwooo.github.io/about



© Hackingcamp. All Rights Reserved

발표 내용

인공지능 기반 보안 프로젝트 소개

- 인공지능 이론
- 인공지능을 해킹에 적용해보려고 한 내용

© Hackingcamp. All Rights Reserved

POC SECURITY

요즘 인공지능

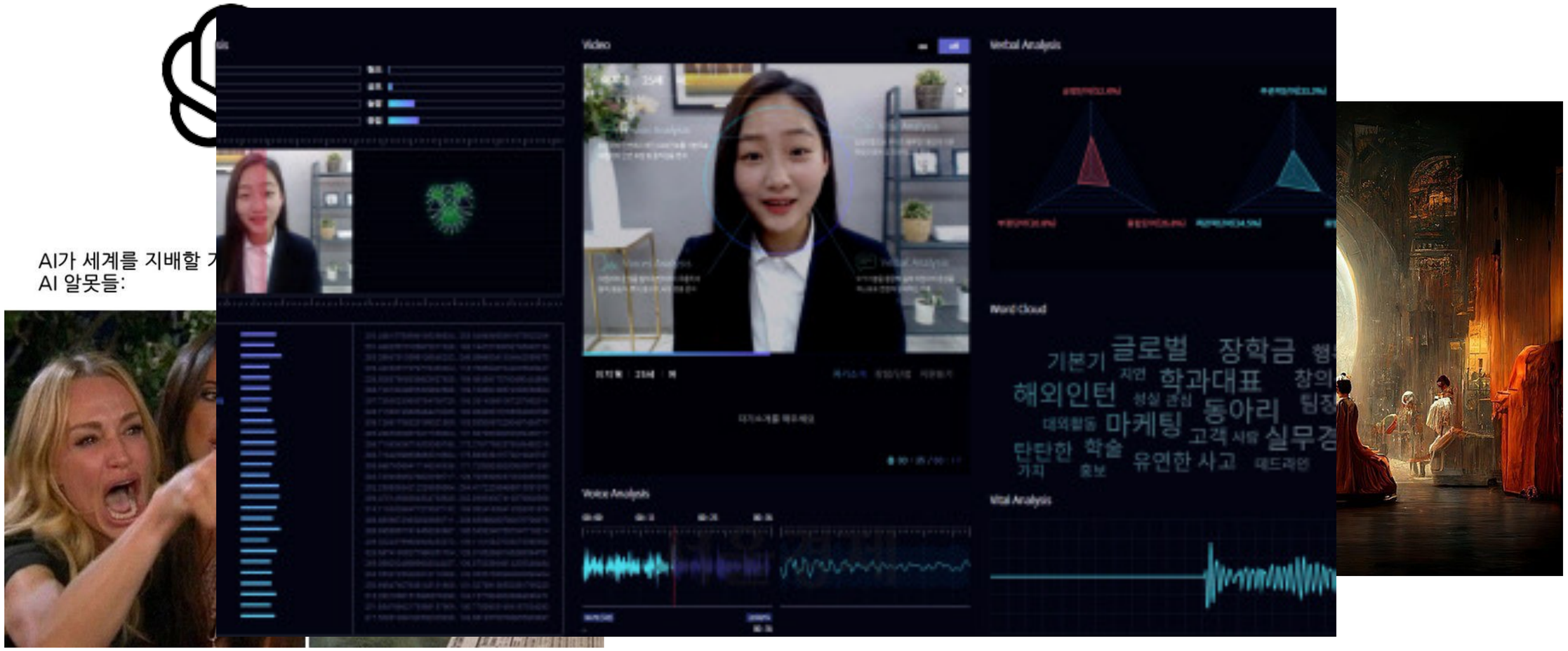
OpenAI





© Hackingcamp. All Rights Reserved

요즘 인공지능



© Hackingcamp. All Rights Reserved

POC SECURITY

요즘 인공지능



© Hackingcamp. All Rights Reserved

POC SECURITY

취약점 점검

Attack Target



OWASP
Open Web Application
Security Project

Vulnerability

© Hackingcamp. All Rights Reserved

이런거는 못하나?

Attack Target



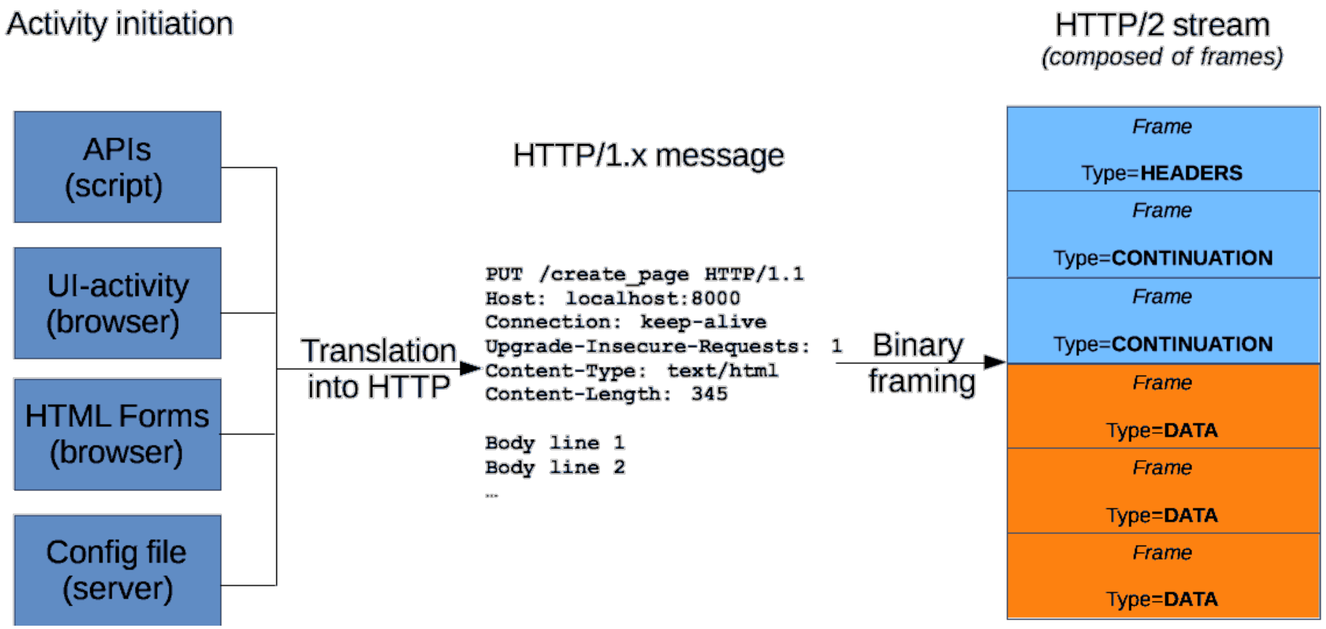
Open Web Application
Security Project

Vulnerability

© Hackingcamp. All Rights Reserved

POC SECURITY

보안 분야에서의 인공지능



공격 탐지

공격 유형 분류

© Hackingcamp. All Rights Reserved

POC SECURITY

보안 분야에서의 인공지능



Introducing VirusTotal Code Insight: Empowering threat analysis with generative AI

MONDAY, APRIL 24, 2023 | BERNARDO.QUINTERO | LEAVE A COMMENT

At the RSA Conference 2023 today, we are excited to unveil VirusTotal Code Insight, a cutting-edge feature that leverages artificial intelligence for code analysis. Powered by Google Cloud Security AI Workbench, Code Insight produces natural language summaries of code snippets with ease. This functionality empowers security experts and analysts by providing them with deeper insights into the purpose and operation of analyzed code, significantly enhancing their capability to detect and mitigate potential threats.

인공지능 기반 악성 코드 분석

© Hackingcamp. All Rights Reserved

보안 분야에서의 인공지능



```
}
dword_4BEC7C = fopen(&FileName, "wb");
if ( dword_4BEC7C )
{
  sub_401F10(lpString2, 0);
  fclose(dword_4BEC7C);
  String1 = 0;
  v10 = 0;
  v11 = 0;
  v12 = 0;
  v4 = strrchr(a2, 92);
  lstrcpyA(&String1, v4 + 1);
  v5 = strrchr(&String1, 46);
```

취약한 함수 탐지

© Hackingcamp. All Rights Reserved

POC SECURITY

가장 큰 문제



질 높은 데이터를 구하기 쉽지 않다.

결국 할 수 있는건 분류, 탐지 뿐

© Hackingcamp. All Rights Reserved

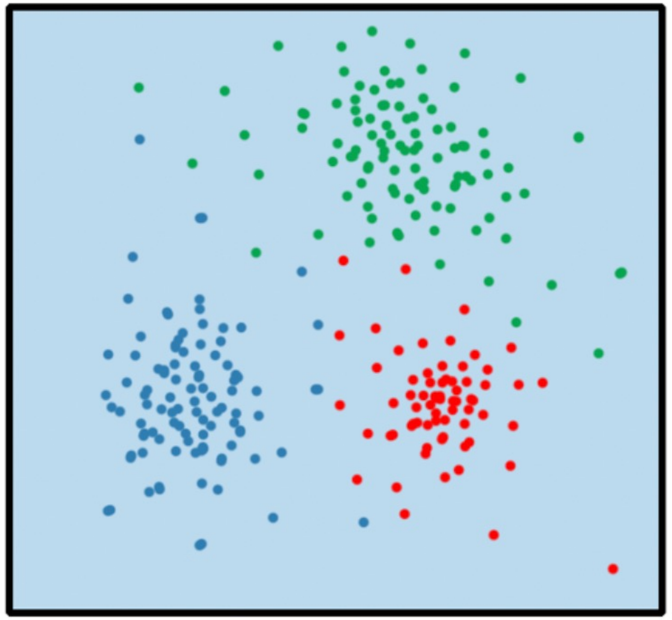
내가 하고 싶은 것

직접 인공지능이 인간처럼 행동하면서
취약점을 점검할 수는 없을까?

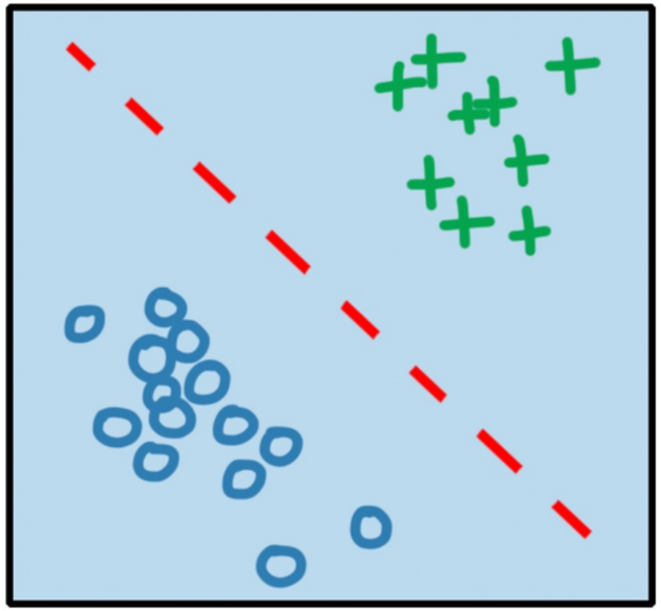
© Hackingcamp. All Rights Reserved

인공지능

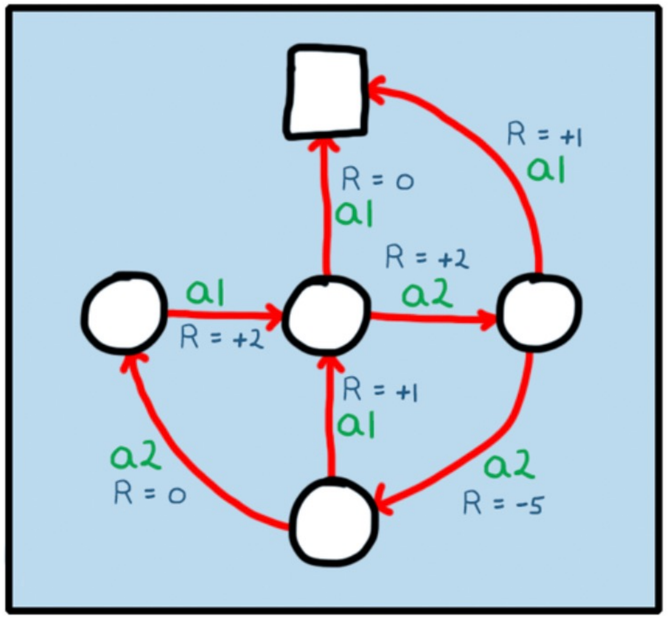
지도 학습



비지도 학습



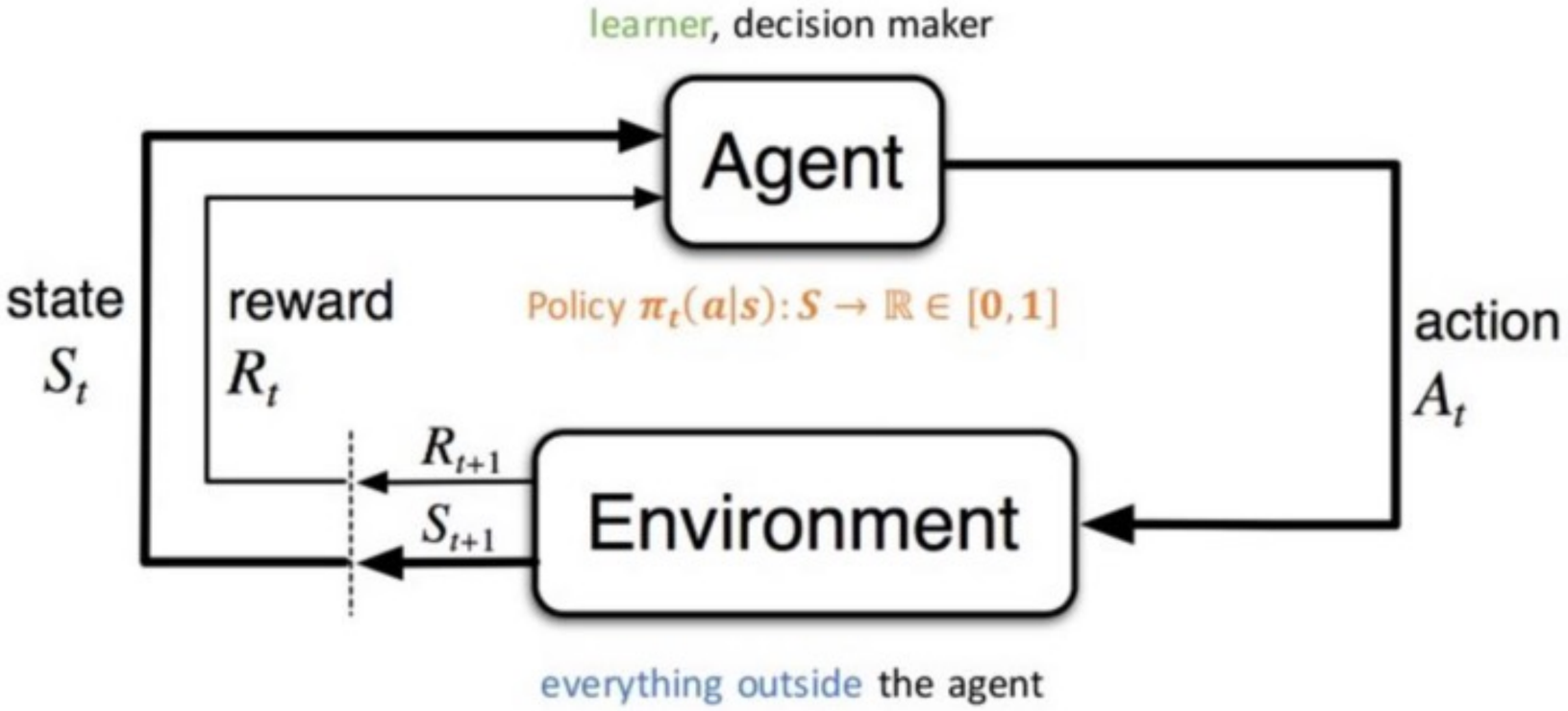
강화 학습



© Hackingcamp. All Rights Reserved

POC SECURITY

강화학습



© Hackingcamp. All Rights Reserved

환경



0: Move left

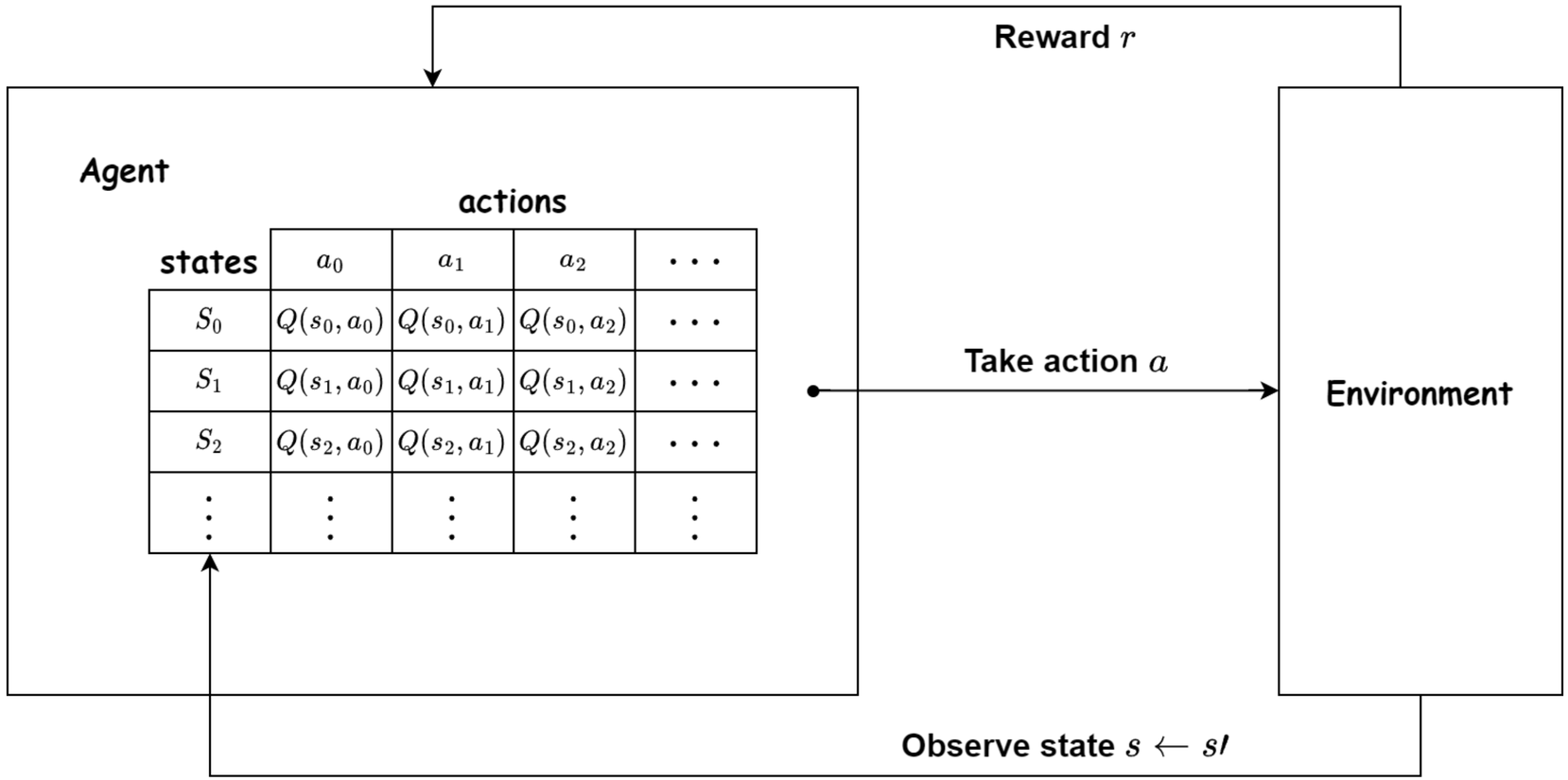
1: Move down

2: Move right

3: Move up

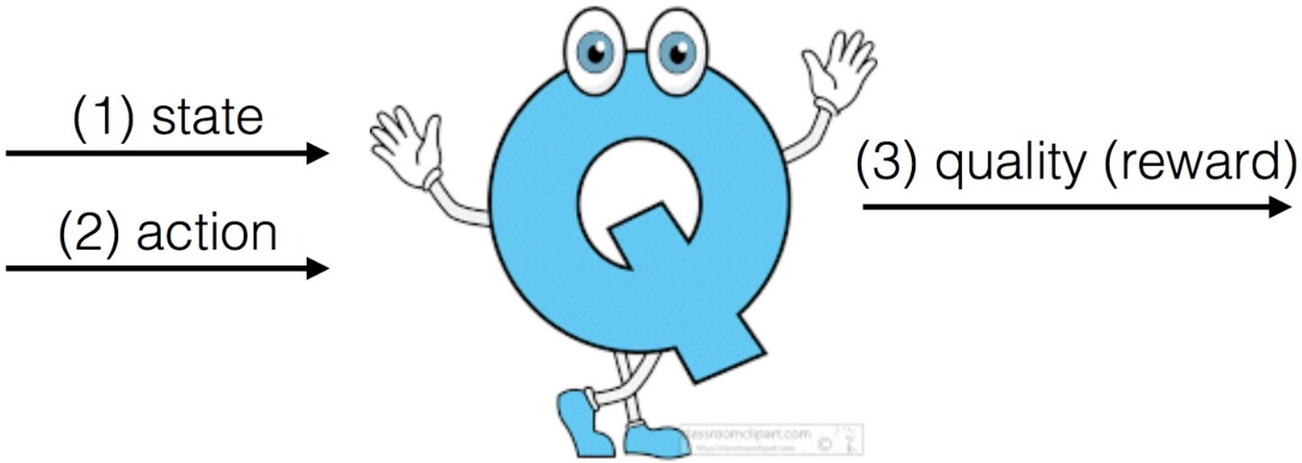
© Hackingcamp. All Rights Reserved

Q-Learning



© Hackingcamp. All Rights Reserved

Q-Learning



Q(상태, 행동)

-> Q value

© Hackingcamp. All Rights Reserved

Q-Learning

s	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15

State

Q(3, Right) = 0

Q(3, Left) = 0.1

Q(3, Up) = 0

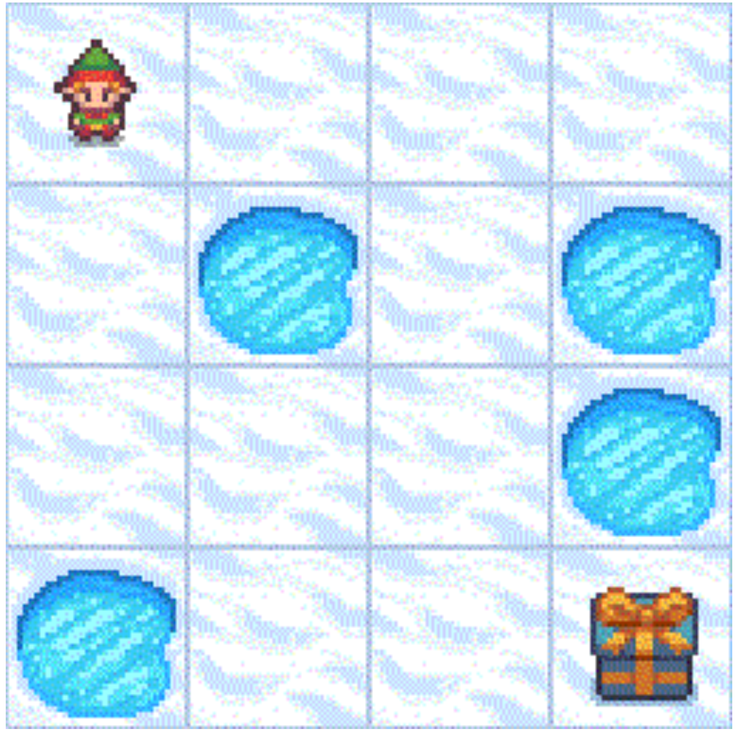
Q(3, Down) = 0.8

© Hackingcamp. All Rights Reserved

Q-Learning

	↑	↓	←	→
state1	-5	0	-5	0
state2	-5	15	0	-5
state3	0	-5	-5	15

Q-Table



© Hackingcamp. All Rights Reserved

Q-Learning

Q value를 어떻게 설정할까?

미래의 보상까지 점수를 매기기 위해 현재 상황부터 끝까지의 상황의 최대 보상을 더해준다.

$$R_t = r_t + r_{t+1} + r_{t+2} + \cdots + r_n$$

© Hackingcamp. All Rights Reserved

Q-Learning

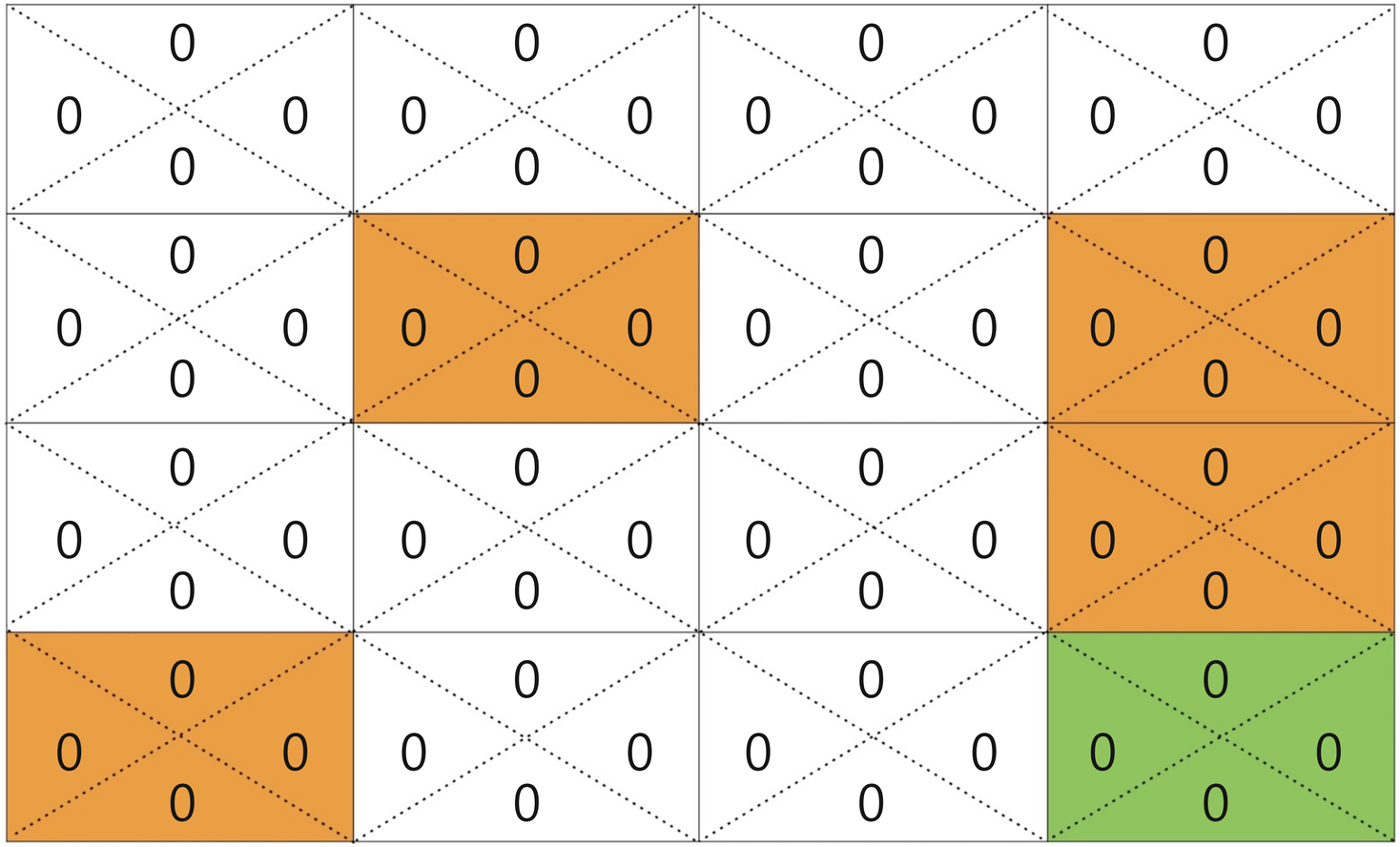
γ 는 할인률로 미래의 보상보다는 현재의 보상을 중요시하기 위해 추가

$$R_t = \gamma\, r_t + \gamma^2\, r_{t+1} + \gamma^3\, r_{t+2} + \cdots + \gamma^n\, r_n$$

$$Q(s,a) = r + \gamma \max_{a\prime} Q(s',a')$$

© Hackingcamp. All Rights Reserved

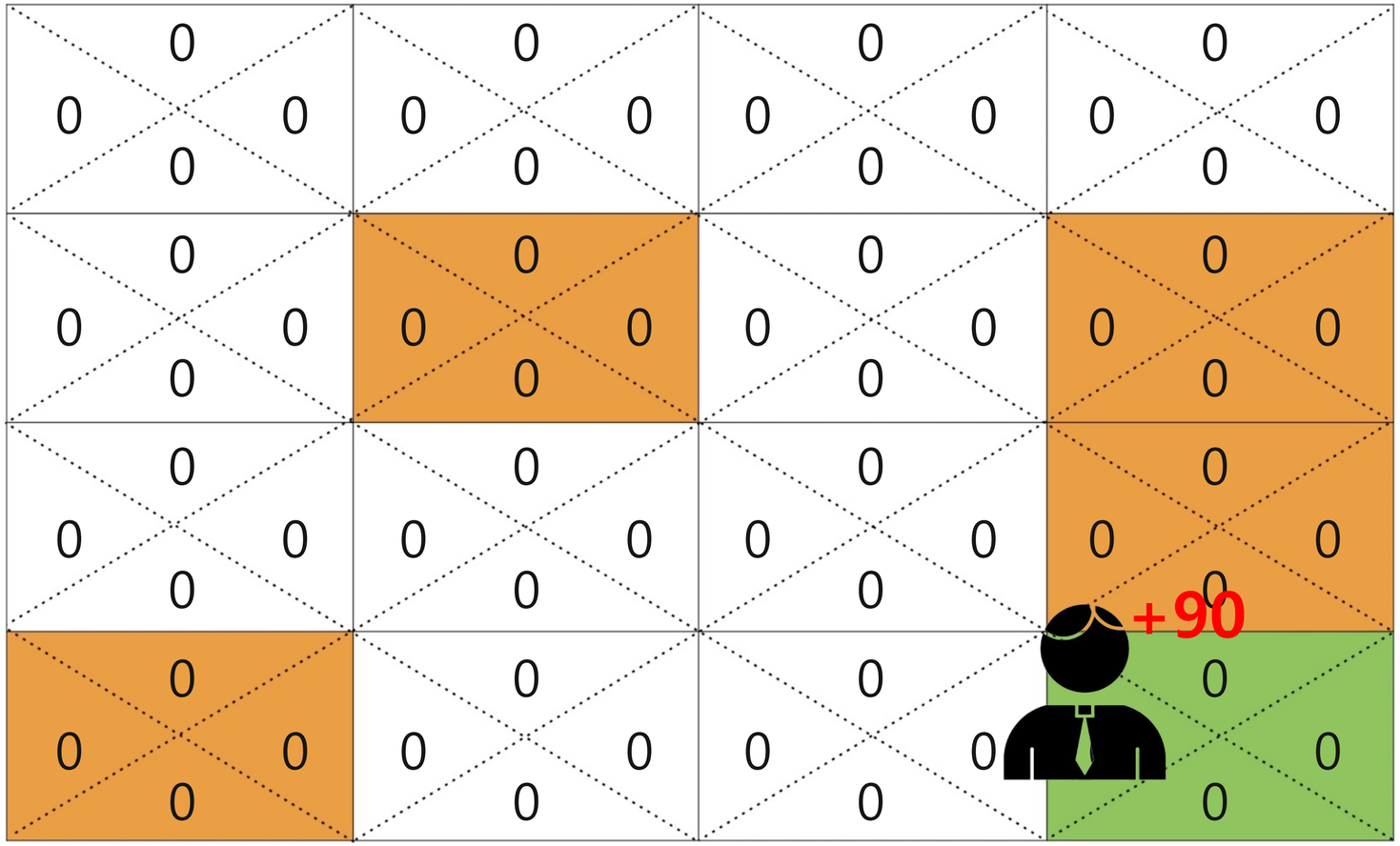
Q-Learning



© Hackingcamp. All Rights Reserved

POC SECURITY

Q-Learning



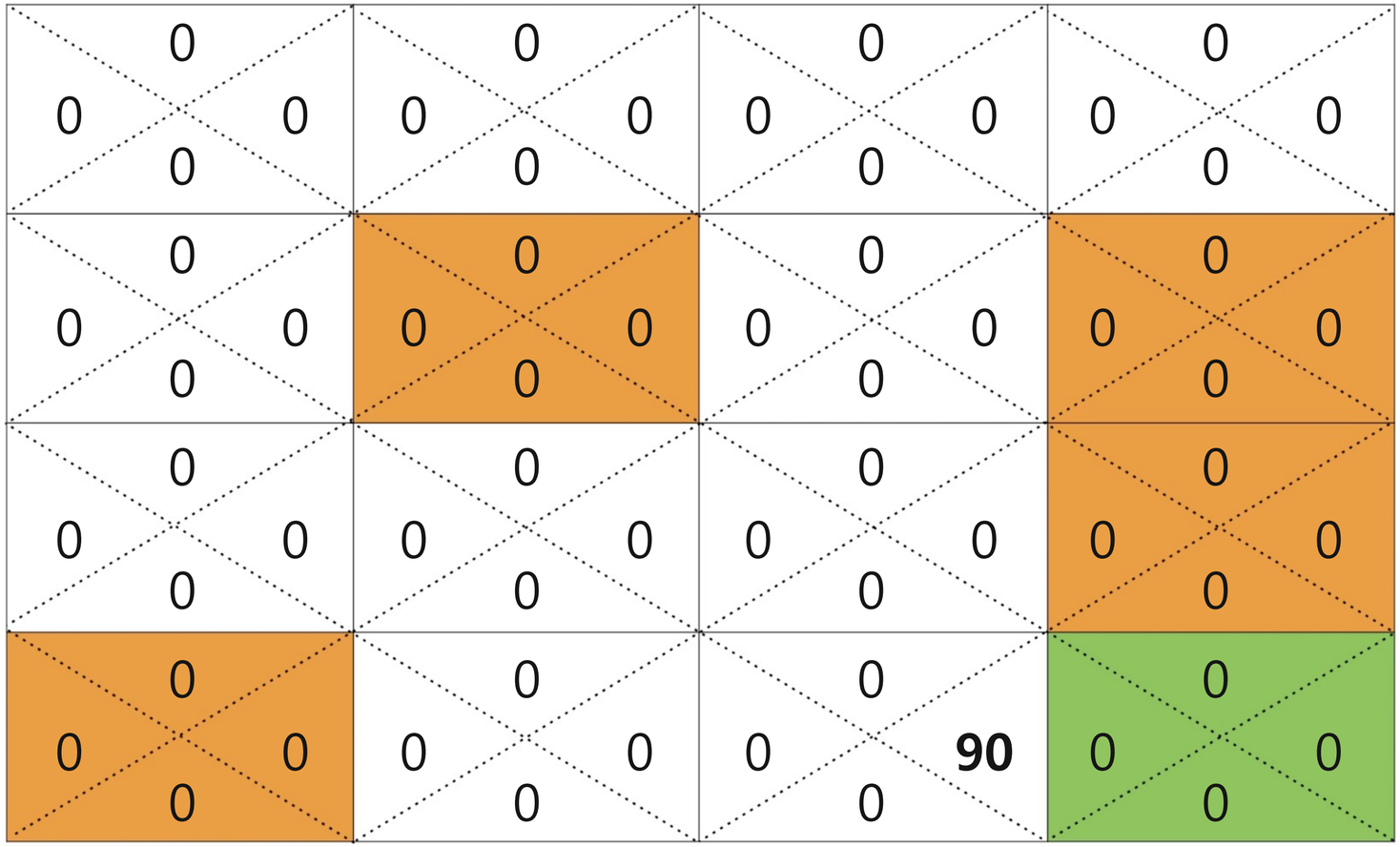
γ(할인율) = 0.9

Q(14, Right)
= r + γ max(Q(s_{15}, a))

Q(14, Right)
= 90 + 0.9*0
= 90

© Hackingcamp. All Rights Reserved

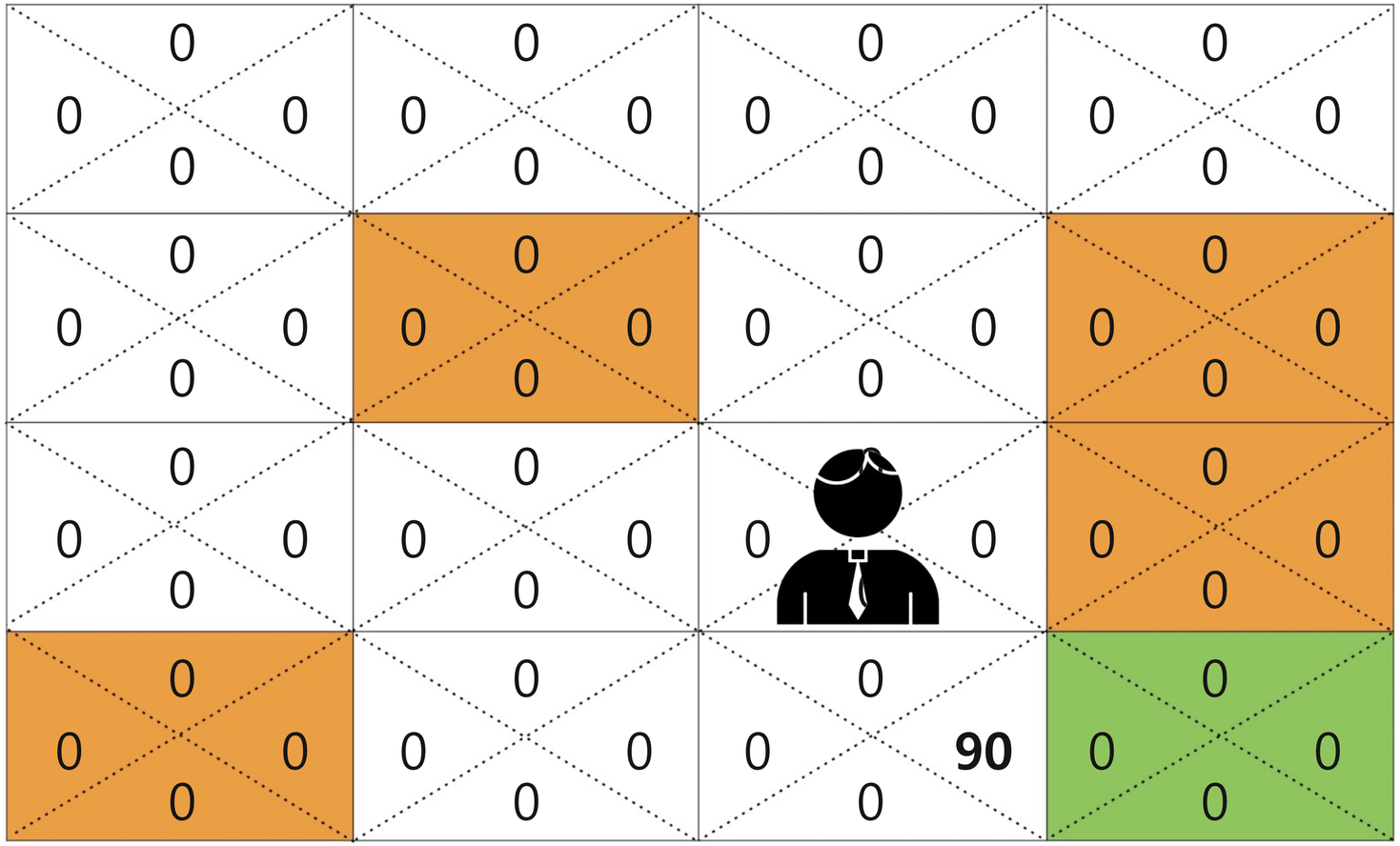
Q-Learning



© Hackingcamp. All Rights Reserved

POC SECURITY

Q-Learning



γ(할인율) = 0.9

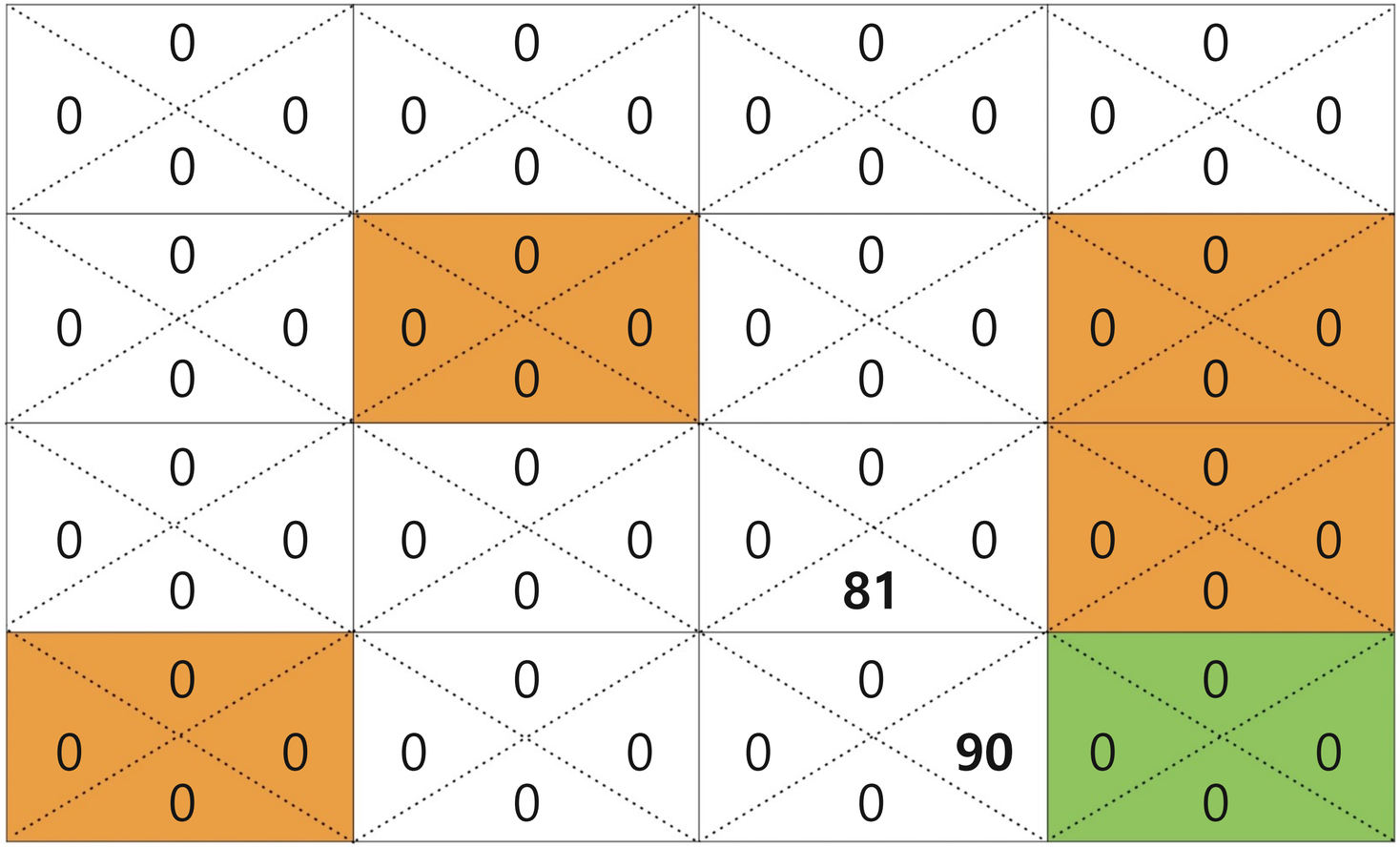
Q(10, Down)
= r + γ max(Q(s_{14}, a))

Q(10, Down)
= 0 + 0.9*90
= 81

© Hackingcamp. All Rights Reserved

POC SECURITY

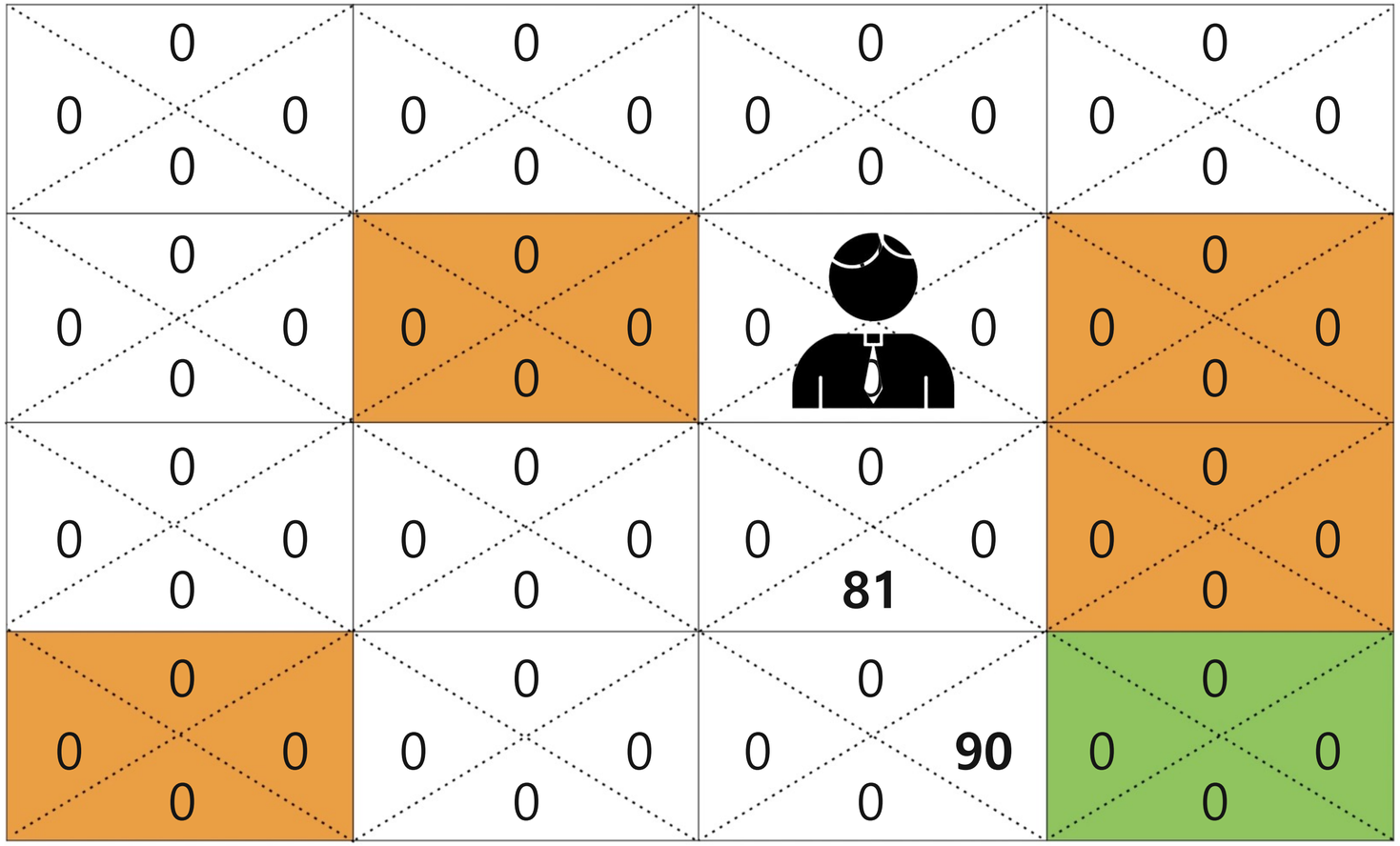
Q-Learning



© Hackingcamp. All Rights Reserved

POC SECURITY

Q-Learning



γ(할인율) = 0.9

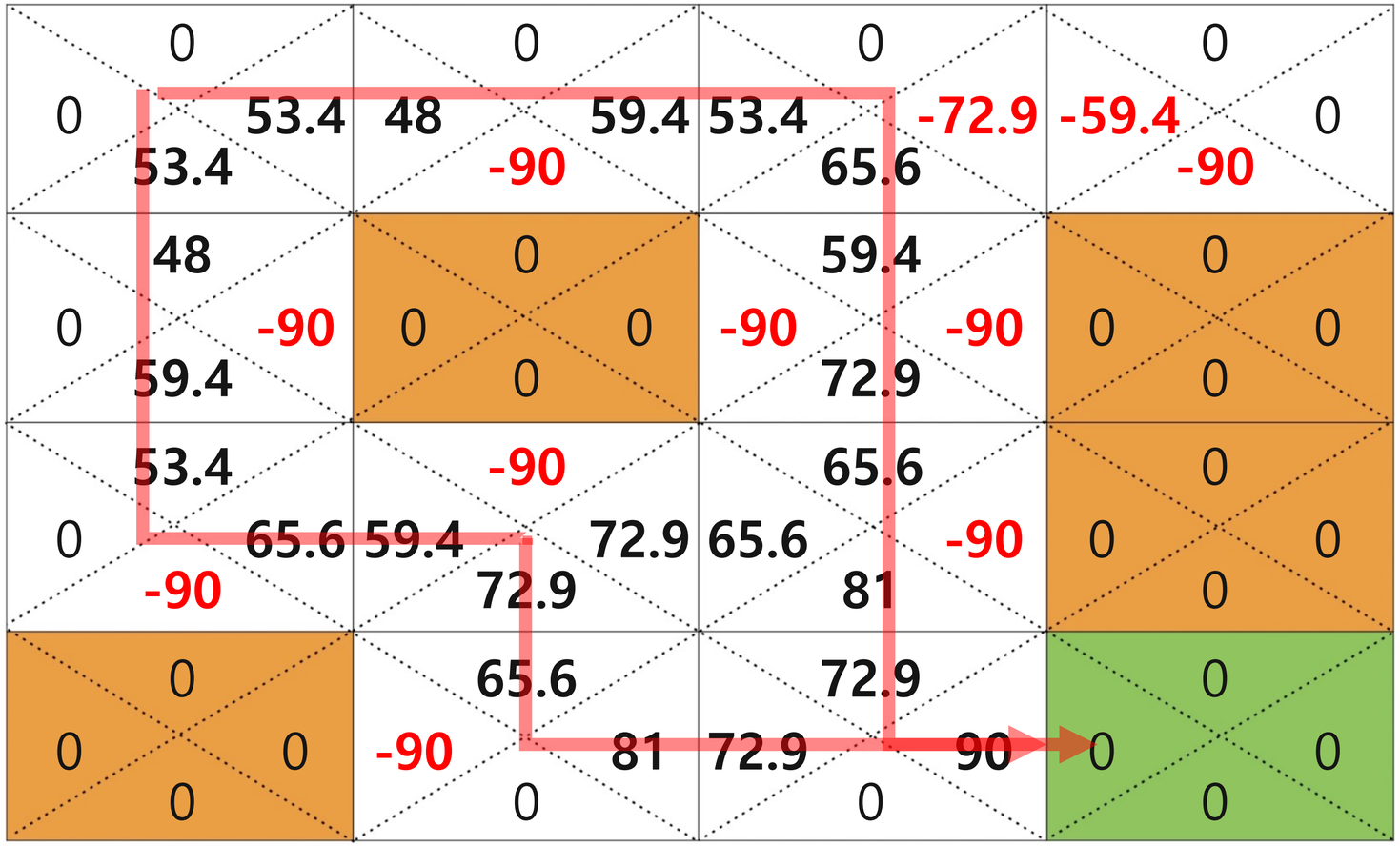
Q(10, Down)
= r + γ max(Q(s_{10}, a))

Q(10, Down)
= 0 + 0.9*81
= 72.9

© Hackingcamp. All Rights Reserved

POC SECURITY

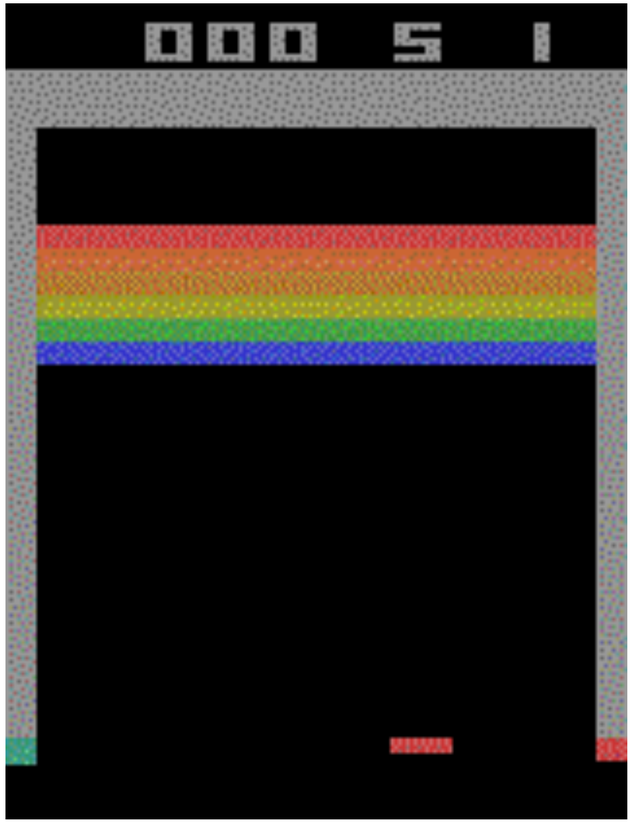
Q-Learning

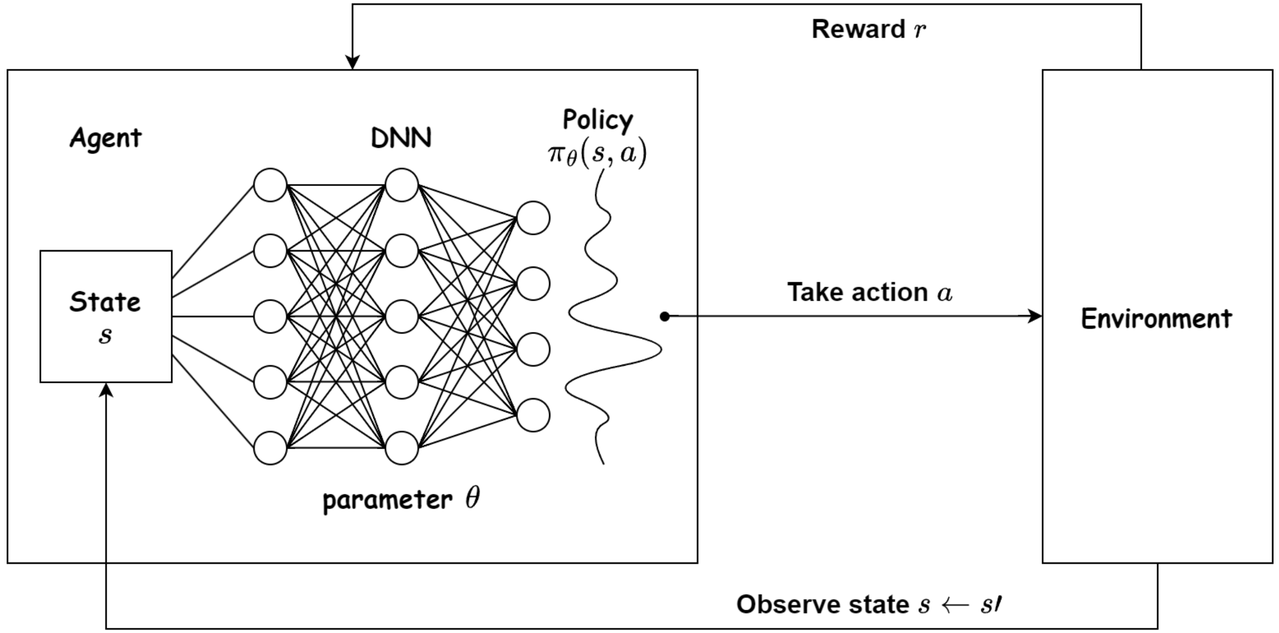


© Hackingcamp. All Rights Reserved

POC SECURITY

Q-Network

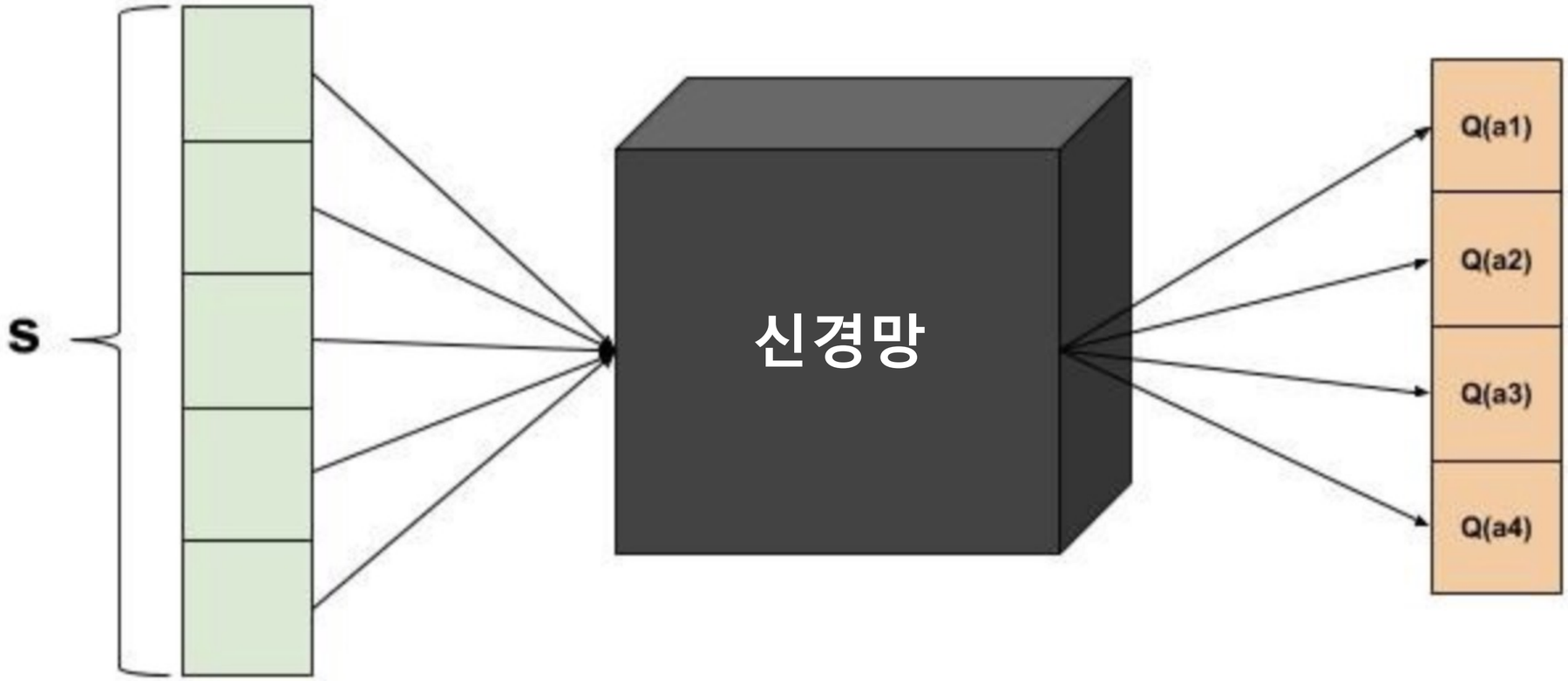




© Hackingcamp. All Rights Reserved

POC SECURITY

Q-Network



© Hackingcamp. All Rights Reserved

웹 해킹을 강화학습 문제로 정의하기

© Hackingcamp. All Rights Reserved

강화학습 기반 웹 해킹

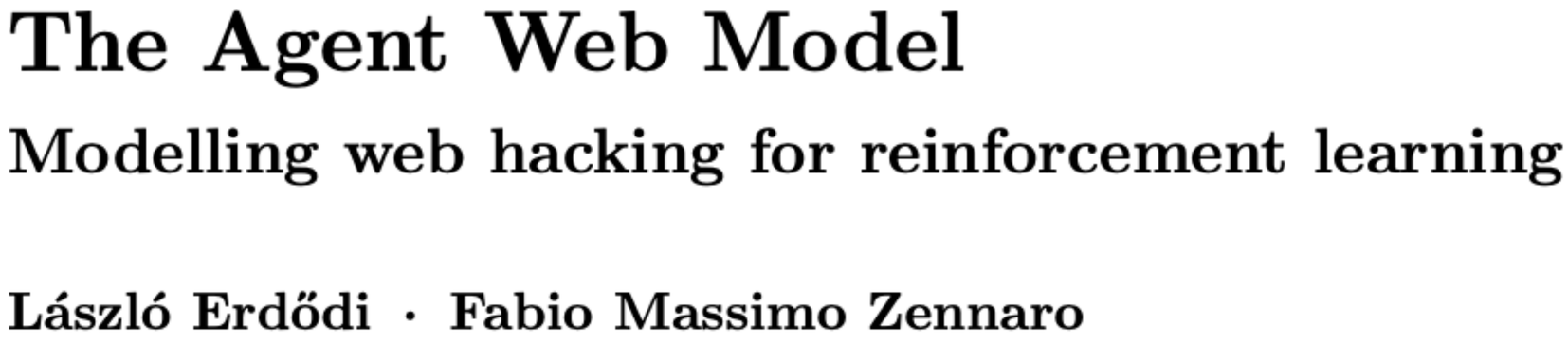


목적, 상태, 행동, 보상, 종료

© Hackingcamp. All Rights Reserved



강화학습 기반 웹 해킹



© Hackingcamp. All Rights Reserved

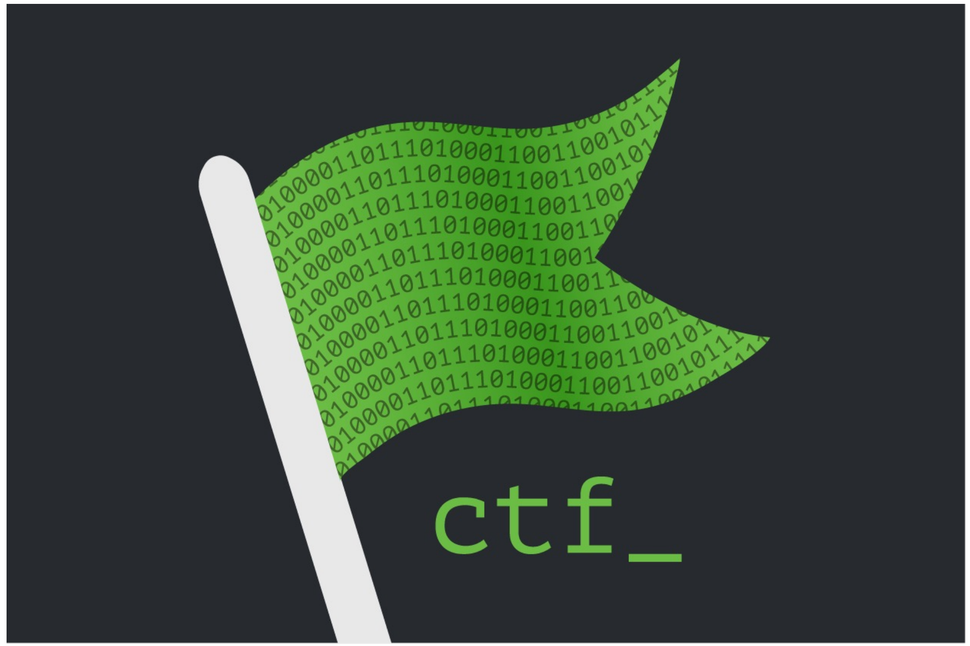
사용 라이브러리

gym은 OpenAI에서 개발한 강화학습 알고리즘 개발과 테스트를 위한 라이브러리로, 다양한 환경과 통일된 인터페이스를 제공하여 알고리즘 평가와 개발을 지원합니다.



© Hackingcamp. All Rights Reserved

강화학습 기반 웹 해킹



CTF TIME

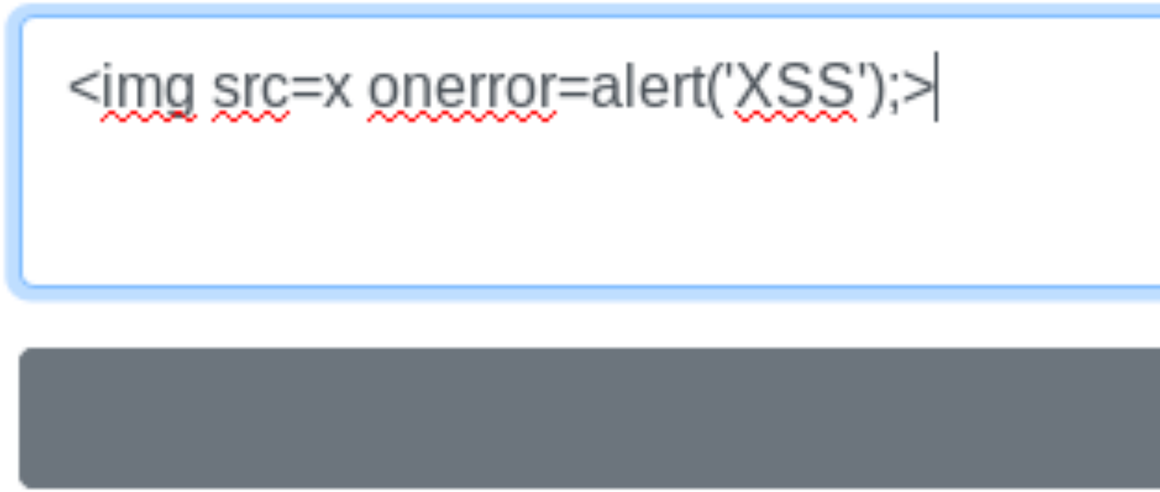
© Hackingcamp. All Rights Reserved

POC SECURITY

Purpose

- XSS(Cross-Site Scripting) 탐지

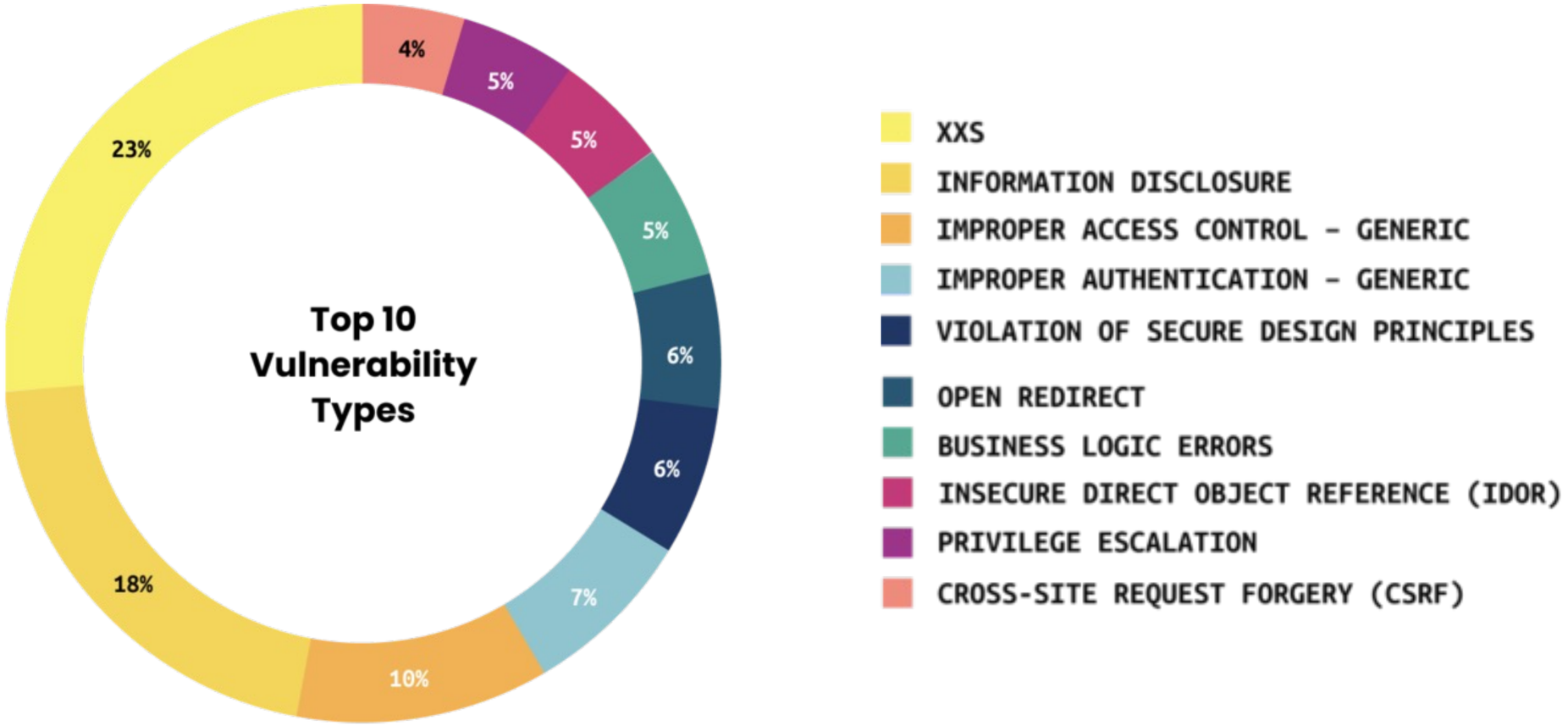
- SQL Injection 탐지



© Hackingcamp. All Rights Reserved

POC SECURITY

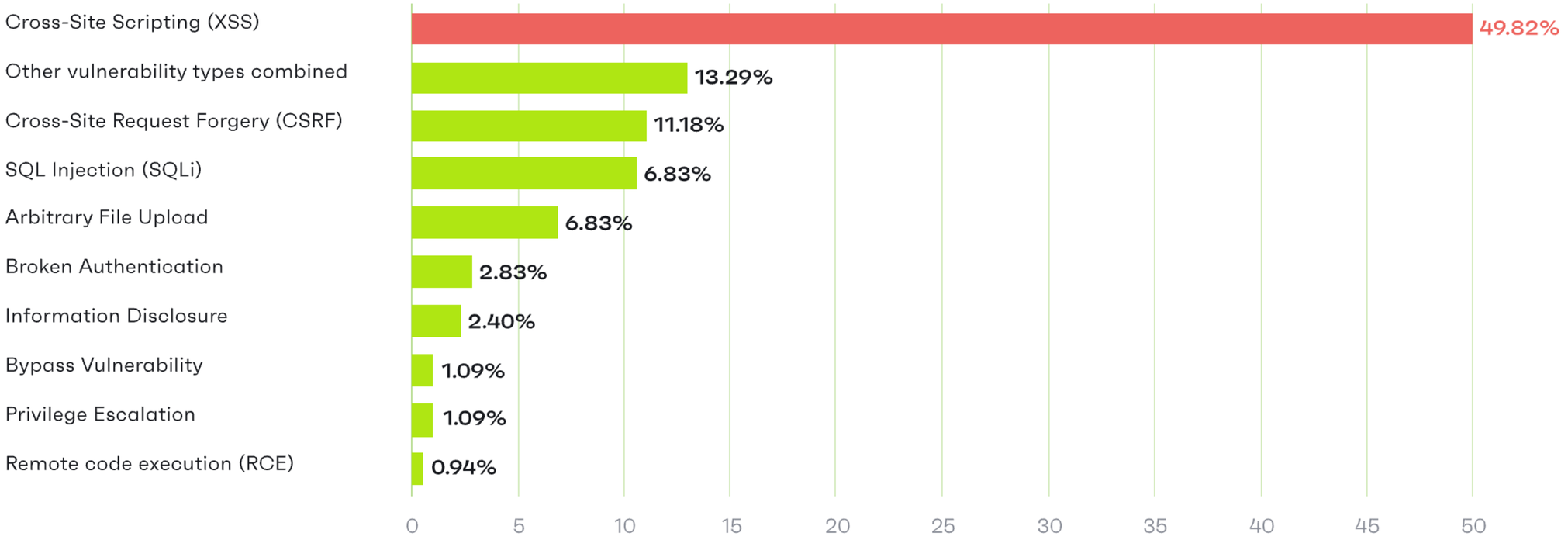
Top Ten Vulnerabilities - Hackerone



© Hackingcamp. All Rights Reserved

POC SECURITY

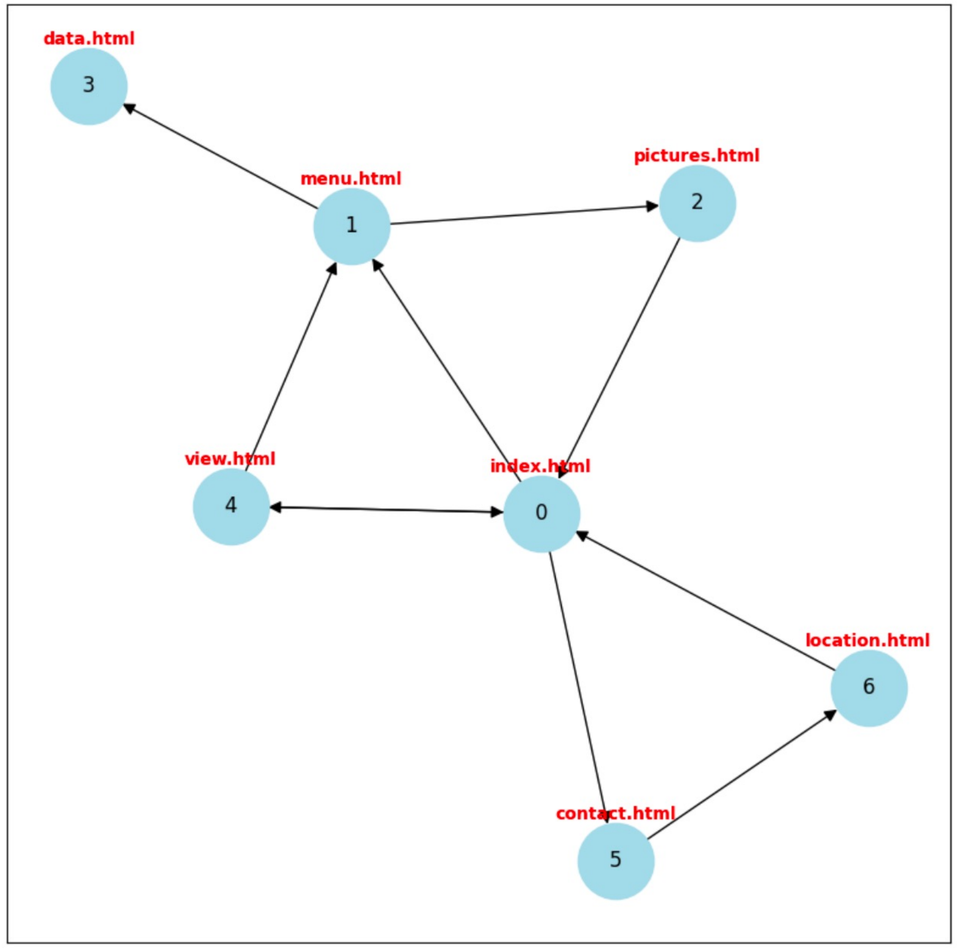
Top Ten Vulnerabilities - Patchstack



© Hackingcamp. All Rights Reserved

POC SECURITY

State



어떤 정보를 통해 취약점을 찾을 수 있을까?

- 어떤 페이지가 있는지
- 각 페이지의 파라미터
- 페이지의 URL

© Hackingcamp. All Rights Reserved

State

[1, 0, 0, 0, 0, 0, 0, ... 0]

- 1 : Found

- 0 : Not Found

© Hackingcamp. All Rights Reserved

Action

[CMD_Action, Target_Page]

- 0 (None) - 1~
- 1 (Read)
- 2 (XSS_Attack)
- 3 (SQLI_Attack)

© Hackingcamp. All Rights Reserved

Code

관찰 공간(State) 정의

```
self.observation_space = spaces.MultiBinary(self.n_files)
```

행동 공간(Action) 정의

```
self.action_space = spaces.MultiDiscrete([4,self.n_files])
```

© Hackingcamp. All Rights Reserved

Reward

- 공격 성공 시 (+100)

- 새로운 페이지 발견 시 (+5)

- 그 외 행동 (-1)

- 존재하지 않는 페이지를 타겟팅 할 경우 (-5)

© Hackingcamp. All Rights Reserved

종료 조건

- 공격 성공 시

- N초의 시간이 지나면

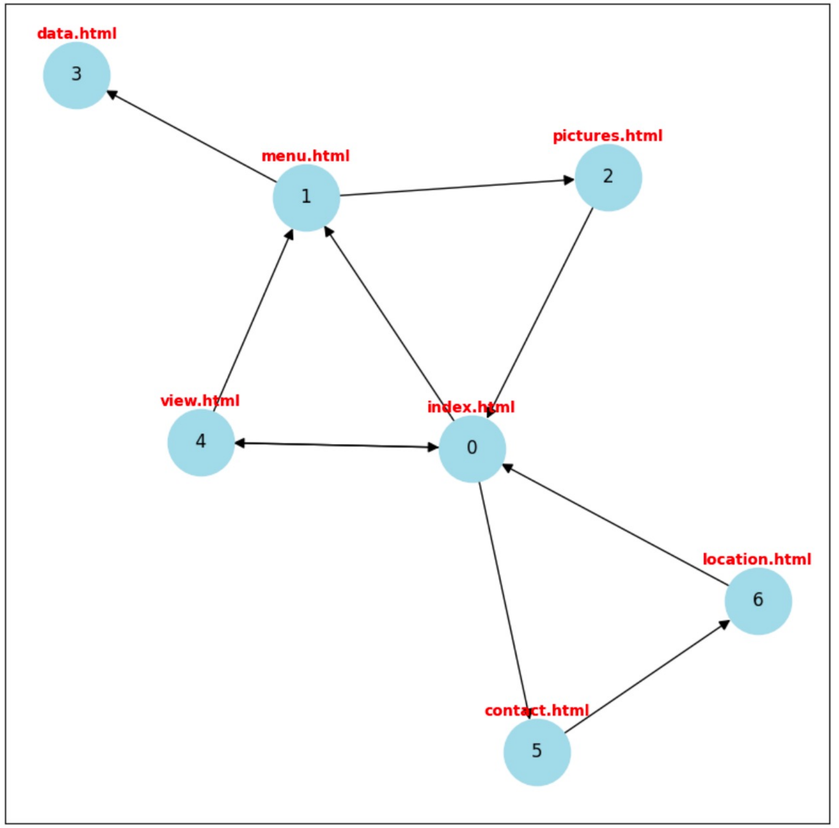
© Hackingcamp. All Rights Reserved

가상 WebServer

```
# XSS Attack
elif(command==2):
    if(targetfile==self.flag and self.vul_type == 1):
      self.done = True
      return self.state,100,self.done,{'msg':'XSS_Attack Success'}
    else:
      return self.state,-5, self.done,{'msg':'XSS_Attack Fail'}

# SQLi Attack
elif(command==3):
    if(targetfile==self.flag and self.vul_type == 2):
        self.done = True
        return self.state,100,self.done,{'msg':'SQLI_Attack Success'}
    else:
        return self.state,-1,self.done,{'msg':'SQLi_Attack Fail'}

else:
    return self.state,-1,self.done,{'msg':'Wrong command'}
```



© Hackingcamp. All Rights Reserved

POC SECURITY

Code

Reset 함수

```
def reset(self):
    self.flag = np.random.randint(self.n_files)
    self.state = np.zeros(self.n_files,dtype=np.bool)
    self.vul_type = np.random.randint(2)
    self.state[0] = 1
    self.done = False
    return self.state
```

환경이 종료된 이후 호출되는 함수로 환경을 초기화

Step 함수

```
def step(self, action):
    return self.webserver(action)
```

환경과 상호작용하는 함수로 보상, 상태, 종료 여부 등이 반환

© Hackingcamp. All Rights Reserved

가상 WebServer - 학습 시키기




```
gym_env = AWM()

env = DummyVecEnv([lambda: gym_env])
modelPPO = PPO("MlpPolicy", env, verbose=1, tensorboard_log="./logs/")

modelPPO.learn(total_timesteps=100000)
modelPPO.save("PPO_Agent")

del modelPPO
```

© Hackingcamp. All Rights Reserved

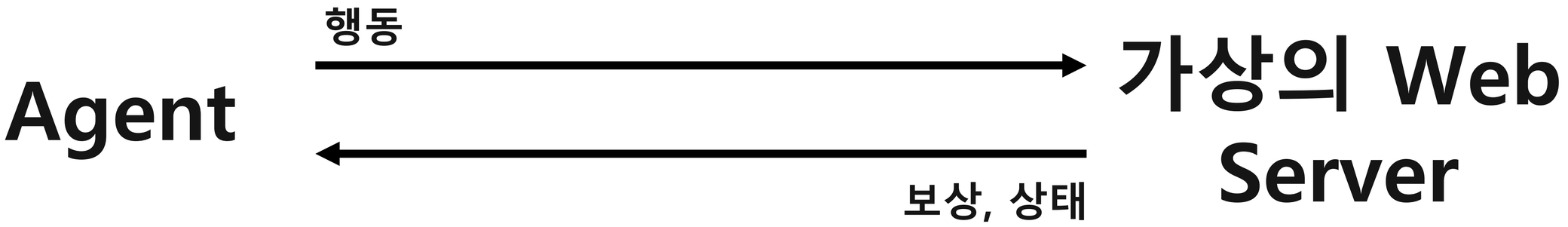
가상 WebServer - 학습 시키기




```
================================================================================
Action : SQLI_Attack
Target Page 2
State [ True  True False  True False]
Total Reward -65
================================================================================
Action : SQLI_Attack
Target Page 0
State [ True  True False  True False]
Total Reward -66
================================================================================
Action : SQLI_Attack
Target Page 4
State [ True  True False  True False]
Total Reward -71
================================================================================
Action : None
Target Page 4
State [ True  True False  True False]
Total Reward -76
================================================================================
Action : XSS_Attack
Target Page 1
State [ True  True False  True False]
Total Reward 24
```

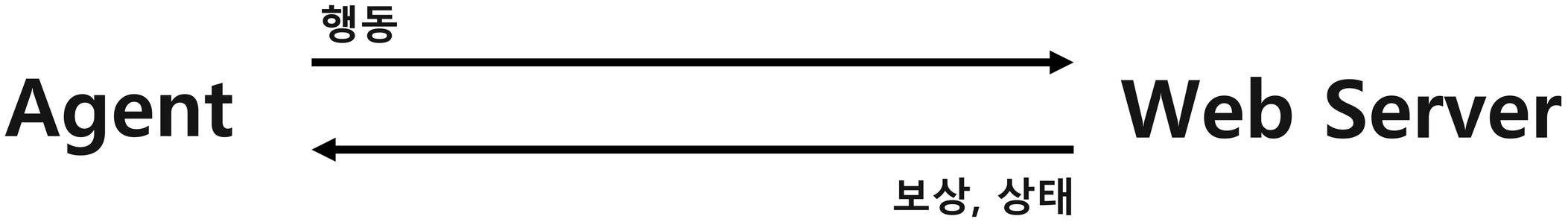
© Hackingcamp. All Rights Reserved

작동 과정 요약



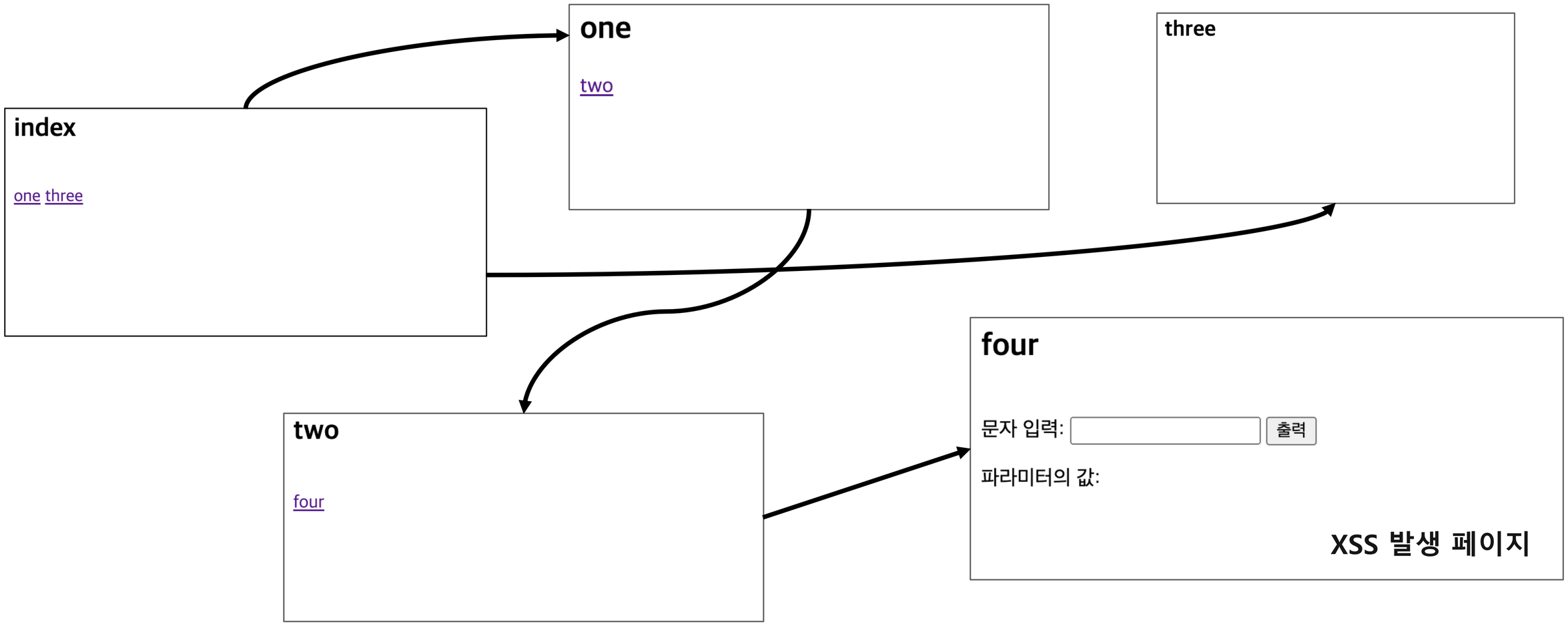
© Hackingcamp. All Rights Reserved

작동 과정 요약



© Hackingcamp. All Rights Reserved

웹 페이지 설계



© Hackingcamp. All Rights Reserved

셀레리움 - Read

```
def crawling(url, cookie=None):
    driver.get(url)
    if cookie != None:
        driver.add_cookie(cookie)
    time.sleep(1)
    soup = BeautifulSoup(driver.page_source)
    return soup
```

© Hackingcamp. All Rights Reserved

셀레리움 - Read




```
def find_sub_page(now, text):
    sub_page = []

    for link in text.find_all('a'):
        href = link.get('href')

        if href == None or len(href) == 0 or href[0] == '#': continue # url이 #이거나 없으면 패스
        if urlparse(href).scheme != 'http' and urlparse(href).scheme != 'https' and urlparse(href).scheme != '': continue

        href = pre_href(now, href) # 절대 경로, 상대 경로 모두 도메인 붙여서 변경

        if urlparse(href).netloc != attack_domain: continue # 다른 도메인이면 패스
        if attack_path != '': # 만약 경로 스코프가 존재한다면 검사
            if attack_path not in urlparse(href).path: continue

        sub_page.append(href)

    return sub_page
```

© Hackingcamp. All Rights Reserved

셀레리움 - Attack

```python
def attack(data, pl, num, cookie):
    pl = pl.replace('{mes}', str(num)).strip() # 페이로드 완성
    get_payload = make_payload(data[1], pl)

    result = [f'alert({num})', data[0] + get_payload, []]

    url = data[0] + get_payload # 공격 url 생성
    try:
        crawling(url, cookie)
    except:
        print(f"error with {url}")
        result.append('')
        return result

    try:
        alert = Alert(driver)
        alert_txt = alert.text
        if ord('0') <= ord(alert_txt) and ord(alert_txt) <= ord(str(num)):
            print("Vulnerability Found :", url)
            result.append(alert_txt)
        while True:
            try:
                alert.dismiss()
            except:
                break
    except:
        result.append('')

    return result
```

© Hackingcamp. All Rights Reserved

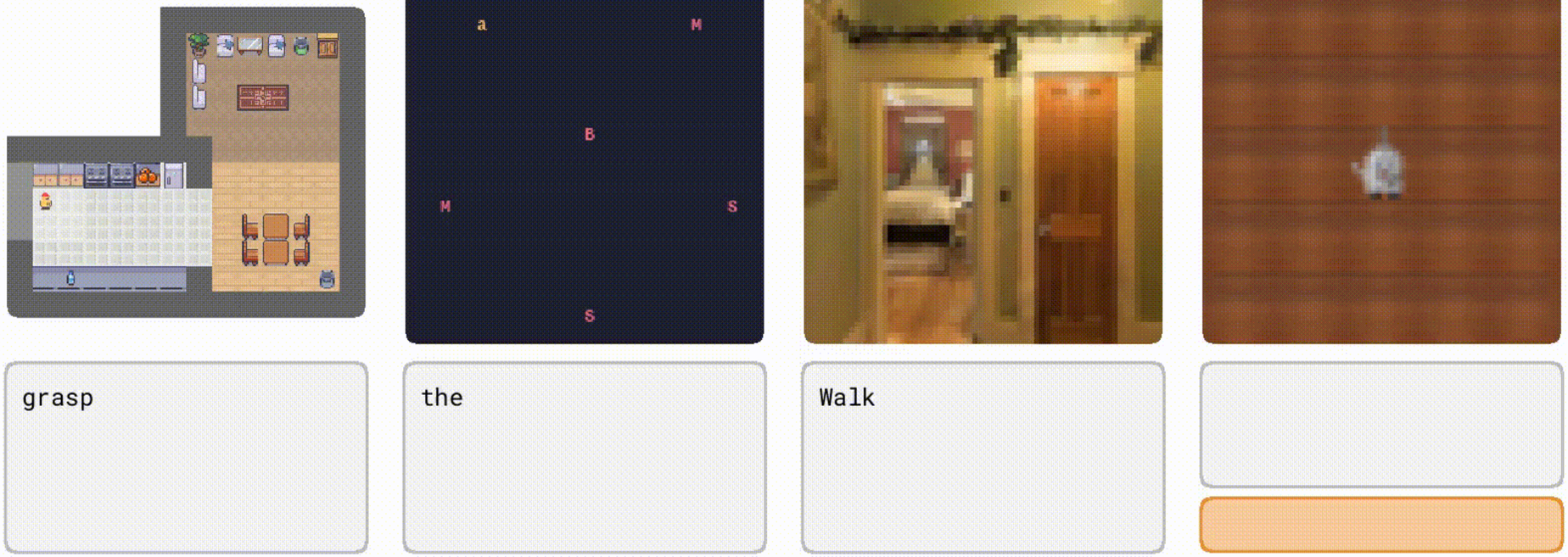
결과

```
(venv) ✘ Otwooo ~/Documents/Project/AI_Web_Fuzzer  main ± python -u "/Users/baesonghyeon/Documents/Project/AI_Web_Fuzzer/realword_env.py"
Read.. http://localhost:8888/
XSS Attack.. http://localhost:8888/one.php
Read.. http://localhost:8888/one.php
XSS Attack.. http://localhost:8888/
XSS Attack.. http://localhost:8888/three.php
Read.. http://localhost:8888/two.php
XSS Attack.. http://localhost:8888/four.php
================================================================================
Vulnerability Found :  "http://localhost:8888/four.php?text=<script>alert(1)</script>" !!!!
Step : 18
================================================================================
```

© Hackingcamp. All Rights Reserved

POC SECURITY

추후 해볼 것들



https://dynalang.github.io/

© Hackingcamp. All Rights Reserved

Reference

[1] 모두를 위한 머신러닝/딥러닝 강의 – 강화학습

[2] The Agent Web Model -- Modelling web hacking for reinforcement learning

[3] Deep Reinforcement Learning for Cyber Security

© Hackingcamp. All Rights Reserved

퀴즈 1

Q. 인간처럼 행동하는 인공지능을 만들기 위해서는 다음 중 어떤 종류의 알고리즘을 사용해야할까?

1. 지도학습　2. 비지도학습　3. 강화학습

© Hackingcamp. All Rights Reserved

퀴즈 1

Q. 인간처럼 행동하는 인공지능을 만들기 위해서는 다음 중 어떤 종류의 알고리즘을 사용해야할까?

1. 지도학습　2. 비지도학습　3. 강화학습

© Hackingcamp. All Rights Reserved

퀴즈 2

Q. 제 이름은?

© Hackingcamp. All Rights Reserved

퀴즈 2

Q. 제 이름은?

배송현

© Hackingcamp. All Rights Reserved

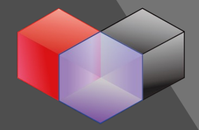


감사합니다.

QnA

배송현 Otwooo

© Hackingcamp. All Rights Reserved



이후에 질문이 있다면?

인스타: @bshyun05
디코: otwooo



© Hackingcamp. All Rights Reserved

POC SECURITY