

Web Fuzzer Tool

2025 INCOGNITO CONFERENCE

PRESENTER | Ping! 임승연

EMAIL | Insndus@gmail.com

INDEX.

1 Intro.

1. 계기
2. OWASP Top 10.

2 퍼징 (Fuzzing)

1. 소개
2. 동작 원리
3. 대표적인 웹 퍼징 도구 소개
4. 웹 퍼징 도구의 활용 예시

3 Web Fuzzing Tool

1. 개발 과정
2. 주요 기능
3. 시연 영상

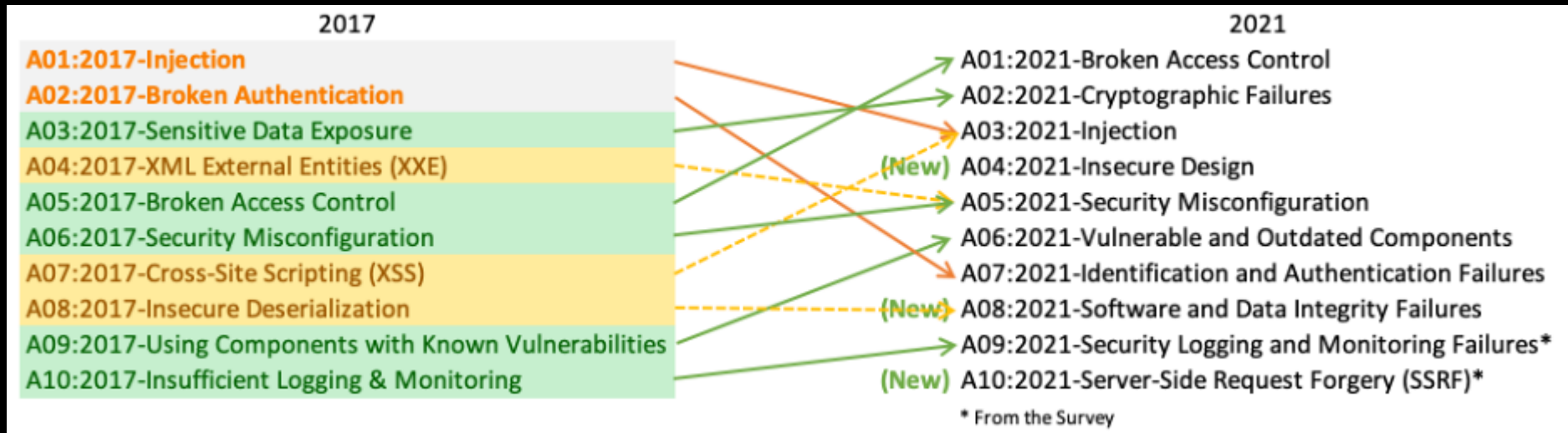
Intro.

계기

- 원하는 수준의 탐지와 자동화된 분석을 통해 웹 취약점을 알 수 있는 퍼징 도구의 필요성

OWASP Top 10.

- Open Web Application Security Project
- 웹 애플리케이션에서 자주 발생하는 보안 취약점 10가지 발표



Intro.

OWASP Top 10. (2021)

순위	항목명	주요 변화 및 설명
1	Broken Access Control	5위 → 1위, 접근 권한 통제 실패
2	Cryptographic Failures	명칭 변경, 암호화 실패
3	Injection	XSS 포함, Injection 전반
4	Insecure Design	신규, 설계 단계에서의 취약점
5	Security Misconfiguration	XEE 포함, 설정 오류
6	Vulnerable and Outdated Components	명칭 변경, 오래된 컴포넌트 사용
7	Identification and Authentication Failures	명칭 변경, 인증/식별 실패
8	Software and Data Integrity Failures	신규, 무결성 검증 실패
9	Security Logging and Monitoring Failures	이름 변경, 로깅/모니터링 실패
10	Server-Side Request Forgery (SSRF)	신규, SSRF 취약점

Intro.

웹 취약점 탐지 방법

1. 동적 분석

애플리케이션을 실행하며 보안 취약점을 탐지
ex. Fuzzing

2. 정적 분석

소스코드나 바이너리 파일을 실행하지 않고 코드 자체를 분석하여 취약점을 탐지
ex. SAST (Static Application Security Testing) 도구 : SonarQube, Find Security Bugs



퍼징 (Fuzzing)

소개

- 다양한 방식으로 입력값을 변형하거나 생성
→ 사람이 찾기 어려운 예외 상황을 자동화로 빠르게 탐색

테스트 접근 방식에 따른 퍼저 분류

구분	정보 접근성	설명	예시
블랙박스	내부 정보, 소스코드 정보 X	실제 공격 시나리오와 유사	zzuf, Radamsa
화이트박스	내부 정보 모두 활용	깊이 있는 취약점 탐지	SAGE, KLEE
그레이박스	제한된 내부 정보	현실적, 효과적인 탐지	AFL, LibFuzzer, Honggfuzz

퍼징 (Fuzzing)

1. Mutation-based Fuzzing

- 기존의 정상 입력값을 변형
ex. 문자열의 일부, 길이 변형
- AFL(American Fuzzy Lop)
 - 입력값 변형 시 코드 커버리지 정보를 분석
 - 새로운 경로 탐색이 가능한 입력을 우선적으로 선택
- <https://www.fuzzingbook.org/> 에서 실습



퍼징 (Fuzzing)

2. Generation-based Fuzzing

- 입력값의 구조나 문법을 미리 정의
 - 규칙에 따라 체계적으로 testcase를 생성
 - ex. C 컴파일러를 퍼징 시, C언어의 문법에 맞는 코드 자동 생성
- 문법 기반 화이트박스 퍼징의 필요성
 - 심볼릭 실행, 제약 조건 해결로 다양한 경로 탐색
 - 입력이 복잡한 프로그램에서의 어려움
- Godefroid의 논문 : Grammer-based whitebox fuzzing
 - 문법 기반 제약 조건 생성
 - 복잡한 입력 구조를 가진 프로그램에서도 효과적으로 탐지 가능

퍼징 (Fuzzing)

Fuzzer의 동작 원리

1. 입력 지점 식별

웹 애플리케이션의 입력 지점 식별

2. 입력값 생성 및 변조

페이로드를 생성해 특정 파라미터나 필드에 주입

3. 요청 및 실행

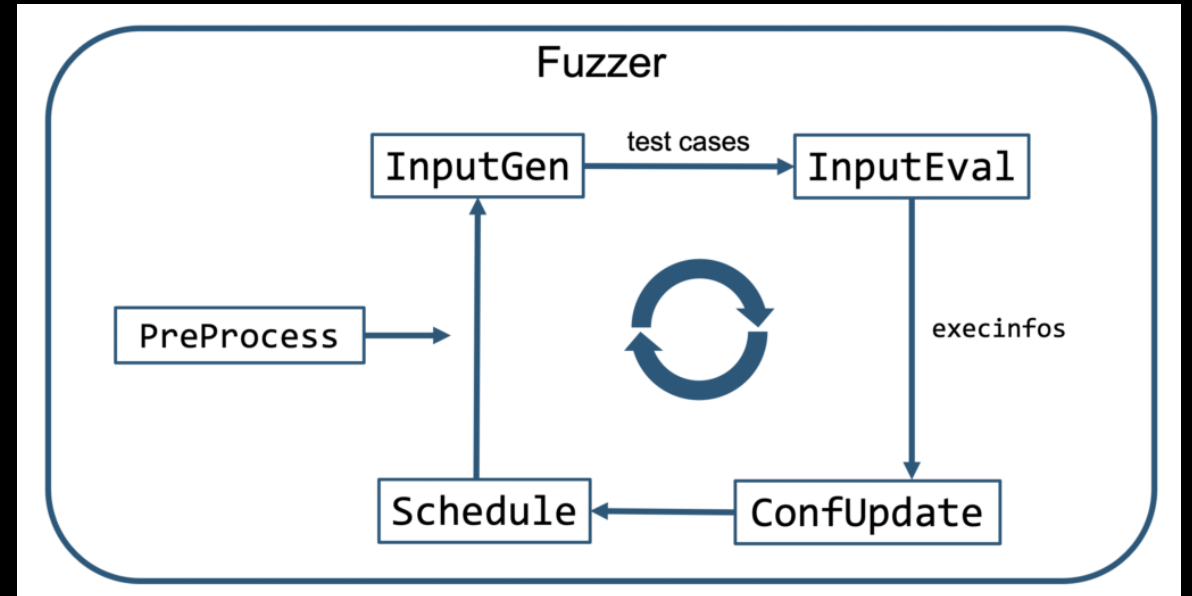
변조된 입력 값으로 요청을 보낸 뒤, 실행 결과 수집

4. 응답 및 동작 분석

서버의 응답 분석 → 분석 데이터 탐지

5. 자동 반복 및 피드백

반복적으로 자동화, 결과를 바탕으로 개선



퍼징 (Fuzzing)

Tool 소개

1. OWASP ZAP

- 오픈소스 기반의 웹 취약점 진단 도구
- GUI 기반, 다양한 플러그인 지원



2. Burp Suite (Intruder / Scanner)

- Intruder 기능 : 특정 파라미터에 페이로드 주입
- Scanner 기능 : 자동화된 취약점 진단



3. wfuzz

- CLI 기반의 오픈소스 퍼징 도구
- 다양한 위치에 wordlist를 주입하여 퍼징



4. Ffuf

- CLI 기반의 오픈소스 도구
- 숨겨진 디렉토리 및 파일 탐지



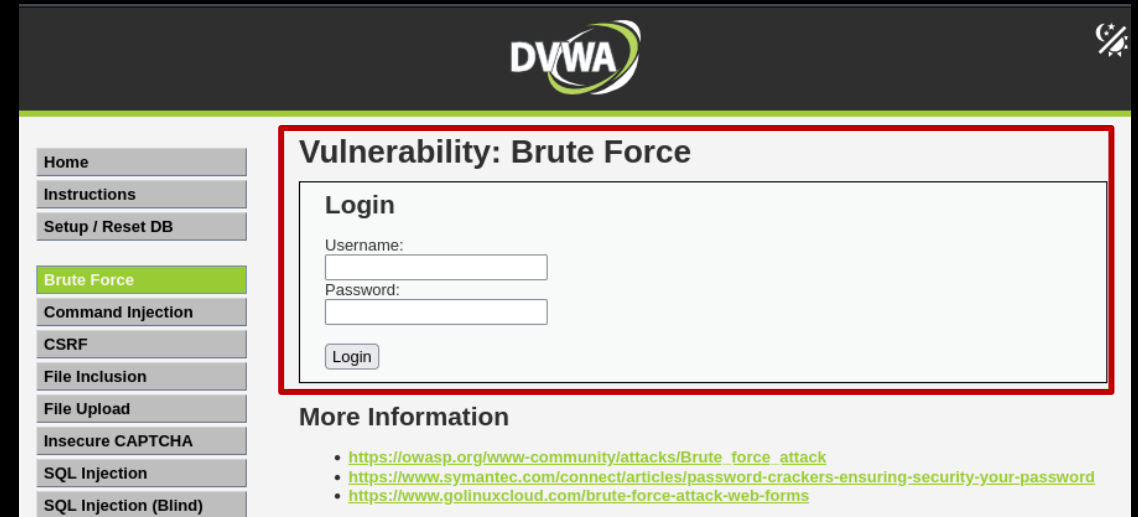
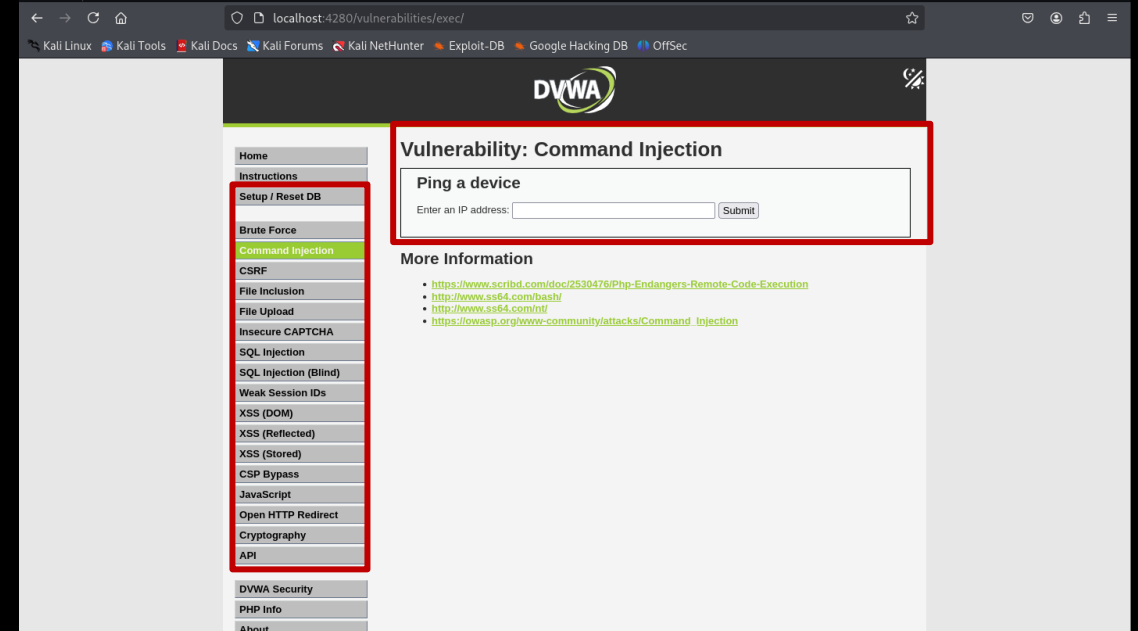
퍼징 (Fuzzing)

DVWA

- Damn Vulnerable Web Application
→ 매우 취약한 PHP/MySQL 웹 애플리케이션

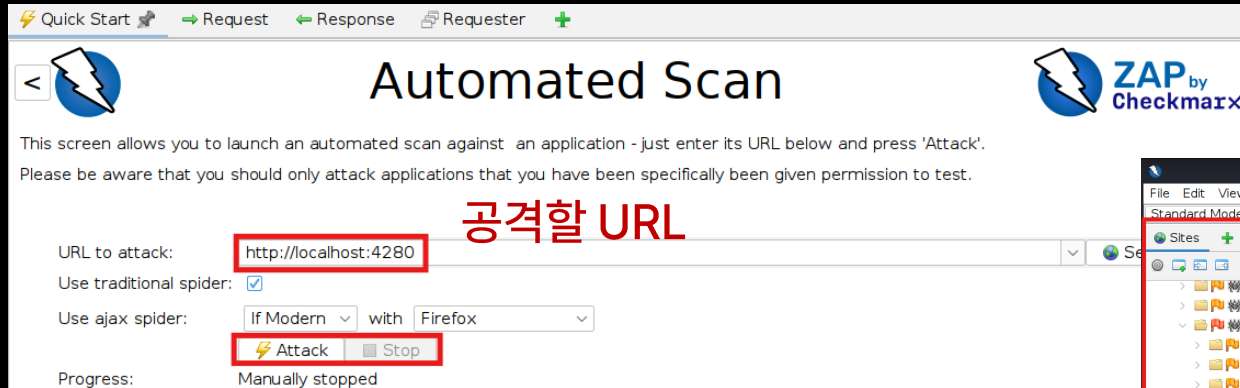
사용 대상

- 보안인 : 법적 환경에서 자신의 기술과 도구를 테스트
- 웹 개발자 : 보안 프로세스 이해
- 보안을 배우고자 하는 모든 분들



퍼징 (Fuzzing)

OWASP ZAP



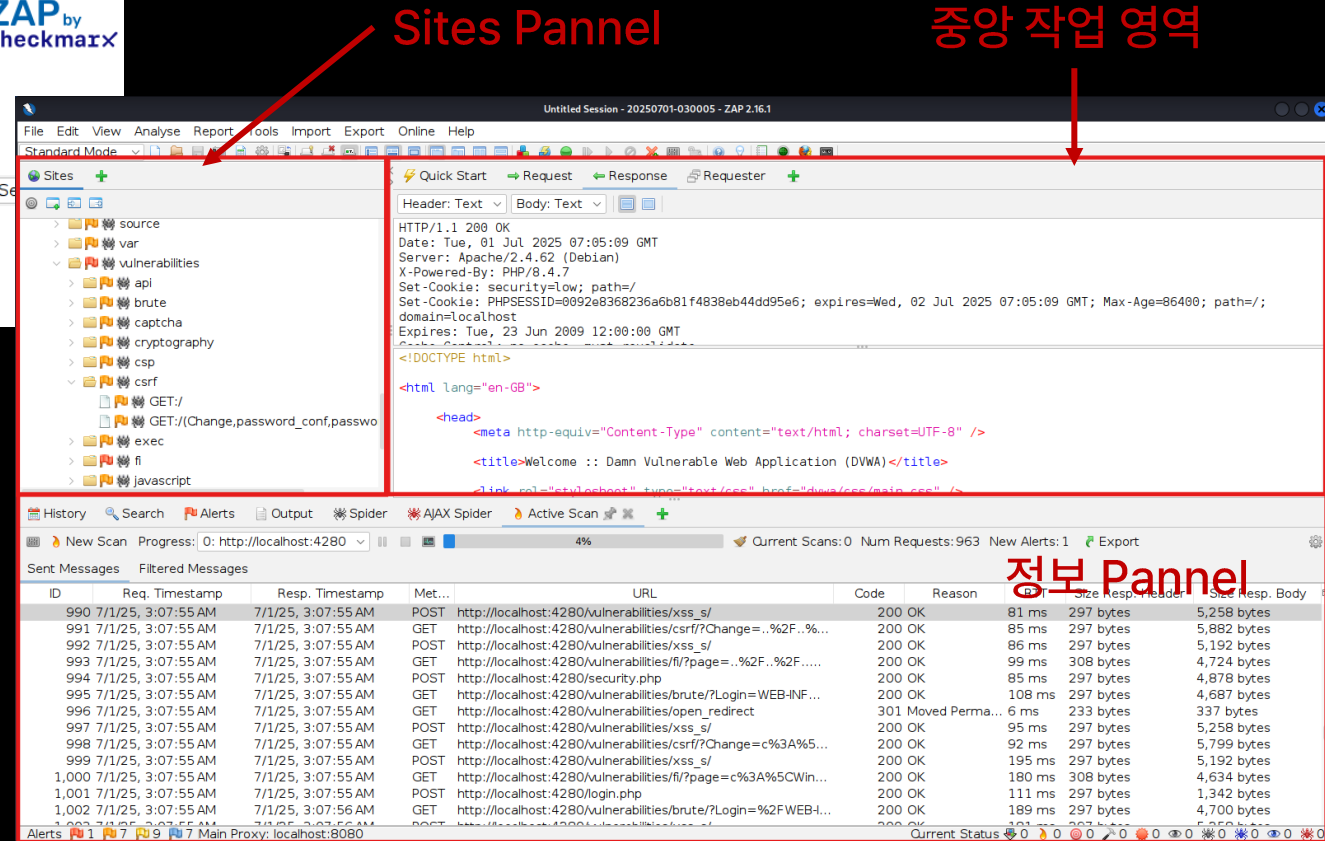
공격할 URL

1. Automated scan 설정 화면

- 크롤링
- 취약점 진단

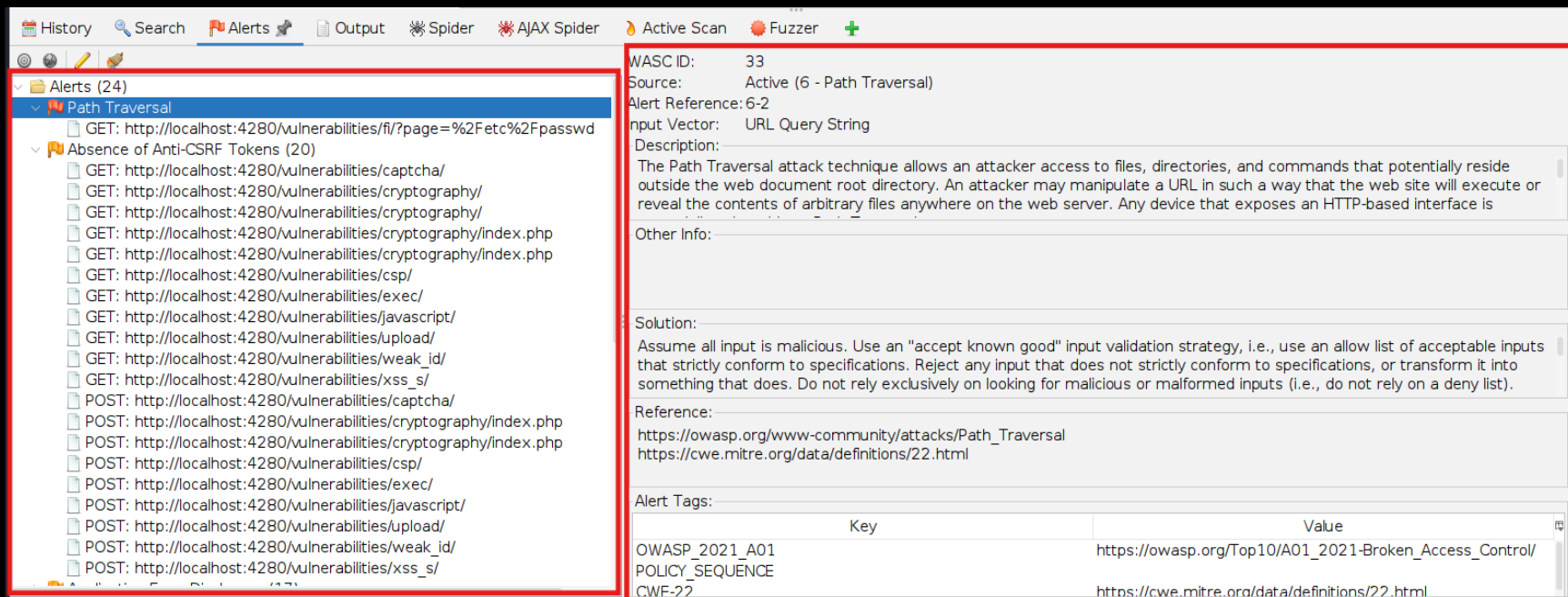
2. 스캔이 진행된 분석 화면

- Sites : 분석한 서버의 디렉토리 와 취약점 메뉴
- 중앙 작업 영역 : HTTP 헤더, 응답 본문, 코드
- 정보 패널 : 요청, 응답 코드, History/Alerts/Spider/Active Scan/Fuzzer 탭



퍼징 (Fuzzing)

OWASP ZAP – Alerts(취약점 경고) 패널



The screenshot shows the OWASP ZAP Alerts panel. The left sidebar lists 24 alerts, with 'Path Traversal' selected. The main panel displays details for the 'Path Traversal' alert, including its WASC ID (33), Source (Active (6 - Path Traversal)), Alert Reference (6-2), Input Vector (URL Query String), and a detailed Description. It also provides a Solution, Reference links, and Alert Tags.

Alerts (24)

- Path Traversal
 - GET: http://localhost:4280/vulnerabilities/fi/?page=%2Fetc%2Fpasswd
 - Absence of Anti-CSRF Tokens (20)
 - GET: http://localhost:4280/vulnerabilities/captcha/
 - GET: http://localhost:4280/vulnerabilities/cryptography/
 - GET: http://localhost:4280/vulnerabilities/cryptography/index.php
 - GET: http://localhost:4280/vulnerabilities/cryptography/index.php
 - GET: http://localhost:4280/vulnerabilities/csp/
 - GET: http://localhost:4280/vulnerabilities/exec/
 - GET: http://localhost:4280/vulnerabilities/javascript/
 - GET: http://localhost:4280/vulnerabilities/upload/
 - GET: http://localhost:4280/vulnerabilities/weak_id/
 - GET: http://localhost:4280/vulnerabilities/xss_s/
 - POST: http://localhost:4280/vulnerabilities/captcha/
 - POST: http://localhost:4280/vulnerabilities/cryptography/index.php
 - POST: http://localhost:4280/vulnerabilities/cryptography/index.php
 - POST: http://localhost:4280/vulnerabilities/csp/
 - POST: http://localhost:4280/vulnerabilities/exec/
 - POST: http://localhost:4280/vulnerabilities/javascript/
 - POST: http://localhost:4280/vulnerabilities/upload/
 - POST: http://localhost:4280/vulnerabilities/weak_id/
 - POST: http://localhost:4280/vulnerabilities/xss_s/

Alert Details:

- WASC ID: 33
- Source: Active (6 - Path Traversal)
- Alert Reference: 6-2
- Input Vector: URL Query String
- Description: The Path Traversal attack technique allows an attacker access to files, directories, and commands that potentially reside outside the web document root directory. An attacker may manipulate a URL in such a way that the web site will execute or reveal the contents of arbitrary files anywhere on the web server. Any device that exposes an HTTP-based interface is
- Other Info:
- Solution: Assume all input is malicious. Use an "accept known good" input validation strategy, i.e., use an allow list of acceptable inputs that strictly conform to specifications. Reject any input that does not strictly conform to specifications, or transform it into something that does. Do not rely exclusively on looking for malicious or malformed inputs (i.e., do not rely on a deny list).
- Reference: https://owasp.org/www-community/attacks/Path_Traversal
<https://cwe.mitre.org/data/definitions/22.html>
- Alert Tags:

Key	Value
OWASP_2021_A01	https://owasp.org/Top10/A01_2021-Broken_Access_Control/
POLICY_SEQUENCE	
CWE-22	https://cwe.mitre.org/data/definitions/22.html

Alerts

- 탐지된 취약점 목록
 - 취약점이 발견된 URL
 - 상세 설명
 - 대응 방안
 - 참고 링크

퍼징 (Fuzzing)

Ffuf (Fuzz Faster u Fool)

```
ffuf -u http://ffuf.me/cd/basic/FUZZ -w /usr/share/wordlists/wfuzz/general/common.txt
```

```
(kali@kali)-[~]  
$ ffuf -u http://localhost:4280/vulnerabilities/FUZZ -w /usr/share/wordlists/wfuzz/general/common.txt  
  
v2.1.0-dev  
  
:: Method      : GET  
:: URL         : http://localhost:4280/vulnerabilities/FUZZ  
:: Wordlist     : FUZZ: /usr/share/wordlists/wfuzz/general/common.txt  
:: Follow redirects : false  
:: Calibration : false  
:: Timeout     : 10  
:: Threads     : 40  
:: Matcher     : Response status: 200-299,301,302,307,401,403,405,500  
  
api      [Status: 301, Size: 327, Words: 20, Lines: 10, Duration: 4ms]  
captcha  [Status: 301, Size: 331, Words: 20, Lines: 10, Duration: 3ms]  
exec     [Status: 301, Size: 328, Words: 20, Lines: 10, Duration: 4ms]  
javascript [Status: 301, Size: 334, Words: 20, Lines: 10, Duration: 3ms]  
upload   [Status: 301, Size: 330, Words: 20, Lines: 10, Duration: 3ms]  
:: Progress: [951/951] :: Job [1/1] :: 39 req/sec :: Duration: [0:00:05] :: Errors: 0 ::
```

명령어 해석

- -u [URL]/FUZZ
URL의 FUZZ에 wordlist
단어들이 대입
- -w [wordlist 경로]

퍼징 (Fuzzing)

Ffuf (Fuzz Faster u Fool)

```
Home
v2.1.0-dev

:: Method      : GET
:: URL         : http://localhost:4280/vulnerabilities/FUZZ
:: Wordlist    : FUZZ: /usr/share/wordlists/wfuzz/general/common.txt
:: Follow redirects : false
:: Calibration : false
:: Timeout     : 10
:: Threads    : 40
:: Matcher     : Response status: 200-299,301,302,307,401,403,405,500

api      [Status: 301, Size: 327, Words: 20, Lines: 10, Duration: 4ms]
captcha  [Status: 301, Size: 331, Words: 20, Lines: 10, Duration: 3ms]
exec     [Status: 301, Size: 328, Words: 20, Lines: 10, Duration: 4ms]
javascript [Status: 301, Size: 334, Words: 20, Lines: 10, Duration: 3ms]
upload   [Status: 301, Size: 330, Words: 20, Lines: 10, Duration: 3ms]
:: Progress: [951/951] :: Job [1/1] :: 39 req/sec :: Duration: [0:00:05] :: Errors: 0 ::
```

퍼징 결과

- 발견된 유효 경로
→ api, captcha, exec, javascript, upload
- 상태코드 / 응답 크기 / 단어 수 / 줄 수 / 응답 시간
→ 301 : 리다이렉트 응답 반환

Web Fuzzer Tool

개발 과정

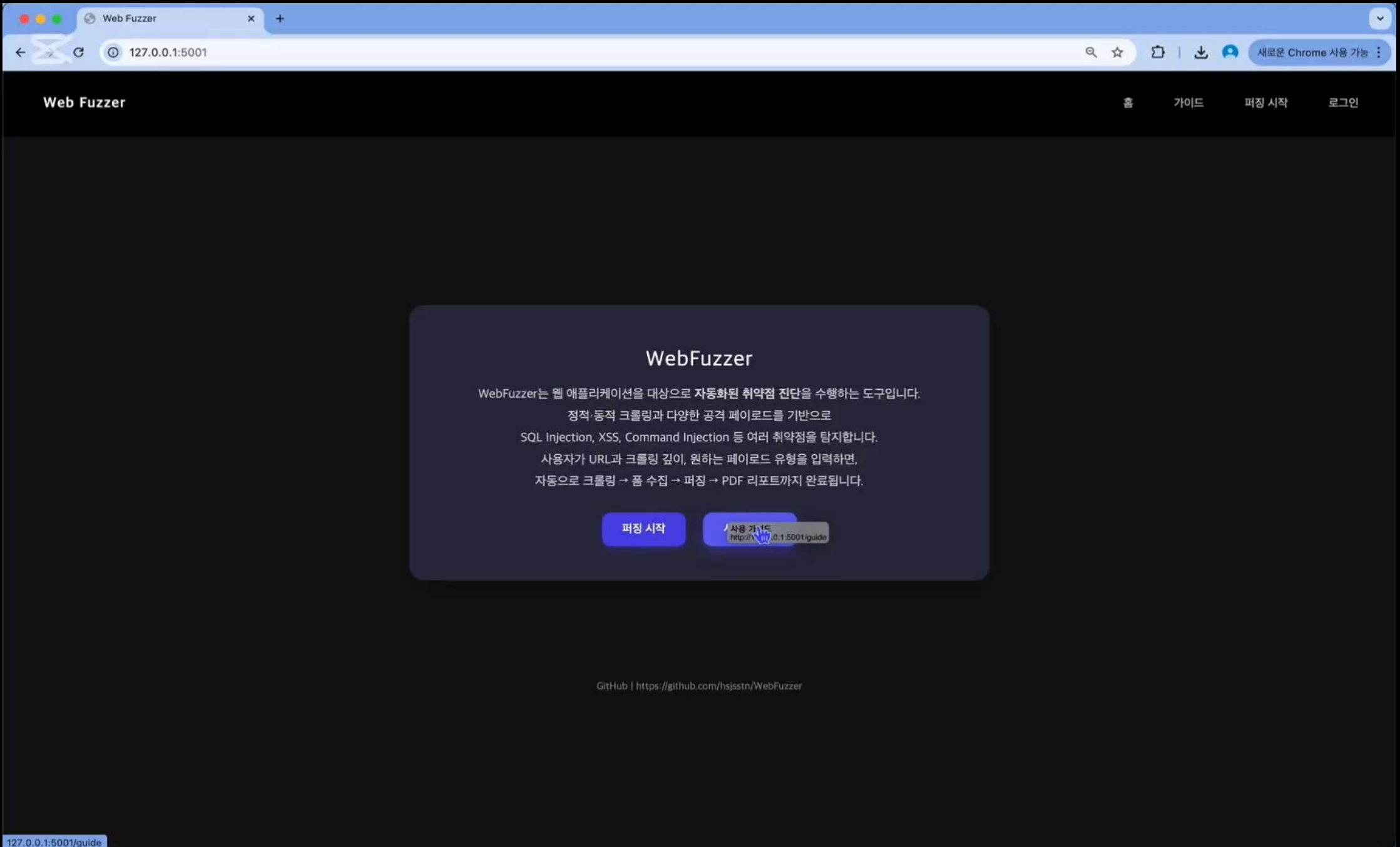
- 2024년 9월 시작
- 퍼징 대상
 - OWASP Juice Shop 페이지
 - 간단한 Testing page 개발
- 구조
 - 크롤러 모듈
 - 퍼저 모듈
 - 결과 보고서 생성 모듈

- 2025년 4월 리팩토링 시작
- 퍼징 대상
 - DVWA 대상
- Develop한 기능
 - 기능별 모듈 분리
 - 공격 페이로드 관리
 - GUI 추가

Web Fuzzer Tool

주요 기능

기능	설명	
Crawler	정적 크롤러	HTML 구조 기반
	동적 크롤러	Selenium 기반의 Headless 브라우저를 활용해 JavaScript에 의해 동적으로 로드되는 요소 탐색
폼/입력 벡터 추출	입력 필드 및 독립 입력 필드 파악하여 퍼징 대상 지정	
Fuzzer	비동기 방식으로 입력 필드에 공격 페이로드 주입 후 취약점 별 응답 분석	
취약점 탐지 로직	각 취약점 별 패턴 및 정규식 기반 탐지	
PDF 리포트 생성	퍼징 결과 및 탐지된 취약점을 PDF로 저장	



REFERENCE

1. <https://csrc.kaist.ac.kr/blog/2020/12/30/a-fuzzing-mirkwood/>
2. <https://www.fuzzingbook.org/>
3. <https://people.eecs.berkeley.edu/~dawnsong/teaching/f12-cs161/readings/toorcon.pdf>
4. <https://owasp.org/>
5. https://lcamtuf.coredump.cx/afl//technical_details.txt
6. https://aflplus.plus/docs/afl-fuzz_approach/
7. Godefroid, Patrice, Adam Kiezun, and Michael Y. Levin. 2008. "Grammar-based Whitebox Fuzzing." SIGPLAN Notices 43, no. 6 (June): 206–215. <https://doi.org/10.1145/1379022.1375607>.
8. Manès, Valentin J. M., HyungSeok Han, Choongwoo Han, Sang Kil Cha, Manuel Egele, Edward J. Schwartz, Maverick Woo, and David Brumley. 2018. "Fuzzing: Art, Science, and Engineering." arXiv preprint arXiv:1812.00140.
9. Boehme, Marcel, Cristian Cadar, and Abhik Roychoudhury. 2021. "Fuzzing: Challenges and Reflections." IEEE Software 38, no. 3 (May-June): 79–86. <https://doi.org/10.1109/MS.2020.3016773>.
10. Li, Junjie, Bin Zhao, and Chao Zhang. 2018. "Fuzzing: A Survey." Cybersecurity 1, no. 6: 1–14. <https://doi.org/10.1186/s42400-018-0002-y>. Li, J., Zhao, B. & Zhang, C. Fuzzing: a survey. Cybersecur 1, 6 (2018). <https://doi.org/10.1186/s42400-018-0002-y>
11. <https://github.com/digininja/DVWA>
12. <https://github.com/ffuf/ffuf>
13. <https://www.zaproxy.org/docs/>
14. <https://github.com/zaproxy/zap-api-docs>

Thank you