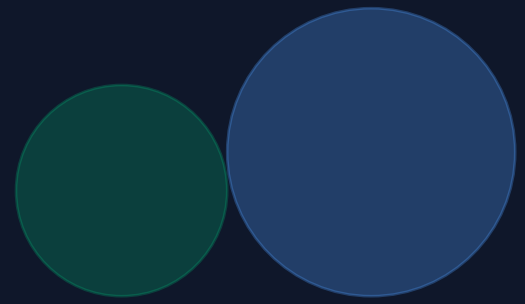
Dreamhack Wargame
Wargame

CVE 분석 발표

ColorGen문제로 보는 Prototype Pollution(CVE-2020-28495)

CVE-2020-28495 (total.js set 함수)



발표자: 채우혁 | 날짜: 2026-01-07

발표 목차

- 1. ColorGen 문제와 total.js의 역할
- 2. CVE-2020-28495 개요 (영향 버전/심각도)
- 3. Prototype Pollution 핵심 원리
- 4. ColorGen 코드에서 취약 패턴 찾기
- 5. 영향
- 6. 예방법

ColorGen과 total.js

ColorGen (Dreamhack)

- 입력값을 서버가 처리
- 요청 body의 키/값을 처리
- 내부 설정 객체인(obj)에 반영

total.js utils.set

- set(obj, path, value) 형식으로 값을 받음
- 예) "a.b.c" → obj.a.b.c = value
- path가 사용자 입력이면 위험

핵심 포인트: 사용자 입력이 path(경로)로 들어가면 Prototype Pollution 이 발생가능

CVE 개요

CVE-2020-28495 요약

CVE-2020-28495

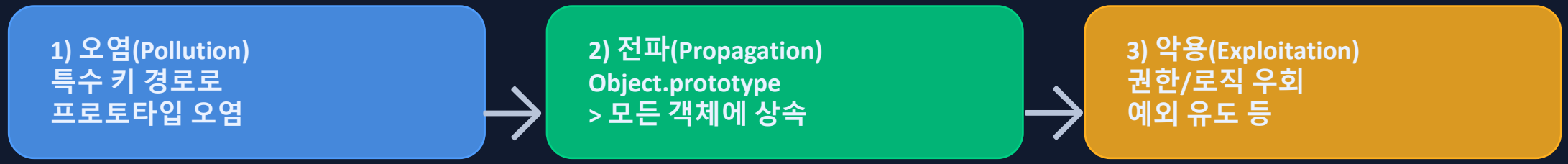
CVSS 7.3 HIGH

취약점 유형: Prototype Pollution

- 대상: total.js < 3.4.7
- 원인: set() 함수에서 경로키 검증이 미흡
- 영향: 앱 구조에 따라 설정 오염 / Property Injection / (일부 상황) RCE 가능
- 조치: 3.4.7+로 업그레이드

Prototype Pollution 핵심

정의와 발생 흐름



Prototype Pollution은 데이터만으로 Object.prototype 같은 기본 프로토타입을 오염시키는 취약점이다.
오염된 속성이 앱 로직에서 사용될 때 예기치 않은 동작(권한 우회, 로직 오류 등)으로 이어진다.

취약점 분석: set(path)

원인: path 키 검증 미흡

- total.js의 set()은 문자열 경로(path)를 파싱해 객체에 값을 넣는다.
- CVE-2020-28495에서는 path에 포함되는 키가 제대로 검증되지 않아
- __proto__ / constructor / prototype 같은 특수 경로를 통해 prototype 오염이 가능해진다.

```
01  // 위험 패턴(개념):
02  set(targetObj, userControlledPath, userValue);
03
04  // 방어 포인트(개념):
05  validatePath(userControlledPath);  // __proto__/constructor/prototype 차단
```

※중요(값이 위험한게 아닌 경로를 입력할수있을때 위험

ColorGen 코드 연결

사용자 입력 > util.set(obj, key, value)

```
01  function change() {
02    // ...
03    for (i in this.body) {
04      util.set(obj, i, this.body[i])
05    }
06    this.plain('OK, changed');
07  }
```

취약점 포인트

- i: 요청 body의 키 (사용자 컨트롤 가능)
- util.set: total.js set()과 유사한 경로 기반
- 따라서 키가 곧 path(경로)가 되는 순간, Prototype Pollution 위험이 생김

영향(Impact)

왜 위험한가? (앱 구조에 따라 달라짐)

Property Injection

프로토타입이 오염되면 Object.prototype에 공격자 값이 들어가고 앱은 원래 없던 속성을 모든 객체에서 기본값처럼 보게 됨
예) 권한/역할/옵션을 속성값으로 판단하는 로직이면 isAdmin, role 같은 값 주입으로 인증·권한 체크가 흔들릴 수 있음

보안 설정 오염

전역 설정/옵션 객체가 오염되면 영향이 앱 전체로 퍼짐
입력 검증, 필터링, 권한 체크 같은 안전장치가 비활성화될 수 있음

연계 시 RCE 가능

어떤 앱은 오염된 속성이 코드 경로/템플릿/명령 실행과 연결될 수 있음

예방 / 대응

대응법

1) 패치 적용
- total.js를 3.4.7 이상으로 업그레이드

2) 입력 검증 로직 추가
- path로 사용할 키는 허용 목록만 통과
- 중첩 키 파싱이 필요 없다면 아예 금지

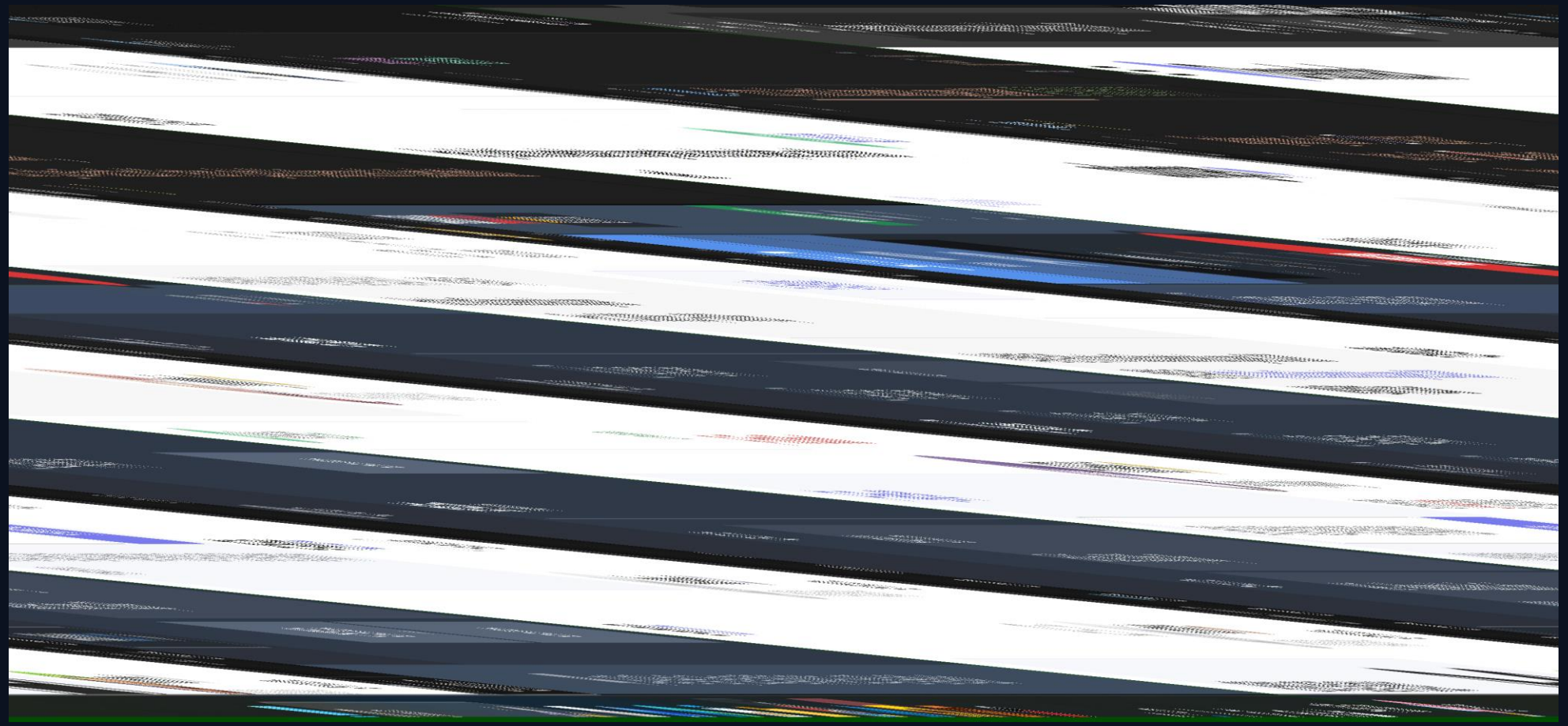
3) 위험 키 차단
- __proto__, constructor, prototype 등

4) 구조적 완화
- 필요 시 Object.create(null) 같은 프로토타입 없는 객체 사용

한 줄 요약

사용자 입력을 경로(path)로 받지 말 것

업데이트 + 키 차단 + 입력 검증 로직 추가가 가장 안전한 조합



감사합니다!