타기관 vs OTTER(KFish) 보안 비교 및 서버 보안 계획

인증·접근제어·요청제한·데이터 노출 방어·운영 서버 보안 기준 비교

타기관

보안 통제 미흡

인증·도메인 검증·요청 제한·데이터 노출 방어가 부족하여 외부 직접 호출과 크롤링 위험이 높음

OTTER (KFish)

주요 통제 적용

JWT 인증, Domain Lock, CORS, Rate Limit, Referer 검증, 개인정보 마스킹 등 운영 보안 체계 적용

1. 보안 항목별 비교

#	항목	타기관	OTTER (KFish)
1	**API 인증**	없음	JWT 토큰 적용
2	**Domain Lock**	없음	6개 허용 도메인 등록
3	**CORS 정책**	미설정	oversea 도메인만 허용
4	**Rate Limiting**	없음	5회 / 5분 제한
5	**관리자 페이지**	WAF 차단	인증 필요
6	**관리자 가입**	오픈 가능성 있음 / 미확인	관리자 토큰 필요
7	**로그인 Brute Force 방어**	무제한 시도 가능	5회 실패 후 429 차단
8	**프롬프트 인젝션 대응**	취약 가능성 있음 / documents 배열 노출	통제 처리
9	**RAG 원문 노출**	전문 그대로 반환	summary만 반환
10	**데이터 건수 노출**	키워드당 8~10건 노출	제한적 노출
11	**evil Origin 접근**	가능	차단
12	**Origin 없는 직접 접근**	가능	인증 필요
13	**Referer 검증**	없음	검증 적용

#	항목	타기관	OTTER (KFish)
14	**user_id 조작**	임의 값 사용 가능	인증 기반 처리
15	**내부 IP 노출**	N/A	server_url 제거
16	**민원 개인정보**	N/A	마스킹 처리
17	**QA 내부망 분리**	N/A	192.168.x.x 내부망만 접근
18	**보안 헤더**	없음	X-Frame / XSS / Referrer 정책 적용
19	**DEBUG 모드**	확인 불가	false
20	**Vite dev 모드**	N/A	프로덕션 빌드
21	**embed 외부 삽입**	어디든 가능	등록 도메인만 허용
22	**API 직접 크롤링**	무제한 가능	Domain Lock 차단

2. 종합 평가

구분	타기관	OTTER (KFish)
인증 체계	미흡	적용
접근 제어	미흡	Domain Lock / Referer / CORS 적용
공격 방어	미흡	Rate Limit / Brute Force 차단 적용
데이터 보호	원문·건수 노출 가능	요약 반환 / 개인정보 마스킹
운영 보안	확인 필요	DEBUG false / 프로덕션 빌드
전체 평가	보안 취약점 다수 확인 필요	주요 보안 통제 적용 완료

총평

타기관은 현재 확인된 기준상 인증·접근 제어·요청 제한·데이터 노출 방어가 부족하여, 외부에서 API 직접 호출 및 데이터 수집이 가능할 위험이 높다.

반면 OTTER(KFish)는 JWT 인증, Domain Lock, CORS 제한, Rate Limiting, Referer 검증, 개인정보 마스킹, 내부망 분리 등 주요 보안 통제가 적용되어 있어 운영 서비스 기준의 기본 보안 체계를 갖춘 상태로 평가된다.

보안성 기준으로는 OTTER(KFish)가 타기관 대비 명확히 우수하다.

3. 서버 보안 계획

Ubuntu 기반 웹/AI 서버 운영 기준으로, 백신 자체보다 방화벽·SSH 보호·자동 보안 업데이트·업로드 파일 검사 체계를 우선 적용한다.

권장 설치 조합

```
sudo apt install clamav clamav-daemon clamav-freshclam ufw fail2ban unattended-upgrades
```

운영 원칙

운영 중인 웹/AI 서버는 ClamAV 실시간 감시보다 업로드 폴더 정기 스캔이 현실적이다. 대용량 모델·RAG·파일 업로드가 함께 동작하는 환경에서는 실시간 감시가 I/O 병목을 만들 수 있으므로, 업로드 디렉터리를 대상으로 정기 검사와 로그 기록을 우선 적용한다.

업로드 폴더 정기 검사 예시

```
sudo crontab -e
```

```
# 매일 새벽 3시 검사
0 3 * * * clamscan -r -i /home/weaver/uploads >> /var/log/clamav-upload-scan.log
```

4. 서버 보안 적용 체크리스트

#	보안 영역	적용 내용	우선순위
1	**기본 보안 패키지**	ClamAV, UFW, Fail2ban, unattended-upgrades 설치	즉시 적용
2	**방화벽**	UFW로 SSH/HTTP/HTTPS 등 필요한 포트만 허용	필수
3	**SSH 보호**	비밀번호 로그인 제한, 키 기반 로그인, 포트 접근 제한 검토	필수
4	**로그인 방어**	Fail2ban으로 SSH·웹 로그인 반복 실패 차단	필수
5	**백신 검사**	실시간 감시보다 업로드 폴더 정기 스캔 중심 운영	권장
6	**자동 업데이트**	unattended-upgrades로 보안 패치 자동 적용	권장
7	**업로드 폴더 관리**	/home/weaver/uploads 기준 정기 검사 및 감염 파일 로그화	권장
8	**서비스 운영**	DEBUG=false, 프로덕션 빌드, 내부 IP·server_url 노출 제거	필수
9	**웹/API 보호**	JWT, Domain Lock, CORS, Referer, Rate Limit 유지	필수
10	**개인정보 보호**	민원·문서 데이터 마스킹 및 원문 반환 최소화	필수

한 줄 결론

우분투 백신은 ClamAV면 충분하고, 진짜 중요한 것은 방화벽·SSH 보호·자동 보안 업데이트·업로드 폴더 정기 스캔이다.