

VULNMART

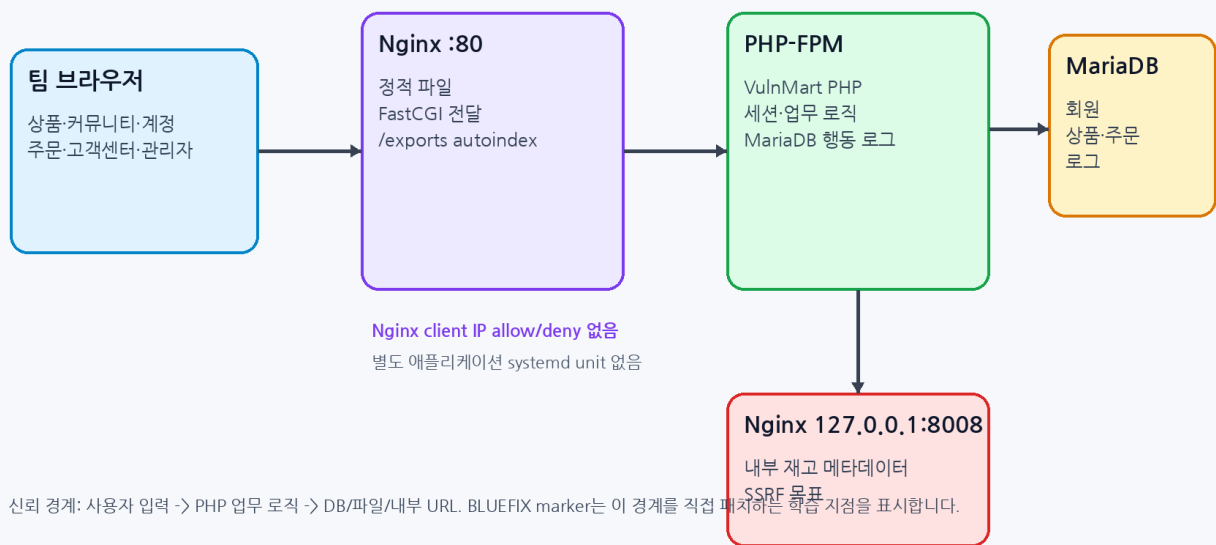
Scenario Range

Rocky Linux 9.7 배포 · 시나리오형 웹 침투테스트 · 코드 패치 통합 매뉴얼

Nginx + PHP-FPM + MariaDB | Version 4.0.0 | 2026-07-19

VulnMart Scenario Range - 실행 아키텍처

정상 쇼핑 업무와 침투테스트 시나리오가 같은 애플리케이션 데이터 흐름을 사용합니다.



사용 범위

이 자료와 패키지는 승인된 사설 교육망의 침투테스트 훈련용입니다. 공개 인터넷, 실제 고객 정보, 실제 자격증명, 외부 시스템을 대상으로 사용하지 않습니다.

문서 성격

블루·퍼플·레드팀이 실습 환경을 구상하고 학습하는 데 참고할 수 있는 예시입니다. 조직의 규정, 법적 요구사항, 운영 표준을 대체하는 강제 지침이 아닙니다.

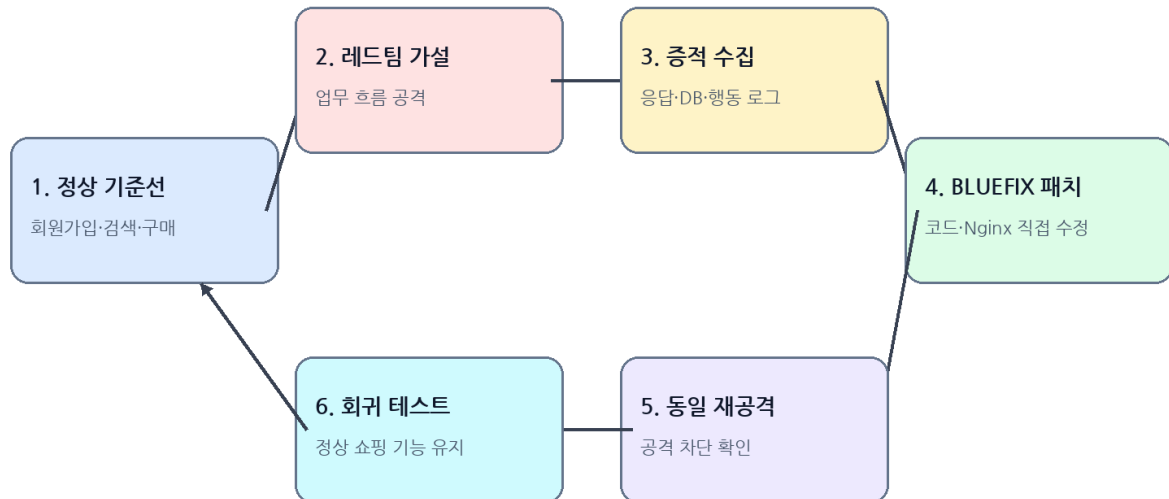
문서 ID: VM-SR-MANUAL-4.0.0

이 버전에서 달라진 실습 흐름

VulnMart 4.0.0은 취약점 이름을 누르면 즉시 실행되는 독립 실습 페이지를 제거했습니다. 취약점은 상품 검색, 커뮤니티, 계정 관리, 주문, 고객센터, 브라우저 기능, 관리자 업무처럼 쇼핑몰의 실제 기능에 존재합니다. 훈련 보드는 공격 버튼이 아니라 시나리오의 출발점과 관련 코드 marker를 안내합니다.

- 레드팀: 정상 사용자 또는 제한된 권한에서 시작하여 여러 업무 기능을 연결하고, 영향과 증거를 정리합니다.
- 블루팀: LOW/MEDIUM/SAFE 전환 없이 실제 PHP 또는 Nginx 소스를 수정합니다.
- 퍼플팀: 같은 입력을 재실행하여 공격 차단, 정상 기능, 애플리케이션 로그를 함께 검증합니다.

공격 - 관찰 - 패치 - 재검증 순환



패치 성공의 기준은 공격 차단 하나가 아니라 정상 기능, 관찰 가능성, 운영 복구 가능성까지 함께 유지되는 것입니다.

marker 우선 원칙

라인 번호는 패치 과정에서 변할 수 있습니다. 매뉴얼의 라인 번호는 최초 배포본 기준이며, 실제 수정 지점은 grep으로 찾는 [BLUEFIX:...] marker를 기준으로 판단합니다.

```
grep -RIn '\[BLUEFIX:' /var/www/vulnmart /etc/nginx/conf.d/vulnmart.conf /etc/nginx/nginx.conf
```

목차

구분	내용
Part I	범위 설계와 토폴로지
1	목적, 역할과 안전 경계
2	실행 아키텍처와 신뢰 경계
3	네트워크 토폴로지 구상 예시
4	정상 쇼핑물 기준선
Part II	Rocky Linux 9.7 배포와 운영
5	설치 전 확인
6	원클릭 설치와 배포 결과
7	파일, 서비스, 데이터베이스와 SELinux
8	설치 검증, 장애 분석, 초기화
Part III	팀별 주도 실습
9	PHP 애플리케이션 행동 로그
10	블루팀 코드 패치 흐름
11	퍼플팀 검증 흐름
12	레드팀 증거와 통보 방식
Part IV	12 개 시나리오 실습
13	시나리오 전체 맵
14-25	SCN-01 부터 SCN-12
Part V	방어 구현 참조
26	공통 방어 패턴
27	BLUEFIX 전체 인덱스
28	정상 기능 회귀 체크리스트
29	참고 문서

Part I. 범위 설계와 토폴로지

1. 목적, 역할과 안전 경계

이 환경의 목적은 취약점 한 개의 문법을 암기하는 것이 아니라, 정상 업무 흐름에서 신뢰 경계가 어디에서 깨지고 그 결과가 다른 기능으로 어떻게 확장되는지 관찰하는 것입니다. 팀은 실습 전에 범위, 계정, 대상 IP, 허용 시간, 증거 저장 위치와 복구 기준을 합의하는 편이 좋습니다.

역할	주요 질문	대표 산출물
블루팀	어떤 서비스와 데이터가 있으며, 변경 시 정상 기능을 어떻게 보존할 것인가?	토폴로지, 백업, 로그 기준선, 패치 diff, 회귀 결과
퍼플팀	공격 가설과 방어 변경을 같은 증거로 비교할 수 있는가?	가설 카드, 공격 전후 비교, 탐지·차단·회귀 기록
레드팀	어떤 정상 기능에서 시작해 권한·데이터·서버 영향으로 확대되는가?	타임라인, 요청·응답, 영향, 재현 절차, 개선 제안

안전 제한

명령·코드 실행 실습은 hostname, id, phpversion 같은 무해한 증명으로 제한합니다. 파일 삭제, 권한 지속성, 외부 콜백, 대량 자동화, 서비스 중단은 시나리오 범위에 포함하지 않습니다.

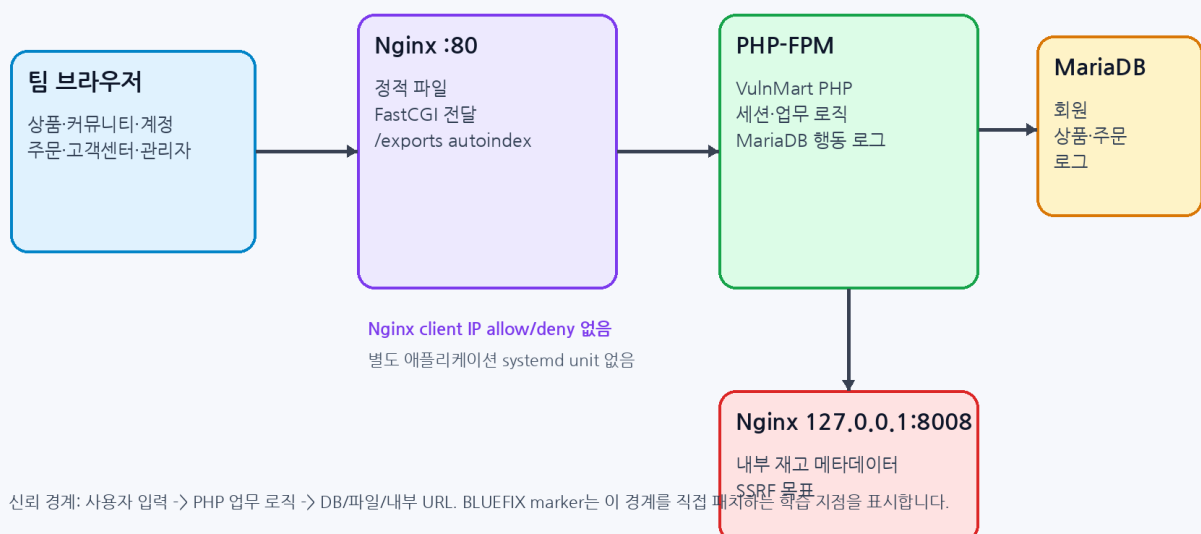
권장 준비물

- 스냅샷 가능한 Rocky Linux 9.7 VM 1 대와 팀 브라우저
- 허가된 사설 네트워크, 서버 IP와 실습 시간대 기록
- 브라우저 개발자 도구 또는 승인된 프록시, 선택적 패킷 캡처 도구
- 패치 전 소스 복사본 또는 Git 저장소, DB 초기화 절차

2. 실행 아키텍처와 신뢰 경계

VulnMart Scenario Range - 실행 아키텍처

정상 쇼핑 업무와 침투테스트 시나리오가 같은 애플리케이션 데이터 흐름을 사용합니다.



메인 사이트는 TCP 80의 Nginx가 정적 파일을 제공하고 PHP 요청을 Unix socket의 PHP-FPM에 전달합니다. PHP는 MariaDB의 회원, 상품, 주문, 게시판, 고객센터, 로그 데이터를 사용합니다. SSRF 시나리오의 내부 메타데이터는 같은 Nginx가 127.0.0.1:8008에만 별도 호스팅합니다. VulnMart 전용 systemd 서비스는 만들지 않습니다.

경계	정상 데이터 흐름	의도된 실패 예
브라우저 -> PHP	폼, 쿠키, URL, JSON	클라이언트 가격·역할 쿠키 신뢰, XSS, CSRF
PHP -> MariaDB	준비문, 객체 소유권	문자열 결합 SQL, IDOR, 평문 인증정보
PHP -> 파일 시스템	업로드, 송장, 도움말	경로 조작, 실행 업로드, LFI
PHP -> 내부 URL	고객센터 참고 URL 미리보기	loopback/file scheme SSRF
Nginx -> PHP-FPM	SCRIPT_FILENAME과 Unix socket	업로드 디렉터리의 PHP 실행
관리자 -> 서버 기능	가격 정책, 공급사 진단	eval, shell command 결합

3. 네트워크 토폴로지 구상 예시

아래 예시는 선택 가능한 학습 토폴로지입니다. 패키지는 Nginx에 source IP allow/deny 규칙을 넣지 않으며, 메인 사이트는 서버의 모든 인터페이스에서 TCP 80을 수신합니다. 팀은 사설망, 가상 스위치, 상위 방화벽과 관찰 장비의 배치를 별도로 설계할 수 있습니다.

형태	구성 아이디어	관찰 포인트
단일 VM	브라우저와 서버가 같은 host-only 또는 사설 VLAN	빠른 기능 검증, 서버 로그 중심
팀 분리	레드 클라이언트, 블루 관리 단말, 서버를 같은 사설 세그먼트에 배치	역할별 계정과 증적 분리
관찰 구간	가상 스위치 mirror 또는 승인된 센서 추가	HTTP 평문, 반복 로그인, API Method 패턴
패치 실습	서버 snapshot A는 취약, snapshot B는 패치 상태	같은 시나리오의 전후 비교

IP 제한과 네트워크 통제의 구분

패키지는 Nginx allow/deny나 특정 client CIDR 제한을 설치하지 않습니다. 공개 여부와 접근 경계는 사설 주소, 가상화 네트워크, 상위 방화벽과 팀 운영 절차에서 결정하는 모델입니다.

4. 정상 쇼핑물 기준선

공격을 시작하기 전에 정상 기능을 한 번 완료해 두면 패치 후 회귀 여부를 판단하기 쉽습니다.

최초 기준선 절차

1. admin/admin123, alice/alice123, bob/bob123, victim/spring2026 계정으로 로그인 가능 여부를 확인합니다.
2. 상품 목록에서 상품을 열고 후기, 장바구니, 쿠폰 확인, 주문 완료까지 진행합니다.
3. 커뮤니티에 일반 텍스트 게시글을 작성하고 다른 계정에서 읽습니다.
4. 프로필에서 이메일, 비밀번호, 프로필 이미지를 정상적으로 변경합니다.
5. 고객센터에 참고 URL 없이 문의를 등록하고 최근 문의 목록을 확인합니다.
6. 관리자 콘솔에서 상품, 주문, 회원, 행동 로그와 로그 옵션을 확인합니다.

기능	기대 결과
검색	정상 상품명 검색 시 관련 상품만 표시
회원가입/로그인	신규 계정 생성과 세션 유지
장바구니/결제	DB 가격을 표시하고 주문·송장 생성

기능	기대 결과
주문 조회	본인 주문 목록과 상세 표시
커뮤니티/후기	일반 텍스트 작성·조회
관리자	상품·주문·회원·로그 조회와 수정

Part II. Rocky Linux 9.7 배포와 운영

5. 설치 전 확인

install.sh 는 다른 설정을 적용하지 않은 Rocky Linux 9.7 초기 상태를 기준으로 작성되어 있습니다. 설치는 root 권한과 DNF 저장소 접근이 필요합니다. 애플리케이션 런타임은 Nginx, PHP-FPM, MariaDB 이며 Composer, Python, Node.js, Docker 를 사용하지 않습니다.

설치 전 운영체제와 네트워크 확인 예시

```
cat /etc/os-release
uname -m
ip -br address
getent hosts mirrors.rockylinux.org || true
```

스냅샷 시점

설치 직전 snapshot 과 설치 완료 후 정상 기준선 snapshot 을 각각 남겨두면 시나리오 반복과 블루팀 패치 비교가 단순해집니다.

패키지 구조

경로	용도
install.sh	패키지 설치, DB 초기화, Nginx/PHP-FPM 배포, 검증
verify.sh	서비스, socket, HTTP, 내부 endpoint, IP 제한 부재 확인
reset_lab.sh	DB·업로드·송장 데이터 초기화, 선택적 행동 로그 삭제
uninstall.sh	VulnMart 파일과 설정 제거
app/public	메인 쇼핑물과 관리자 화면
app/inc	DB, 세션, 인증, 행동 로그 공통 코드
deploy	Nginx main/vhost 와 PHP-FPM pool
docs/PATCH_MAP_KO.md	소스 marker 기반 빠른 패치 맵

6. 원클릭 설치와 배포 결과

```
tar -xzf VulnMart_Lab_Rocky97_PHP_MariaDB_v4.0.0.tar.gz
cd VulnMart_Lab_Rocky97_PHP_MariaDB_v4.0.0
sudo ./install.sh
```

설치기는 패키지 설치, 표준 서비스 활성화, MariaDB 계정·스키마 생성, 샘플 데이터 seed, PHP-FPM pool, Nginx 설정, SELinux label/boolean, HTTP 방화벽 서비스, 문법 검사와 end-to-end 요청을 순서대로 수행합니다.

설치 결과	값
메인 쇼핑물	http://SERVER_IP/

설치 결과	값
훈련 보드	http://SERVER_IP/scenarios.php
커뮤니티	http://SERVER_IP/community.php
관리자	http://SERVER_IP/admin/
내부 SSRF 목표	http://127.0.0.1:8008/metadata.php - 서버 loopback
표준 서비스	mariadb, php-fpm, nginx
VulnMart 전용 unit	없음
Nginx client IP 제한	없음

설치 직후 확인

```
sudo systemctl status mariadb php-fpm nginx --no-pager
sudo ./verify.sh
sudo ss -lntp | grep -E '(:80|127\.0\.\0\.\1:8008)'
```

7. 파일, 서비스, 데이터베이스와 SELinux

구분	경로 또는 이름	설명
웹 루트	/var/www/vulnmart/public	Nginx document root
공통 코드	/var/www/vulnmart/inc	bootstrap, 인증, DB, 로그
설정	/etc/vulnmart/config.php	DB socket·계정·app secret, root:nginx 0640
PHP-FPM pool	/etc/php-fpm.d/vulnmart.conf	nginx 사용자, /run/php-fpm/vulnmart.sock
Nginx vhost	/etc/nginx/conf.d/vulnmart.conf	TCP 80, loopback 8008
상태 데이터	/var/lib/vulnmart	세션, 임시 업로드, 송장
업로드	/var/www/vulnmart/public/uploads	의도적으로 웹 실행 경로에 존재
DB	vulnmart	회원·상품·주문·로그 등
DB 계정	vulnmart@localhost	Unix socket 연결

SELinux Enforcing 상태를 유지하면서 웹 루트는 httpd_sys_content_t, 업로드와 상태 디렉터리는 httpd_sys_rw_content_t로 등록합니다. SSRF 실습을 위해 PHP의 outbound 연결이 필요하므로 httpd_can_network_connect가 활성화됩니다. 블루팀은 egress 통제를 별도의 방어 과제로 설계할 수 있습니다.

SELinux 확인 예시

```
getenforce
ls -ldZ /var/www/vulnmart/public/uploads /var/lib/vulnmart
getsebool httpd_can_network_connect httpd_can_network_connect_db
matchpathcon /var/www/vulnmart/public/index.php
```

8. 설치 검증, 장애 분석, 초기화

502 또는 빈 화면 진단 순서

1. nginx -t와 php-fpm -t로 구성 문법을 먼저 확인합니다.
2. systemctl status php-fpm nginx와 journalctl -u php-fpm -u nginx를 확인합니다.
3. /run/php-fpm/vulnmart.sock의 존재, 소유자 nginx:nginx, mode 0660을 확인합니다.
4. /var/log/nginx/vulnmart_error.log와 /var/log/php-fpm/vulnmart-error.log를 확인합니다.
5. /etc/vulnmart/config.php의 권한과 MariaDB socket, 계정 연결을 확인합니다.

6. ausearch -m AVC -ts recent로 SELinux 차단을 확인한 뒤 원인을 수정합니다. SELinux를 비활성화하는 방식은 비교 실습의 관찰력을 떨어뜨립니다.

장애 분석 명령 예시

```
sudo nginx -t
sudo php-fpm -t
sudo systemctl status mariadb php-fpm nginx --no-pager
sudo journalctl -u php-fpm -u nginx -n 100 --no-pager
sudo ls -lZ /run/php-fpm/vulnmart.sock
sudo ausearch -m AVC -ts recent | tail -80
```

초기화와 제거

```
sudo ./reset_lab.sh
# 행동 로그까지 삭제하는 반복 실습
sudo ./reset_lab.sh --clear-logs
# 전체 제거
sudo ./uninstall.sh
```

패치 보존

reset_lab.sh는 DB 실습 데이터, 업로드, 송장과 서버 측 PHP 세션을 되돌리지만 블루팀이 수정한 PHP/Nginx 소스는 되돌리지 않습니다. 브라우저 쿠키는 별도로 삭제하고, 소스 원복은 snapshot 또는 버전 관리로 수행합니다.

Part III. 팀별 주도 실습

9. PHP 애플리케이션 행동 로그

VulnMart의 사용자 행동 로그는 Nginx access log를 재사용하지 않습니다. PHP 코드의 log_event()와 shutdown logger가 MariaDB action_logs 테이블에 업무 이벤트와 요청 메타데이터를 직접 기록합니다.

필드	예시	분석 의미
event_name	community.post.view	업무 또는 보안 관련 이벤트
result	success/failure/error	결과
username/user_role	victim/user	세션 기준 주체
session_fingerprint	SHA-256 축약값	세션 원문을 저장하지 않는 연계 키
client_ip/user_agent	192.168.x.x / Browser	클라이언트 식별 보조
http_method/request_path	GET /profile.php?...	요청 흐름
scenario_key	community-account-takeover	시나리오별 필터
query/form/detail	마스킹 JSON	입력·애플리케이션 판단 증거

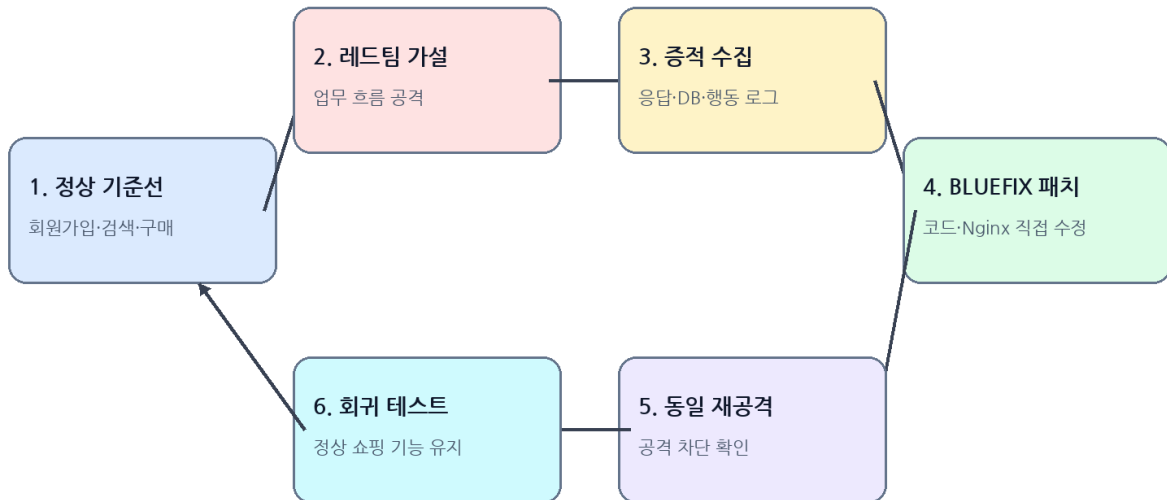
password, token, csrf, cookie, session, card 등 민감 키는 [REDACTED]로 저장됩니다. 관리자 콘솔의 /admin/logs.php에서 event, scenario, actor로 필터링할 수 있습니다. 로그 옵션은 기록과 입력 캡처를 바꾸지만 취약점 자체를 고치지 않습니다.

MariaDB에서 시나리오 로그를 조회하는 예시

```
SELECT created_at,event_name,username,http_method,request_path,scenario_key
FROM action_logs
WHERE scenario_key='community-account-takeover'
ORDER BY id DESC LIMIT 30;
```

10. 블루팀 코드 패치 흐름

공격 - 관찰 - 패치 - 재검증 순환



패치 성공의 기준은 공격 차단 하나가 아니라 정상 기능, 관찰 가능성, 운영 복구 가능성까지 함께 유지되는 것입니다.

한 개 marker 를 패치하는 권장 학습 흐름

1. 정상 요청 1 건과 공격 요청 1 건을 실행하고 응답, DB 변화, action_logs 를 저장합니다.
2. grep 으로 BLUEFIX marker 와 인접 코드를 찾습니다.
3. 입력 차단만 하지 말고 신뢰할 데이터의 출처, 권한, 상태 전이를 다시 정의합니다.
4. PHP 파일은 php -l, Nginx 는 nginx -t 를 통과시킨 뒤 reload 합니다.
5. 같은 공격 입력이 차단되는지 확인합니다.
6. 정상 구매·로그인·관리 기능이 계속 동작하는지 회귀 테스트합니다.
7. 차단 결과와 정상 요청이 애플리케이션 로그에 구분되어 남는지 확인합니다.

패치 작업 예시

```

sudo cp /var/www/vulnmart/public/profile.php /root/profile.php.before
sudo vi /var/www/vulnmart/public/profile.php
php -l /var/www/vulnmart/public/profile.php
sudo systemctl reload php-fpm
sudo ./verify.sh
  
```

소스 패치의 단위

한 번에 모든 marker 를 수정하기보다 시나리오 한 개의 신뢰 경계를 묶어 패치하면 공격 차단과 정상 회귀의 인과관계를 비교하기 쉽습니다.

11. 퍼플팀 검증 흐름

단계	레드팀	블루팀	공동 확인
가설	업무 출발점과 공격 체인 정의	관찰 가능한 이벤트 정의	성공·실패 기준 합의
기준선	정상 기능 수행	로그·DB·파일 기준선 저장	타임라인 동기화
공격	최소 영향 입력 실행	로그·상태 변화 관찰	증적 ID 연결
패치	재현 절차 고정	BLUEFIX 코드 수정	변경 diff 검토

단계	레드팀	블루팀	공동 확인
재공격	동일 요청 재실행	차단 이벤트 확인	우회와 오탐 검토
회귀	정상 사용자 흐름 반복	서비스·데이터 정상성 확인	완료 조건 승인

좋은 완료 조건

HTTP 403 하나만 나오면 완료가 아닙니다. 공격 결과가 바뀌고, 정상 기능이 유지되며, 차단 또는 검증 실패가 적절한 로그로 남고, 운영자가 복구할 수 있어야 합니다.

12. 레드팀 증거와 통보 방식

통보는 "취약하다"는 결론보다 재현 가능한 업무 맥락, 전제 권한, 최소 요청, 실제 영향, 로그 상관관계와 개선 검증 기준을 함께 제공하는 형태가 학습에 유용합니다.

필드	작성 예
제목	커뮤니티 게시글 열람으로 비밀번호 변경 가능
전제	attacker=alice, victim 로그인 상태, 게시글 열람
업무 시작점	/community.php -> /post_new.php
요청 체인	게시글 저장 -> victim GET /post.php -> GET /profile.php? action=password_change...
영향	victim 계정의 인증정보가 공격자가 정한 값으로 변경
증적	게시글 ID, HTTP 요청, users 변경 시각, action_logs event ID
BLUEFIX	STORED-XSS-01, CSRF-PASSWORD-01
재검증	HTML이 문자로 표시되고 GET 요청은 405, 정상 POST+CSRF는 성공

공격 통보 범위

실습 보고서에는 로컬 사설 주소와 가짜 계정만 사용합니다. 실제 자격증명, 외부 도메인, 실제 조직 정보는 첨부하지 않습니다.

Part IV. 12 개 시나리오 실습

13. 시나리오 전체 맵

SCN	이야기	업무 시작점	핵심 취약점
01	커뮤니티 -> 계정 탈취	community.php	Stored XSS, CSRF
02	검색·로그인 정찰	search.php	SQLi, Reflected XSS, 오류 누출
03	인증·복구 연쇄	login.php	Brute force, weak password, reset, session
04	주문 금액 조작	cart.php	process validation, cookie, automation
05	주문·파일 접근	orders.php	IDOR, CSRF, traversal
06	고객센터 내부 피벗	support.php	SSRF, internal metadata
07	프로필 실행 foothold	profile.php	upload, PHP execution, LFI
08	브라우저 캠페인	compare.php	DOM XSS, prototype pollution, CSP, redirect
09	관리자 진입	robots.txt, /admin/	admin exposure, cookie authz, info leak
10	관리자 서버 영향	admin pricing/diagnostics	code injection, command injection
11	백오피스 API 악용	api/products.php	PUT/DELETE, authz, automation
12	운영 노출면 정찰	health.php, /exports/	HTTP, indexing, banners, errors

SCN-01

커뮤니티 게시글에서 시작하는 계정 탈취

정상 회원인 공격자가 커뮤니티 글을 작성하고, 로그인한 피해자가 글을 읽는 업무 흐름을 비밀번호 변경 기능과 연결합니다. 이 시나리오는 저장형 XSS/HTML 주입과 CSRF 보호 부재가 결합될 때 단일 입력 검증 실패가 계정 탈취로 확대되는 모습을 보여줍니다.

항목	내용
업무 시작점	/community.php -> /post_new.php -> /post.php?id=...
취약점 체인	Stored XSS, 상태 변경 GET, CSRF token 부재, 현재 비밀번호 미검증
BLUEFIX 위치	app/public/post_new.php:2 [STORED-XSS-01] app/public/post.php:3 [STORED-XSS-01] app/public/profile.php:6 [CSRF-PASSWORD-01]

정상 업무 기준선

- alice 로 로그인하여 일반 텍스트 게시글을 작성합니다.
- victim 으로 로그인하여 게시글이 정상 표시되는지 확인합니다.
- victim 이 /profile.php 에서 일반 폼으로 비밀번호를 변경하고 다시 로그인합니다.

공격 체인

1. 공격자 역할의 alice 로 로그인하고 커뮤니티 글쓰기 화면을 엽니다.
2. 제목은 팀이 식별할 수 있는 값으로 하고, 본문에 아래 교육용 img 태그를 저장합니다.
3. 다른 브라우저 프로필에서 victim/spring2026 으로 로그인한 뒤 해당 게시글을 읽습니다. 게시글 HTML 이 그대로 렌더링되면서 victim 세션으로 비밀번호 변경 GET 이 발생합니다.
4. victim 에서 로그아웃한 뒤 공격자 브라우저에서 victim/TeamRange!2026 으로 로그인하여 영향 범위를 확인합니다.
5. 관리자 로그에서 community.post.create, community.post.view, profile.password.change 와 동일 시간대 요청을 연결합니다.

교육용 입력 또는 요청 예시 - SERVER/REDTEAM_HOST 는 승인된 사설 범위로 치환

```

```

성공 증거와 관찰

- 게시글 본문에 저장된 HTML 과 게시글 ID
- victim 브라우저 Network 의 /profile.php?action=password_change 요청
- users 테이블 또는 관리자 회원 화면의 변경된 교육용 비밀번호
- action_logs 의 community.post.view 및 profile.password.change 타임라인

애플리케이션 로그

scenario_key=community-account-takeover. new_password 입력은 민감 키 마스킹으로 [REDACTED] 처리됩니다.

블루팀 코드 패치

수정 위치

app/public/post_new.php:2 [STORED-XSS-01]
app/public/post.php:3 [STORED-XSS-01]
app/public/profile.php:6 [CSRF-PASSWORD-01]

게시글 본문을 HTML 로 실행하지 않고 출력 컨텍스트에 맞게 인코딩합니다. 비밀번호 변경은 POST 만 허용하고 CSRF token, 현재 비밀번호, 재인증, 변경 후 세션 회전을 적용합니다. 두 지점을 함께 수정해야 공격 체인이 끊어집니다.

방어 구현 예시 - 환경에 맞게 검토하여 적용

```
// post.php
<div class="post-body"><?=php nl2br(e($p['body'])) ?&gt;&lt;/div&gt;

// profile.php
if ($_SERVER['REQUEST_METHOD'] !== 'POST') {
    http_response_code(405); exit;
}
if (!verify_csrf()) { http_response_code(403); exit('CSRF 검증 실패'); }
// DB 의 password_hash 를 조회하고 password_verify($_POST['current_password'], ...) 수행
session_regenerate_id(true);</pre

```

동일 공격 재검증과 정상 기능 회귀

- 같은 img 본문이 실행되지 않고 문자로 표시되는지 확인합니다.
- GET /profile.php?action=password_change...가 405 또는 무변경으로 끝나는지 확인합니다.
- 정상 사용자 폼에 csrf_input()과 현재 비밀번호를 넣었을 때만 변경되는지 확인합니다.
- 일반 텍스트 게시글 작성·열람과 정상 비밀번호 변경이 유지되는지 확인합니다.

SCN-02

상품 검색과 로그인에서 데이터 정찰

검색과 로그인이라는 일상 기능의 문자열 결합, 반사 출력, 상세 예외를 연결하여 데이터베이스 구조와 인증 경계를 정찰합니다.

항목	내용
업무 시작점	/search.php, /login.php
취약점 체인	SQL Injection, Reflected XSS, 상세 SQL 오류와 stack 정보 누출
BLUEFIX 위치	app/public/search.php:6 [SEARCH-SQLI-01] app/public/search.php:20 [SEARCH-XSS-02] app/public/search.php:13 [ERROR-DISCLOSURE-03] app/inc/auth.php:7 [LOGIN-SQLI-01]

정상 업무 기준선

- 상품명 NeoWave 를 검색하여 관련 상품만 표시되는지 확인합니다.
- admin 또는 alice 의 정상 아이디·비밀번호로 로그인합니다.

공격 체인

6. 검색어에 작은따옴표 한 개를 넣어 SQL 오류와 생성된 SQL 이 응답에 표시되는지 확인합니다.
7. 검색어에 아래 SQL 조건을 넣어 정상 검색 범위를 벗어난 상품이 표시되는지 관찰합니다.
8. 검색어에 무해한 반사형 XSS 표식을 넣어 HTML 이 실행되는지 확인합니다.
9. 로그인 아이디에 admin' -- 와 임의 비밀번호를 넣어 비밀번호 조건이 주석 처리되는지 확인합니다.
10. catalog.search 및 auth.login 관련 로그에서 성공, 오류, 요청 입력을 비교합니다.

교육용 입력 또는 요청 예시 - SERVER/REDTEAM_HOST는 승인된 사설 범위로 치환

```
검색 SQLi: ' OR 1=1 --
반사형 XSS: <img src=x onerror="document.body.dataset.vulnerable-xss">
로그인 SQLi 아이디: admin' --
비밀번호: anything
```

성공 증적과 관찰

- 검색 결과 건수 변화와 SQL 오류 응답
- 브라우저 Elements 에서 검색어가 raw HTML 로 삽입된 위치
- 로그인 후 관리자 메뉴 표시 또는 /admin/ 접근
- action_logs 의 catalog.search, auth.login.sql_error, auth.login

애플리케이션 로그

scenario_key=catalog-recon. 오류 시 detail_data 에 SQL 과 DB 예외가 기록됩니다.

블루팀 코드 패치

수정 위치

```
app/public/search.php:6 [SEARCH-SQLI-01]
app/public/search.php:20 [SEARCH-XSS-02]
app/public/search.php:13 [ERROR-DISCLOSURE-03]
app/inc/auth.php:7 [LOGIN-SQLI-01]
```

검색과 로그인 모두 준비문을 사용하고, 비밀번호는 password_hash/password_verify로 검증합니다. 검색어는 SQL 바인딩과 별도로 HTML 출력 시 e()로 인코딩하며, 사용자 응답에는 일반 오류와 request-id 만 제공합니다.

방어 구현 예시 - 환경에 맞게 검토하여 적용

```
// search.php
$stmt = db()->prepare(
    'SELECT id,sku,name,description,price,stock,image FROM products '
    . 'WHERE active=1 AND (name LIKE ? OR description LIKE ?)'
);
$like = '%' . $q . '%';
$stmt->execute([$like, $like]);

// 출력
echo e($q);

// auth.php
$stmt = db()->prepare('SELECT * FROM users WHERE username=? LIMIT 1');
$stmt->execute([$username]);
$user = $stmt->fetch();
$ok = $user && password_verify($password, $user['password_hash']);
```

동일 공격 재검증과 정상 기능 회귀

- 작은따옴표와 SQL 메타문자가 일반 검색 문자열로 처리되는지 확인합니다.
- XSS 태그가 문자로 보이고 DOM 속성이나 스크립트가 실행되지 않는지 확인합니다.
- 정상 검색과 정상 로그인은 그대로 성공하는지 확인합니다.
- 오류 응답에 SQL, 파일 경로, stack trace 가 없는지 확인합니다.

SCN-03

약한 인증, 복구와 세션 연쇄

작은 후보군 자동 시도, 약한 비밀번호 등록, 계정 열거, 복구 답변 힌트, 예측 가능한 재설정 token 과 session fixation 을 한 계정 수명주기로 관찰합니다.

항목	내용
업무 시작점	/register.php, /login.php, /forgot.php, /reset.php
취약점 체인	Brute force, weak password policy, insufficient authentication, predictable/reusable reset, weak session IDs, fixation, incomplete logout
BLUEFIX 위치	app/inc/auth.php:16 [BRUTEFORCE-01] app/inc/auth.php:48 [WEAK-PASSWORD-01] app/public/forgot.php:3 [RECOVERY-01] app/public/reset.php:3 [RECOVERY-02] app/inc/bootstrap.php:18 [SESSION-FIXATION-01] app/inc/auth.php:24 [SESSION-WEAK-ID-02] app/inc/auth.php:64 [SESSION-LOGOUT-03]

정상 업무 기준선

- 새 계정을 4 자 이상 비밀번호로 가입하고 로그인합니다.
- victim 계정의 정상 로그인과 로그아웃을 확인합니다.
- forgot 화면에서 올바른 교육용 복구 답변을 입력해 재설정 링크를 사용합니다.

공격 체인

11. 회원가입에서 1234 같은 짧고 흔한 비밀번호가 허용되는지 확인합니다.
12. victim 에 대해 아래의 작은 고정 후보군만 순서대로 시도하고 잠금·지연이 없는지 확인합니다. 대량 시도는 하지 않습니다.
13. forgot 화면에 victim 을 입력하고 답변을 비워 제출하여 계정 존재와 복구 답변 힌트가 노출되는지 확인합니다.
14. 정답을 제출해 발급된 token 을 한 번 사용한 뒤 같은 token 으로 다시 비밀번호를 변경하여 재사용 가능 여부를 확인합니다.
15. /login.php?sid=TRAININGFIXED1234 로 접속한 뒤 로그인하고 VMSESSID 가 공격자가 정한 값으로 유지되는지 확인합니다.

교육용 입력 또는 요청 예시 - SERVER/REDTEAM_HOST는 승인된 사설 범위로 치환

```
# 작은 교육용 후보군 - SERVER는 사설 범위의 VulnMart 만 지정
for p in password 1234 spring2026 admin123; do
  curl -s -o /dev/null -w "$p -> ${http_code}\n" \
    -d "username=victim&password=$p" http://SERVER/login.php
done
```

고정 세션: http://SERVER/login.php?sid=TRAININGFIXED1234

성공 증적과 관찰

- 반복 실패 간 응답 지연·잠금이 없는 기록
- forgot 의 서로 다른 존재/부재 메시지와 답변 힌트
- 동일 reset token 의 두 번째 성공
- 로그인 전후 동일 VMSESSID 와 session_records.source

애플리케이션 로그

auth.login, auth.recovery.issue, auth.recovery.reset, auth.logout 이벤트를 actor 와 session_fingerprint 로 연결합니다.

블루팀 코드 패치

수정 위치

```
app/inc/auth.php:16 [BRUTEFORCE-01]
app/inc/auth.php:48 [WEAK-PASSWORD-01]
app/public/forgot.php:3 [RECOVERY-01]
app/public/reset.php:3 [RECOVERY-02]
app/inc/bootstrap.php:18 [SESSION-FIXATION-01]
app/inc/auth.php:24 [SESSION-WEAK-ID-02]
app/inc/auth.php:64 [SESSION-LOGOUT-03]
```

계정/IP 기반 rate limit과 점진적 지연, 통합 인증 메시지, 12자 이상 정책과 차단 목록을 적용합니다. reset token은 random_bytes 기반 hash를 저장하고 짧은 만료와 1회 사용을 원자적으로 처리합니다. sid 쿼리 수용과 예측 ID 생성은 제거하고 로그인 시 session_regenerate_id(true), 로그아웃 시 세션·쿠키를 완전히 폐기합니다.

방어 구현 예시 - 환경에 맞게 검토하여 적용

```
ini_set('session.use_strict_mode', '1');
// sid 쿼리 처리 삭제

if ($login_ok) {
    session_regenerate_id(true);
    $_SESSION['user_id'] = (int)$user['id'];
}

$rawToken = bin2hex(random_bytes(32));
$stored = hash('sha256', $rawToken);
// token_hash, expires_at, used=0 저장 후 UPDATE ... WHERE used=0 AND expires_at>NOW()
```

동일 공격 재검증과 정상 기능 회귀

- 5회 내외 실패 후 정책에 맞는 지연 또는 일시 제한이 적용되는지 확인합니다.
- 존재·부재 계정의 복구 응답이 동일한지 확인합니다.
- reset token을 재사용하거나 만료 후 사용할 수 없는지 확인합니다.
- 공격자가 지정한 sid가 로그인 후 교체되고, 로그아웃 후 보호 페이지에 접근할 수 없는지 확인합니다.
- 정상 가입, 로그인, 복구와 로그아웃이 유지되는지 확인합니다.

SCN-04

장바구니에서 주문 금액 조작

정상 장바구니와 결제 품을 만든 뒤 클라이언트 hidden total, 서명 없는 할인 쿠키와 무제한 쿠폰 검증을 조작하여 주문 프로세스 검증 누락을 확인합니다.

항목	내용
업무 시작점	/products.php -> /cart.php -> /checkout.php
취약점 체인	Process validation missing, cookie tampering, coupon automation, client-side price trust
BLUEFIX 위치	app/public/checkout.php:4 [CHECKOUT-TRUST-01] app/public/checkout.php:8 [COOKIE-DISCOUNT-01] app/public/coupon.php:3 [COUPON-AUTOMATION-01]

정상 업무 기준선

- 상품 두 개를 장바구니에 담고 수량·표시 합계를 확인합니다.
- TEAM10 같은 정상 쿠폰을 한 번 확인합니다.
- 가짜 카드 값과 배송지로 주문을 완료하고 주문 금액과 송장을 확인합니다.

공격 체인

16. 브라우저 개발자 도구 또는 승인된 프록시로 checkout POST 를 가로칩니다.
17. hidden total_amount 를 100 으로 바꾸고 vm_discount 쿠키를 99 로 설정한 뒤 요청을 보냅니다.
18. 주문 상세와 관리자 주문 화면에서 charged_total 이 서버 상품 합계와 다른지 확인합니다.
19. WELCOME5, TEAM10, BLUE20, ADMIN50 같은 작은 고정 목록을 coupon.php 에 순서대로 요청하여 빈도·사용 제한이 없는지 관찰합니다.

교육용 입력 또는 요청 예시 - SERVER/REDTEAM_HOST 는 승인된 사설 범위로 치환

```
Cookie: vm_discount=99
POST /checkout.php
shipping_address=교육망 테스트 주소&total_amount=100&card_number=4111-1111-1111-1111
GET /coupon.php?code=ADMIN50
```

성공 증적과 관찰

- 원래 장바구니 합계, 변조 POST, 생성 주문의 total_amount
- checkout action log 의 server_total, submitted_total, charged_total, discount_cookie
- coupon_attempts 의 짧은 시간 다수 후보 기록

애플리케이션 로그

scenario_key=order-fraud. 카드 번호는 민감 키 규칙으로 마스킹됩니다.

블루팀 코드 패치

수정 위치

```
app/public/checkout.php:4 [CHECKOUT-TRUST-01]
app/public/checkout.php:8 [COOKIE-DISCOUNT-01]
app/public/coupon.php:3 [COUPON-AUTOMATION-01]
```

주문 트랜잭션 안에서 상품 ID 와 수량을 다시 읽고 가격·재고·할인 자격을 서버 데이터로 재계산합니다. 쿠폰은 사용자별 사용 횟수와 원자적 소비를 기록하며, vm_discount 같은 권한성 쿠키를 제거합니다.

방어 구현 예시 - 환경에 맞게 검토하여 적용

```
$details = cart_details(); // DB 가격으로 재조회
$discount = load_valid_coupon_for_user($u['id']);
$final = calculate_total($details['items'], $discount);

// INSERT order와 stock 차감은 하나의 transaction에서 수행
// POST의 total_amount와 vm_discount 쿠키는 계산에 사용하지 않음
```

동일 공격 재검증과 정상 기능 회귀

- total_amount와 vm_discount를 바꿔도 서버 계산 금액이 유지되는지 확인합니다.
- 유효 쿠폰의 사용 횟수와 만료가 원자적으로 적용되는지 확인합니다.
- 정상 장바구니, 쿠폰, 주문, 재고 차감, 송장 생성이 유지되는지 확인합니다.

SCN-05

다른 고객 주문과 송장 파일 접근

순차 주문 ID, 상태 변경 GET, 파일명 기반 송장 다운로드를 연결하여 객체 단위 권한과 파일 경계의 누락을 확인합니다.

항목	내용
업무 시작점	/orders.php -> /order.php?id=...
취약점 체인	IDOR, insufficient authorization, CSRF on cancel, path traversal/file download
BLUEFIX 위치	app/public/order.php:3 [ORDER-IDOR-01] app/public/order.php:5 [ORDER-CSRF-02] app/public/invoice.php:3 [DOWNLOAD-TRAVERSAL-01] app/public/download.php:2 [DOWNLOAD-TRAVERSAL-02]

정상 업무 기준선

- alice 로 로그인하여 seed 주문 VM-DEMO-1001 과 송장을 확인합니다.
- bob 으로 정상 주문을 하나 만들고 본인 주문 목록에서 확인합니다.

공격 체인

20. bob 으로 로그인한 상태에서 /order.php?id=1 처럼 다른 사용자의 순차 ID 를 요청합니다.
21. 구매자 이메일, 배송지, 품목과 송장 링크가 표시되는지 기록합니다.
22. /order.php?id=1&action=cancel 링크로 GET 요청만으로 상태가 바뀌는지 확인합니다.
23. 송장 file 값을 ../../../../etc/hostname 으로 변경하여 기존 디렉터리 밖 파일을 읽습니다.

교육용 입력 또는 요청 예시 - SERVER/REDTEAM_HOST 는 승인된 사설 범위로 치환

```
GET /order.php?id=1
GET /order.php?id=1&action=cancel
GET /invoice.php?file=../../../../etc/hostname
```

성공 증거와 관찰

- bob 세션에서 표시된 alice 주문·배송 정보
- GET 취소 전후 orders.status
- invoice.download 로그의 file 과 resolved 경로

애플리케이션 로그

scenario_key=order-data-access. order.view 의 owner_id 와 viewer_id 차이가 핵심 증거입니다.

블루팀 코드 패치

수정 위치

```
app/public/order.php:3 [ORDER-IDOR-01]
app/public/order.php:5 [ORDER-CSRF-02]
app/public/invoice.php:3 [DOWNLOAD-TRAVERSAL-01]
app/public/download.php:2 [DOWNLOAD-TRAVERSAL-02]
```

주문 조회는 id 와 current_user_id 를 함께 조건으로 사용하고 관리자 예외를 명시합니다. 상태 변경은 POST+CSRF+소유권+허용 상태를 트랜잭션에서 검증합니다. 송장은 사용자가 제출한 경로 대신 소유 주문의 invoice ID 를 조회하고 realpath prefix 를 확인합니다.

방어 구현 예시 - 환경에 맞게 검토하여 적용

```

$stmt = db()->prepare(
    'SELECT * FROM orders WHERE id=? AND (user_id=? OR ?=1)'
);
$stmt->execute([$id, $u['id'], $u['role']=='admin' ? 1 : 0]);

if ($_SERVER['REQUEST_METHOD'] !== 'POST' || !verify_csrf()) {
    http_response_code(403); exit;
}

$base = realpath('/var/lib/vulnmart/invoices');
$path = realpath($base . '/' . basename($dbInvoiceName));
if ($path === false || !str_starts_with($path, $base . DIRECTORY_SEPARATOR)) exit;

```

동일 공격 재검증과 정상 기능 회귀

- bob 이 alice 주문을 403/404 로 받는지 확인합니다.
- 링크/img GET 으로 주문 취소가 되지 않는지 확인합니다.
- 정상 소유자의 POST 취소와 송장 다운로드가 유지되는지 확인합니다.
- ../와 절대 경로 입력이 거부되는지 확인합니다.

SCN-06

고객센터 URL 미리보기에서 내부망 피벗

로그인 사용자가 고객센터 문의에 참고 URL 을 넣는 정상 기능을 이용해 PHP 서버가 loopback 메타데이터와 로컬 file scheme 을 대신 요청하도록 만듭니다.

항목	내용
업무 시작점	/support.php
취약점 체인	SSRF, scheme abuse, loopback/private IP access, internal information disclosure
BLUEFIX 위치	app/public/support.php:4 [SSRF-PREVIEW-01] app/internal/metadata.php:3 [INTERNAL-METADATA-01]

정상 업무 기준선

- alice 로 로그인하여 참고 URL 없이 일반 문의를 등록합니다.
- 허가된 팀 웹 문서 URL 이 있다면 미리보기 길이와 오류 처리를 확인합니다.

공격 체인

24. 참고 URL 에 `http://127.0.0.1:8008/metadata.php` 를 입력해 문의를 접수합니다.
25. 최근 문의의 URL 미리보기에서 `supplier_api_key`, `backup_path`, `admin_hint` 가 표시되는지 확인합니다.
26. 별도 문의에 `file:///etc/hostname` 을 입력하여 file scheme 도 허용되는지 확인합니다.
27. `support.ticket.create` 로그의 url, `preview_bytes` 와 내부 Nginx access log 시간을 연결합니다.

교육용 입력 또는 요청 예시 - SERVER/REDTEAM_HOST는 승인된 사설 범위로 치환

```
http://127.0.0.1:8008/metadata.php
file:///etc/hostname
```

성공 증거와 관찰

- `support_tickets.preview_result` 의 내부 JSON 또는 hostname
- `vulnmart_internal_access.log` 의 127.0.0.1 요청
- `action_logs` 의 `reference_url` 과 `preview_bytes`

애플리케이션 로그

`scenario_key=support-pivot`. SSRF 입력 URL 은 기록되므로 팀 범위의 로컬 주소만 사용합니다.

블루팀 코드 패치

수정 위치

```
app/public/support.php:4 [SSRF-PREVIEW-01]
app/internal/metadata.php:3 [INTERNAL-METADATA-01]
```

지원되는 scheme 을 http/https 로 제한한 뒤 승인 도메인 allowlist, DNS 해석 결과의 loopback/private/link-local 거부, redirect 마다 재검증, 응답 크기와 timeout, egress proxy 를 조합합니다. 내부 메타데이터 자체에서도 비밀 제거와 독립 인증을 적용합니다.

방어 구현 예시 - 환경에 맞게 검토하여 적용

```

$parts = parse_url($url);
if (!in_array($parts['scheme'] ?? '', ['https'], true)) reject();
if (!in_array(strtolower($parts['host'] ?? ''), $approvedHosts, true)) reject();
$ips = dns_get_record($parts['host'], DNS_A | DNS_AAAA);
foreach ($ips as $ip) {
    if (is_private_loopback_or_linklocal($ip)) reject();
}
// redirect off, response size limit, egress proxy 사용

```

동일 공격 재검증과 정상 기능 회귀

- 127.0.0.1, localhost, private IP, file://, redirect 경유가 거부되는지 확인합니다.
- 승인된 정상 문서 URL 은 제한된 크기로 미리보기가 되는지 확인합니다.
- 내부 endpoint 에 민감 key 와 관리자 힌트가 남지 않는지 확인합니다.

SCN-07

프로필 이미지에서 서버 실행 경로
확보

프로필 이미지 업로드가 원본 확장자와 웹 실행 경로를 신뢰하고, 도움말 include 가 상대 경로를 받아들이는 점을 결합합니다.

항목	내용
업무 시작점	/profile.php -> /uploads/... -> /help.php
취약점 체인	Malicious file upload, executable upload, Local File Inclusion, download traversal
BLUEFIX 위치	app/public/profile.php:16 [UPLOAD-EXEC-01] deploy/nginx-vulnmart.conf:27 [UPLOAD-EXEC-NGINX-02] app/public/help.php:5 [LFI-HELP-01] app/public/download.php:2 [DOWNLOAD-TRAVERSAL-02]

정상 업무 기준선

- alice 로 PNG 또는 JPEG 프로필 이미지를 올리고 프로필에 표시되는지 확인합니다.
- 배송·반품·회원 도움말 문서가 정상 표시되는지 확인합니다.

공격 체인

28. 아래 한 줄만 포함한 avatar-proof.php 를 만들어 프로필 이미지로 업로드합니다.
29. /uploads/avatar-proof.php 를 직접 열어 PHP 가 실행되는지 확인합니다.
30. /help.php?article=../uploads/avatar-proof.php 를 열어 도움말 include 흐름에서도 실행되는지 확인합니다.
31. 일반 다운로드의 file 값에 상위 경로를 넣어 파일 경계 누락을 추가 확인합니다.

교육용 입력 또는 요청 예시 - SERVER/REDTEAM_HOST 는 승인된 사설 범위로 치환

```
<?php echo 'VULNMART-UPLOAD-OK'; ?>

/uploads/avatar-proof.php
/help.php?article=../uploads/avatar-proof.php
/download.php?file=../../../../../../etc/hostname
```

성공 증거와 관찰

- uploads 테이블의 original_name/stored_name/mime_type
- 웹 루트의 업로드 파일과 실행 응답 문자열
- help.view 로그의 article 값

애플리케이션 로그

scenario_key=profile-foothold. 업로드 원본명은 detail_data 에 기록됩니다.

블루팀 코드 패치

수정 위치

```
app/public/profile.php:16 [UPLOAD-EXEC-01]
deploy/nginx-vulnmart.conf:27 [UPLOAD-EXEC-NGINX-02]
app/public/help.php:5 [LFI-HELP-01]
app/public/download.php:2 [DOWNLOAD-TRAVERSAL-02]
```

finfo 기반 이미지 MIME allowlist, 이미지 재인코딩, 랜덤 서버 파일명, 웹 루트 밖 저장을 사용합니다. Nginx 에서 /uploads 를 ^~ location 으로 분리하고 .php 를 절대 FastCGI 로 보내지 않습니다. 도움말은 고정 ID 매핑만 include 합니다.

방어 구현 예시 - 환경에 맞게 검토하여 적용

```
// PHP
$allowed = ['image/png'=>'png', 'image/jpeg'=>'jpg'];
$mime = (new finfo(FILEINFO_MIME_TYPE))->file($tmp);
if (!isset($allowed[$mime])) reject();
$name = bin2hex(random_bytes(16)) . '.' . $allowed[$mime];
$target = '/var/lib/vulnmart/uploads/' . $name;

# Nginx
location ^~ /uploads/ {
    try_files $uri =404;
    location ~ /\.php$ { return 403; }
}

$articles = ['shipping'=>'shipping.php', 'returns'=>'returns.php'];
```

동일 공격 재검증과 정상 기능 회귀

- .php, 이중 확장자, 위조 MIME 이 거부되는지 확인합니다.
- 정상 PNG/JPEG 가 랜덤명으로 저장되고 표시되는지 확인합니다.
- /uploads 아래 PHP 가 실행되지 않는지 확인합니다.
- 도움말의 ../ 입력이 거부되고 정상 문서 3 개가 유지되는지 확인합니다.

SCN-08

상품 비교와 마케팅 캠페인의 브라우저 공격

URL hash 기반 비교 설정, 위험한 JavaScript 객체 병합, innerHTML, same-origin JSONP 와 파트너 이동 링크를 연결해 브라우저 신뢰 경계를 검증합니다.

항목	내용
업무 시작점	/compare.php -> /campaign.php -> /out.php
취약점 체인	Prototype pollution, DOM XSS, CSP bypass via JSONP, open HTTP redirect
BLUEFIX 위치	app/public/compare.php:6 [PROTOTYPE-POLLUTION-02] app/public/compare.php:12 [DOM-XSS-01] app/public/campaign.php:6 [CSP-JSONP-01] app/public/api/promo.php:2 [CSP-JSONP-01] app/public/out.php:3 [OPEN-REDIRECT-01]

정상 업무 기준선

- 상품 비교 페이지에서 기본 상품과 메시지를 확인합니다.
- 캠페인 페이지에서 Blue Team Week 프로모션이 표시되는지 확인합니다.
- 정상 파트너 링크의 이동 동작을 확인합니다.

공격 체인

- compare 페이지 콘솔에서 아래 hash JSON 을 설정하고 reload 합니다.
- Object.prototype 에 premium 이 생기고 message 의 img onerror 가 실행되는지 body dataset 으로 확인합니다.
- /campaign.php?callback=alert(document.domain)//를 열어 self JSONP 가 CSP 아래에서도 실행되는지 확인합니다.
- /out.php?url=http://REDTEAM_HOST/landing 처럼 팀이 통제하는 사설 호스트를 목적으로 지정해 임의 redirect 를 확인합니다.

교육용 입력 또는 요청 예시 - SERVER/REDTEAM_HOST 는 승인된 사설 범위로 치환

```
location.hash = encodeURIComponent(
  '{"__proto__":{"premium":true},' +
  '"message":"<img src=x onerror=\"document.body.dataset.vm=\'dom-xss\'\">"}'
);
location.reload();

/campaign.php?callback=alert(document.domain)//
/out.php?url=http://REDTEAM_HOST/landing
```

성공 증적과 관찰

- compare-status 의 inherited premium=true 와 DOM dataset
- CSP header 와 same-origin JSONP 실행
- out.php 의 302 Location header 와 partner.redirect 로그

애플리케이션 로그

browser-campaign 관련 partner.redirect 가 기록됩니다. DOM-only 동작은 필요 시 telemetry.php 의 교육용 event 로 별도 수집할 수 있습니다.

블루팀 코드 패치

수정 위치

app/public/compare.php:6 [PROTOTYPE-POLLUTION-02]
 app/public/compare.php:12 [DOM-XSS-01]
 app/public/campaign.php:6 [CSP-JSONP-01]
 app/public/api/promo.php:2 [CSP-JSONP-01]
 app/public/out.php:3 [OPEN-REDIRECT-01]

재귀 병합에서 __proto__, prototype, constructor 를 거부하고 schema 기반 parser 를 사용합니다. DOM 출력은 textContent 를 사용합니다. JSONP 를 application/json + fetch 로 교체하고 CSP nonce/hash 를 검토합니다. redirect 는 서버측 destination ID allowlist 로 매핑합니다.

방어 구현 예시 - 환경에 맞게 검토하여 적용

```
const blocked = new Set(['__proto__', 'prototype', 'constructor']);
for (const [key, value] of Object.entries(source)) {
  if (blocked.has(key)) throw new Error('invalid key');
  // schema 에 정의된 키만 복사
}
sink.textContent = config.message || config.title;

// promo.php 는 application/json 반환, campaign.php 는 fetch('/api/promo.php')
$destinations = ['warranty'=>'/partner-notice.php'];
```

동일 공격 재검증과 정상 기능 회귀

- 위험 key 가 거부되고 Object.prototype 이 변하지 않는지 확인합니다.
- HTML message 가 문자로 표시되고 정상 비교 메시지는 유지되는지 확인합니다.
- callback 파라미터가 JavaScript 로 실행되지 않는지 확인합니다.
- 허용 destination 은 이동하고 임의 외부 URL 은 거부되는지 확인합니다.

SCN-09

노출된 관리자 표면과 역할 쿠키로 관리자 진입

robots.txt와 고정 경로로 관리자 콘솔을 발견하고, 서명되지 않은 vm_role 쿠키를 권한 근거로 신뢰하는 코드를 이용해 관리자 기능과 평문 인증정보에 접근합니다.

항목	내용
업무 시작점	/robots.txt -> /admin/
취약점 체인	Admin page exposure, cookie tampering, insufficient authentication/authorization, admin information disclosure
BLUEFIX 위치	app/public/robots.txt:6 [ADMIN-EXPOSURE-01] app/inc/functions.php:56 [AUTHZ-COOKIE-01] app/public/admin/users.php:5 [INFO-DISCLOSURE-ADMIN-02]

정상 업무 기준선

- admin/admin123으로 로그인하여 관리자 대시보드, 상품, 주문, 회원, 로그를 확인합니다.
- 일반 alice계정은 /admin/에서 403을 받는지 기준선을 기록합니다.

공격 체인

36. /robots.txt를 열어 /admin/, /exports/, /health.php, /api/ 단서를 수집합니다.
37. 로그아웃 상태 또는 일반 사용자 브라우저 콘솔에서 document.cookie="vm_role=admin; path=/"를 설정합니다.
38. /admin/과 /admin/users.php를 열어 관리자 화면과 교육용 평문 비밀번호·복구 답변이 노출되는지 확인합니다.
39. authorization.denied와 이후 관리자 요청의 actor/user_role 불일치를 로그에서 비교합니다.

교육용 입력 또는 요청 예시 - SERVER/REDTEAM_HOST는 승인된 사설 범위로 치환

```
document.cookie = 'vm_role=admin; path=/';  
location.href = '/admin/';
```

성공 증거와 관찰

- robots.txt 내용
- 브라우저 Cookie의 vm_role=admin
- 관리자 화면의 cookie-admin 또는 일반 사용자 세션
- 회원 화면의 password_plain/recovery_answer

애플리케이션 로그

관리자 요청은 http.request로 남지만 pseudo cookie-admin은 user_id 0으로 나타날 수 있어 탐지 가설이 됩니다.

블루팀 코드 패치

수정 위치

```
app/public/robots.txt:6 [ADMIN-EXPOSURE-01]  
app/inc/functions.php:56 [AUTHZ-COOKIE-01]  
app/public/admin/users.php:5 [INFO-DISCLOSURE-ADMIN-02]
```

robots.txt를 접근제어로 보지 않고 모든 관리자 endpoint에서 DB 세션의 role을 검증합니다. 권한성 client cookie를 제거하고 관리자 재인증/MFA를 검토합니다. DB schema에서 password_plain을 삭제하고 복구 답변은 단방향 검증값만 저장합니다.

방어 구현 예시 - 환경에 맞게 검토하여 적용

```
function require_admin(): array {
    $u = current_user();
    if (!$u || !hash_equals('admin', (string)$u['role'])) {
        log_event('authorization.denied', 'denied');
        http_response_code(403); exit;
    }
    return $u;
}
// vm_role 쿠키 생성 · 신뢰 제거
// users.password_plain 컬럼과 SELECT 제거
```

동일 공격 재검증과 정상 기능 회귀

- vm_role 쿠키만으로 어떤 관리자 URL 에도 접근할 수 있는지 확인합니다.
- admin DB 역할 세션은 정상적으로 모든 관리 기능을 사용할 수 있는지 확인합니다.
- 회원 화면과 DB 에 평문 비밀번호·복구 답변 원문이 없는지 확인합니다.

SCN-10

탈취한 관리자 권한의 서버 측 코드·명령 영향

관리자 업무 기능이 동적 PHP 식과 셸 문자열을 실행하는 구조를 이용해, 관리자 계정 탈취가 서버 측 실행으로 확대되는 경로를 무해한 증명으로 확인합니다.

항목	내용
업무 시작점	/admin/pricing.php, /admin/diagnostics.php
취약점 체인	Code Injection, Command Injection
BLUEFIX 위치	app/public/admin/pricing.php:4 [CODE-INJECTION-01] app/public/admin/diagnostics.php:4 [COMMAND-INJECTION-01]

정상 업무 기준선

- 관리자 가격 공식에 $\max(3000, \$subtotal * 0.03)$ 을 넣어 정상 배송비를 계산합니다.
- 공급사 이름 해석 진단에 localhost 또는 127.0.0.1 을 넣어 getent 결과를 확인합니다.

공격 체인

40. 가격 공식에 `gethostname()`을 넣어 서버에서 PHP 함수가 평가되는지 확인합니다.
41. 공급사 host 에 127.0.0.1; id 를 넣어 셸 구문자가 추가 명령을 실행하는지 확인합니다.
42. 출력, `admin.pricing.evaluate`, `admin.supplier.ping` 이벤트를 저장합니다.
43. 파일 쓰기, 외부 연결, 권한 변경, 서비스 중단 명령은 사용하지 않습니다.

교육용 입력 또는 요청 예시 - SERVER/REDTEAM_HOST는 승인된 사설 범위로 치환

가격 공식: `gethostname()`
공급사 host: 127.0.0.1; id

성공 증적과 관찰

- 가격 계산 결과의 server hostname
- 진단 출력의 uid/gid
- `action_logs`의 `formula/host`, `output length`, `actor`

애플리케이션 로그

`scenario_key=admin-rce`. 입력은 관리자 행동 로그에 남아 코드 실행 이후 조사 단서가 됩니다.

블루팀 코드 패치

수정 위치

app/public/admin/pricing.php:4 [CODE-INJECTION-01]
app/public/admin/diagnostics.php:4 [COMMAND-INJECTION-01]

`eval`을 제거하고 shipping policy ID 를 고정된 서버 함수에 매핑합니다. 이름 해석은 `shell_exec` 없이 `dns_get_record/gethostbyname` 또는 소켓 API 를 사용하고, 허용 형식과 업무 대상 allowlist 를 분리합니다.

방어 구현 예시 - 환경에 맞게 검토하여 적용

```

$policies = [
  'standard' => fn(float $subtotal): float => max(3000, $subtotal * 0.03),
  'free_over_100k' => fn(float $subtotal): float => $subtotal >= 100000 ? 0 : 3000,
];
$result = $policies[$policyId]($subtotal);

if (!filter_var($host, FILTER_VALIDATE_IP) &&
    !preg_match('/^[a-z0-9.-]+\z/i', $host)) reject();
$result = dns_get_record($host, DNS_A | DNS_AAAA);

```

동일 공격 재검증과 정상 기능 회귀

- `gethostname()`, PHP 문법, 세미콜론 입력이 데이터로 거부되는지 확인합니다.
- `127.0.0.1; id`가 추가 명령을 실행하지 않는지 확인합니다.
- 고정 배송 정책과 정상 DNS/주소 조회가 유지되는지 확인합니다.

SCN-11

백오피스 상품 API와 자동화 악용

정상 상품 화면이 의존하는 백오피스 API가 인증 없이 PUT·DELETE를 처리하고, 쿠폰 API가 빈도 제한 없이 상태를 변경하는 점을 이용합니다.

항목	내용
업무 시작점	/api/products.php, /coupon.php
취약점 체인	Unnecessary HTTP methods, missing authentication/authorization, automation attacks
BLUEFIX 위치	app/public/api/products.php:4 [METHOD-ABUSE-01] app/public/coupon.php:3 [COUPON-AUTOMATION-01]

정상 업무 기준선

- GET /api/products.php로 상품 JSON을 확인합니다.
- 관리자 상품 화면에서 정상 재고와 활성 상태를 확인합니다.

공격 체인

44. 인증 cookie 없이 PUT 요청으로 상품 1의 stock을 1로 변경합니다.
45. 인증 없이 DELETE /api/products.php?id=2를 보내 상품을 비활성화합니다.
46. 상품 목록과 관리자 화면에서 변경이 즉시 반영되는지 확인합니다.
47. 짧은 고정 쿠폰 목록을 순차 요청하여 상태 cookie와 coupon_attempts를 확인합니다.

교육용 입력 또는 요청 예시 - SERVER/REDTEAM_HOST는 승인된 사설 범위로 치환

```
curl -X PUT -H 'Content-Type: application/json' \
  --data '{"id":1,"stock":1}' http://SERVER/api/products.php

curl -X DELETE 'http://SERVER/api/products.php?id=2'

for c in WELCOMES TEAM10 BLUE20 ADMIN50; do
  curl -s "http://SERVER/coupon.php?code=$c"; echo
done
```

성공 증적과 관찰

- PUT/DELETE 응답 JSON
- products.stock/active의 변경
- api.products.put/delete 행동 로그와 익명 actor
- coupon_attempts의 시간 집중

애플리케이션 로그

scenario_key=api-abuse와 order-fraud를 함께 필터해 API 상태 변경과 쿠폰 자동화를 비교합니다.

블루팀 코드 패치

수정 위치

app/public/api/products.php:4 [METHOD-ABUSE-01]
app/public/coupon.php:3 [COUPON-AUTOMATION-01]

API endpoint 시작에서 require_admin을 적용하고 필요한 Method만 허용합니다. 상태 변경은 CSRF 또는 별도 API token/Origin 정책, 객체 권한, 입력 schema, 감사 로그를 사용합니다. 쿠폰은 rate limit과 서버측 1회 사용을 적용합니다.

방어 구현 예시 - 환경에 맞게 검토하여 적용

```
$admin = require_admin();
if (!in_array($method, ['GET', 'PUT'], true)) {
    header('Allow: GET, PUT'); http_response_code(405); exit;
}
if ($method === 'PUT' && !verify_csrf()) { http_response_code(403); exit; }
$data = json_decode(request_body(), true, flags: JSON_THROW_ON_ERROR);
// id, stock, price schema와 범위 검증
```

동일 공격 재검증과 정상 기능 회귀

- 익명 PUT/DELETE 가 401/403/405 로 거부되는지 확인합니다.
- 관리자 인증과 검증 token 이 있는 허용 PUT 만 성공하는지 확인합니다.
- 정상 상품 GET 과 관리자 상품 관리가 유지되는지 확인합니다.
- 짧은 시간 반복 쿠폰 요청이 제한되고 정상 1 회 사용은 성공하는지 확인합니다.

SCN-12

운영 노출면에서 공격 단서 수집

HTTP 평문, 디렉터리 인덱싱, health endpoint, export 파일, Nginx 배너와 상세 SQL 오류를 연결해 공격 전 정찰과 블루팀 노출면 검토를 수행합니다.

항목	내용
업무 시작점	/health.php, /exports/, /robots.txt, HTTP headers
취약점 체인	Plaintext transport, directory indexing, information disclosure, verbose error pages, administrator exposure
BLUEFIX 위치	deploy/nginx-vulnmart.conf:12 [PLAINTEXT-HTTP-01] deploy/nginx-vulnmart.conf:19 [DIRECTORY-INDEX-01] app/public/exports/README.txt:3 [DIRECTORY-INDEX-01] app/public/health.php:3 [INFO-DISCLOSURE-01] deploy/nginx-main.conf:16 [INFO-DISCLOSURE-NGINX-03] app/public/search.php:13 [ERROR-DISCLOSURE-03] app/public/robots.txt:6 [ADMIN-EXPOSURE-01]

정상 업무 기준선

- 메인 쇼핑몰과 health가 HTTP에서 응답하는지 확인합니다.
- 운영자가 정상 상품 export를 준비하는 업무 맥락을 확인합니다.

공격 체인

48. /exports/를 열어 autoindex와 users-export.csv, runtime-notes.txt를 확인합니다.
49. /health.php에서 PHP, MariaDB, config 경로, session path, server software를 수집합니다.
50. curl -I로 Server 헤더를 확인하고 /search.php?q='로 상세 SQL 오류를 수집합니다.
51. 승인된 관찰 구간에서 HTTP 로그인 또는 폼 요청이 암호화되지 않는다는 점을 패킷 수준으로 확인합니다. 실제 자격증명은 사용하지 않습니다.

교육용 입력 또는 요청 예시 - SERVER/REDTEAM_HOST는 승인된 사설 범위로 치환

```
curl -i http://SERVER/health.php
curl -i http://SERVER/exports/
curl -I http://SERVER/
http://SERVER/search.php?q='
```

성공 증적과 관찰

- autoindex HTML과 export 파일 목록
- health JSON의 version/path/database/session 값
- Server header와 SQL 오류
- 가짜 계정의 HTTP 요청이 평문인 캡처

애플리케이션 로그

HTTP 정찰은 Nginx access log와 PHP http.request를 함께 비교할 수 있습니다. health 자체도 PHP DB 연결을 사용합니다.

블루팀 코드 패치

수정 위치

```
deploy/nginx-vulnmart.conf:12 [PLAINTEXT-HTTP-01]
deploy/nginx-vulnmart.conf:19 [DIRECTORY-INDEX-01]
app/public/exports/README.txt:3 [DIRECTORY-INDEX-01]
app/public/health.php:3 [INFO-DISCLOSURE-01]
deploy/nginx-main.conf:16 [INFO-DISCLOSURE-NGINX-03]
app/public/search.php:13 [ERROR-DISCLOSURE-03]
app/public/robots.txt:6 [ADMIN-EXPOSURE-01]
```

export는 웹 루트 밖에 저장하고 인증된 다운로드 handler 만 제공합니다. health는 최소 ok/status 만 반환하고 server_tokens off를 적용합니다. 사용자 오류는 request-id와 일반 메시지로 바꿉니다. 사실 PKI 또는 팀 인증서로 TLS server를 만들고 HTTP를 HTTPS로 redirect 합니다.

방어 구현 예시 - 환경에 맞게 검토하여 적용

```
# Nginx
server_tokens off;
location /exports/ { autoindex off; return 404; }

server {
    listen 443 ssl;
    ssl_certificate      /etc/pki/tls/certs/vulnmart.crt;
    ssl_certificate_key  /etc/pki/tls/private/vulnmart.key;
    # 기존 PHP location
}
server { listen 80; return 301 https://$host$request_uri; }

// health.php
echo json_encode(['ok' => true]);
```

동일 공격 재검증과 정상 기능 회귀

- /exports/가 목록을 표시하지 않고 승인된 export 다운로드만 가능한지 확인합니다.
- health에 version, path, DB 이름, session path가 없는지 확인합니다.
- Server banner가 축소되고 상세 예외가 사용자에게 보이지 않는지 확인합니다.
- HTTPS에서 정상 로그인·구매가 작동하고 HTTP가 redirect되는지 확인합니다.

Part V. 방어 구현 참조

26. 공통 방어 패턴

SQL 과 인증

- SQL 문법과 사용자 데이터를 분리하는 준비문을 기본으로 사용합니다.
- 비밀번호 원문 컬럼을 제거하고 password_hash/password_verify 를 사용합니다.
- 오류는 내부 request-id 와 구조화 로그로 남기고 사용자 응답은 일반화합니다.

출력과 브라우저

- HTML text, attribute, URL, JavaScript 등 출력 컨텍스트마다 맞는 인코딩을 적용합니다.
- DOM 에는.textContent 를 우선하고 innerHTML 이 필요한 경우 검증된 sanitizer 와 엄격한 정책을 사용합니다.
- CSP 는 XSS 방어의 보조 계층이며 same-origin JSONP 같은 script gadget 을 제거해야 합니다.

세션과 상태 변경

- 로그인 성공 시 session_regenerate_id(true), strict mode, Secure/HttpOnly/SameSite cookie 를 적용합니다.
- 상태 변경은 POST/PUT 등 명시된 Method, CSRF/Origin, 재인증, 객체 소유권과 허용 상태를 함께 검증합니다.

파일과 URL

- 업로드는 MIME allowlist, 재인코딩, 랜덤명, 웹 루트 밖 저장과 서버 실행 차단을 조합합니다.
- 다운로드·include 는 사용자 경로 대신 서버측 ID 매핑과 realpath prefix 를 사용합니다.
- SSRF 는 scheme/host allowlist, DNS/IP 재검증, redirect 통제와 egress 정책을 함께 사용합니다.

관리자와 서버 실행

- 모든 관리자 endpoint 가 서버 세션의 역할을 검증하며 client 역할 쿠키를 권한 근거로 사용하지 않습니다.
- eval 과 shell 문자열 결합은 제거하고 고정 정책 함수, 라이브러리 API, schema 검증으로 대체합니다.

27. BLUEFIX 전체 인덱스

Marker	최초 배포 위치	방어 변경의 핵심
SESSION-FIXATION-01	app/inc/bootstrap.php:18	sid 쿼리 제거, strict mode, 로그인 후 session regenerate
LOGIN-SQLI-01	app/inc/auth.php:7	준비문과 password_verify
BRUTEFORCE-01	app/inc/auth.php:16	계정/IP rate limit 과 지연
SESSION-WEAK-ID-02	app/inc/auth.php:24	예측 ID 생성 제거
SESSION-LOGOUT-03	app/inc/auth.php:64	세션·쿠키 완전 폐기
WEAK-PASSWORD-01	app/inc/auth.php:48	길이·차단목록·재사용·MFA
AUTHZ-COOKIE-01	app/inc/functions.php:56	DB 역할만 신뢰
SEARCH-SQLI-01	app/public/search.php:6	LIKE 준비문
SEARCH-XSS-02	app/public/search.php:20	HTML 인코딩
ERROR-DISCLOSURE-03	app/public/search.php:13	일반 오류+request-id
STORED-XSS-01	app/public/post_new.php:2; app/public/post.php:3	출력 인코딩 또는 sanitizer
STORED-XSS-REVIEW-01	app/public/review_submit.php:3	후기 저장·출력 정책
STORED-XSS-REVIEW-02	app/public/product.php:4	후기 출력 인코딩
CSRF-PASSWORD-01	app/public/profile.php:6	POST+CSRF+현재 암호+재인증
UPLOAD-EXEC-01	app/public/profile.php:16	MIME allowlist, 랜덤명, 웹 루트 밖
UPLOAD-EXEC-NGINX-02	deploy/nginx-vulnmart.conf:27	uploads PHP 실행 차단

Marker	최초 배포 위치	방어 변경의 핵심
RECOVERY-01	app/public/forgot.php:3	통합 메시지와 random 1 회 token
RECOVERY-02	app/public/reset.php:3	만료·used 원자 검증
CHECKOUT-TRUST-01	app/public/checkout.php:4	서버 가격·재고 재계산
COOKIE-DISCOUNT-01	app/public/checkout.php:8	할인 상태 서버 저장
COUPON-AUTOMATION-01	app/public/coupon.php:3	rate limit 과 원자적 사용
ORDER-IDOR-01	app/public/order.php:3	소유권 조건
ORDER-CSRF-02	app/public/order.php:5	POST+CSRF+소유 상태
DOWNLOAD-TRAVERSAL-01	app/public/invoice.php:3	DB 파일 ID 와 realpath prefix
DOWNLOAD-TRAVERSAL-02	app/public/download.php:2	고정 파일 매핑
LFI-HELP-01	app/public/help.php:5	고정 문서 매핑
SSRF-PREVIEW-01	app/public/support.php:4	목적지 allowlist 와 IP 재검증
INTERNAL-METADATA-01	app/internal/metadata.php:3	내부 인증과 비밀 최소화
PROTOTYPE-POLLUTION-02	app/public/compare.php:6	위험 key 거부와 schema
DOM-XSS-01	app/public/compare.php:12	textContent
CSP-JSONP-01	app/public/campaign.php:6; app/public/api/promo.php:2	JSONP 제거
OPEN-REDIRECT-01	app/public/out.php:3	destination ID allowlist
METHOD-ABUSE-01	app/public/api/products.php:4	admin auth, Method, CSRF
CODE-INJECTION-01	app/public/admin/pricing.php:4	eval 제거, 정책 함수 매핑
COMMAND-INJECTION-01	app/public/admin/diagnostics.php:4	shell 제거, API 사용
DIRECTORY-INDEX-01	deploy/nginx-vulnmart.conf:19; app/public/exports/README.txt:3	autoindex off, 웹 루트 밖
PLAINTEXT-HTTP-01	deploy/nginx-vulnmart.conf:12	TLS 와 HTTP redirect
INFO-DISCLOSURE-01	app/public/health.php:3	최소 health
INFO-DISCLOSURE-ADMIN-02	app/public/admin/users.php:5	평균 인증정보 제거
INFO-DISCLOSURE-NGINX-03	deploy/nginx-main.conf:16	server_tokens off
ADMIN-EXPOSURE-01	app/public/robots.txt:6	경로 은닉 대신 강한 인증

28. 정상 기능 회귀 체크리스트

영역	패치 후 반드시 확인할 정상 동작
계정	회원가입, 정상 로그인, 로그아웃, 비밀번호 변경, 1 회 복구
상품	목록, 상세, 검색, 후기, 비교, 캠페인
주문	장바구니 수량, 쿠폰, 서버 계산, 주문, 재고, 송장
권한	본인 주문, 관리자 상품·주문·회원·로그, 일반 사용자 403
고객센터	일반 문의, 승인 URL 미리보기 또는 명확한 미지원 처리
파일	정상 이미지 업로드, 허용 도움말과 다운로드
관찰	정상·거부 이벤트, actor, request path, scenario_key, 민감값 마스킹
운영	nginx -t, php-fpm -t, mariadb/php-fpm/nginx active, verify.sh

공통 기술 검증

```
php -l /var/www/vulnmart/public/수정파일.php
sudo nginx -t
sudo php-fpm -t
sudo systemctl reload php-fpm nginx
sudo ./verify.sh
```

29. 참고 문서

아래 문서는 환경 설계와 방어 구현을 더 깊게 검토할 때 참고할 수 있는 공식 또는 프로젝트 문서입니다. 특정 구성의 강제 기준이 아니라 원리와 선택지를 비교하는 출발점으로 사용하십시오.

문서	주소
Rocky Linux Documentation	https://docs.rockylinux.org/
Red Hat Enterprise Linux 9 - PHP and web server guidance	https://docs.redhat.com/en/documentation/red_hat_enterprise_linux/9/
Nginx FastCGI module	https://nginx.org/en/docs/http/nginx_http_fastcgi_module.html
PHP manual - FPM and sessions	https://www.php.net/manual/
MariaDB Server documentation	https://mariadb.com/docs/server/
OWASP Web Security Testing Guide	https://owasp.org/www-project-web-security-testing-guide/
OWASP Cheat Sheet Series	https://cheatsheetseries.owasp.org/

마무리

시나리오 완료 후에는 공격 성공 여부뿐 아니라 어떤 신뢰 경계를 바꾸었는지, 정상 업무가 유지되는지, 행동 로그가 조사에 충분한지, 환경을 원래 상태로 복구할 수 있는지를 함께 기록하십시오.