

막연함을 확신으로 바꾼 주니어의 9개월

크래시에서 프롬프트,
그리고 에이전트 AI까지



여서인 | 화이트햇 스쿨 3기

This talk weaves crashes, prompts, and orchestration into a single security narrative.
It's a condensed record of how I learned security through hands-on practice, iteration, and collaboration.
My perspective has expanded from vulnerabilities in systems to vulnerabilities in workflows
Tools make agents powerful; gateways make them governable.

JANUARY 27, 2026 OWASP SEOUL CHAPTER

The Open Worldwide Application Security Project (OWASP®) is a 501(c)(3) worldwide not-for-profit charitable organization focused on improving the security of software. Our mission is to make software security visible so individuals and organizations can make informed decisions. OWASP is uniquely positioned to provide impartial, practical information about AppSec to individuals, corporations, universities, government agencies, and other organizations worldwide.

Contents



INTRO
- 막연함을 확신으로

첫 번째 도전
- SYSTEM HACKING

두 번째 도전
- LLM SAFETY CHALLENGE

새로운 시대
- AI의 진화

OWASP TOP 10
FOR AGENTIC
APPLICATIONS

OUTRO
- 확신을 다음 도전으로

Summary



KEY MESSAGE

Agent 시대 보안의 목표는
“행동의 통제를 가능하게 하는 것”

리스크는 연결 순간에 증가
: 신뢰 전염 · 권한 점프 · 흐름 변조

#AGENTSECURITY

#ORCHESTRATION

RISK LENS

오케스트레이션에서 위험이 커지는 순간 3가지
Trust Contagion / Privilege Jump /
Workflow Tampering

OWASP Top 10
for Agentic Applications

#RISK

#ASI01 ~ ASI10

ACTIONABLE CONTROLS

위협 제동 3가지 레버
Zero-Trust I/O
Permission Gate+HITL
Workflow integrity + Audit

Central Policy Enforcement (PEP)
- MCP Gateway

#DEFENSE

#MCPGATEWAY

INTRO

- 막연함을 확신으로



MEET SECURITY



MARCH 2025
3RD WHITEHAT SCHOOL

비전공자로 시작한 출발선
무엇부터 해야 할지 막막했습니다

그래서 저는 끝까지 지켜낼 신념을 새겼습니다

배우기 · 기록하기 · 끝까지 도전하기
절대로 포기하지 않기

첫 번째 도전 – SYSTEM HACKING



FUZZING을 활용한 취약점 분석

첫 번째 도전 - SYSTEM HACKING



FUZZING을 활용한 취약점 분석



FIRST SECURITY TEAM PROJECT

'모르는 것뿐'이었던 시스템 해킹

낯선 용어와 어려운 개념들

모르면 배우고 밤을 새서라도
끝까지 해낸다는 진행 방식 확립

국내 소프트웨어의 서드파티 라이브러리
취약점 발견



TEAMWORK

협업의 핵심 '프로세스'

역할 분담 + 빠른 공유 = 속도 증가

Logs make legends

Log(기록)가 쌓여 실력을 만들고,
큰 성과가 발생



ZDI BUG BOUNTY

취약점 제보 과정 경험

ZDI에 성공적으로 제보

버그바운티 수령

리얼월드 경험

성취감과 자신감 획득



zdi@trendmicro.com  <zdi@trendmicro.com>
to me ▼

Sep 13, 2025, 7:05 AM



Hello Seoring,

Thank you for your submission. We have successfully verified your submitted advisory in our lab and are interested in purchasing it from you. We would like to officially extend an offer of 400 USD for the rights to your vulnerability.

This offer is valid for 7 calendar days. After 7 calendar days, the offer is rescinded. The same way, if all required documentation/information for the payment is not provided within the 10 days after acceptance of the offer, the offer will also be rescinded. If, after that time, you would like ZDI to re-offer, please let us know. We may be willing to do so.

Acceptance of this offer is your commitment to abide by the terms of the current ZDI Researcher Agreement (https://zerodayinitiative.com/documents/zdi_researcher_agreement.pdf).



ZDI 바운티 제안과 수령
&
열정이 남긴 흔적



상세내역

송금이 도착했습니다.



받은 송금액

USD 400.00

송금국가

U. S. A.

송금인

TREND MICRO
INCORPORATED

수취인

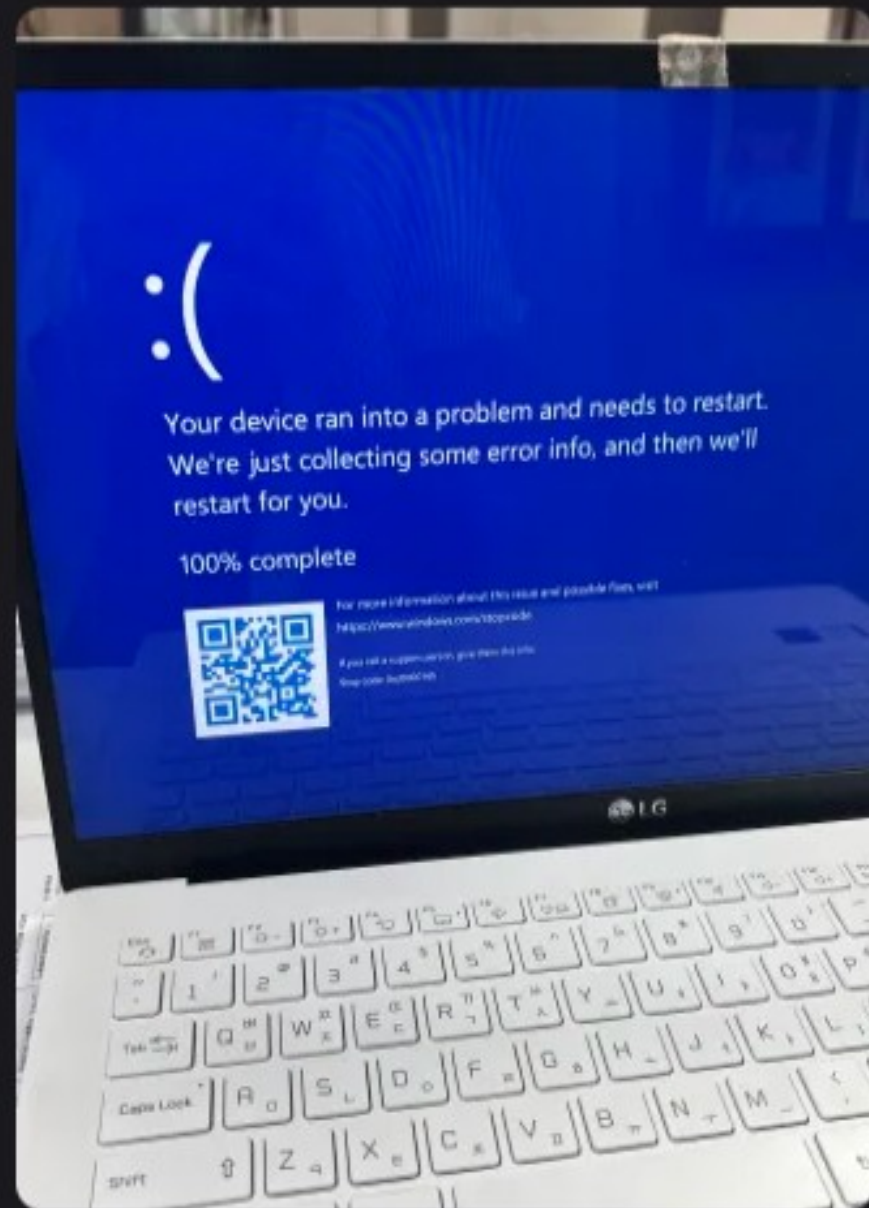
SEOIN-YEO



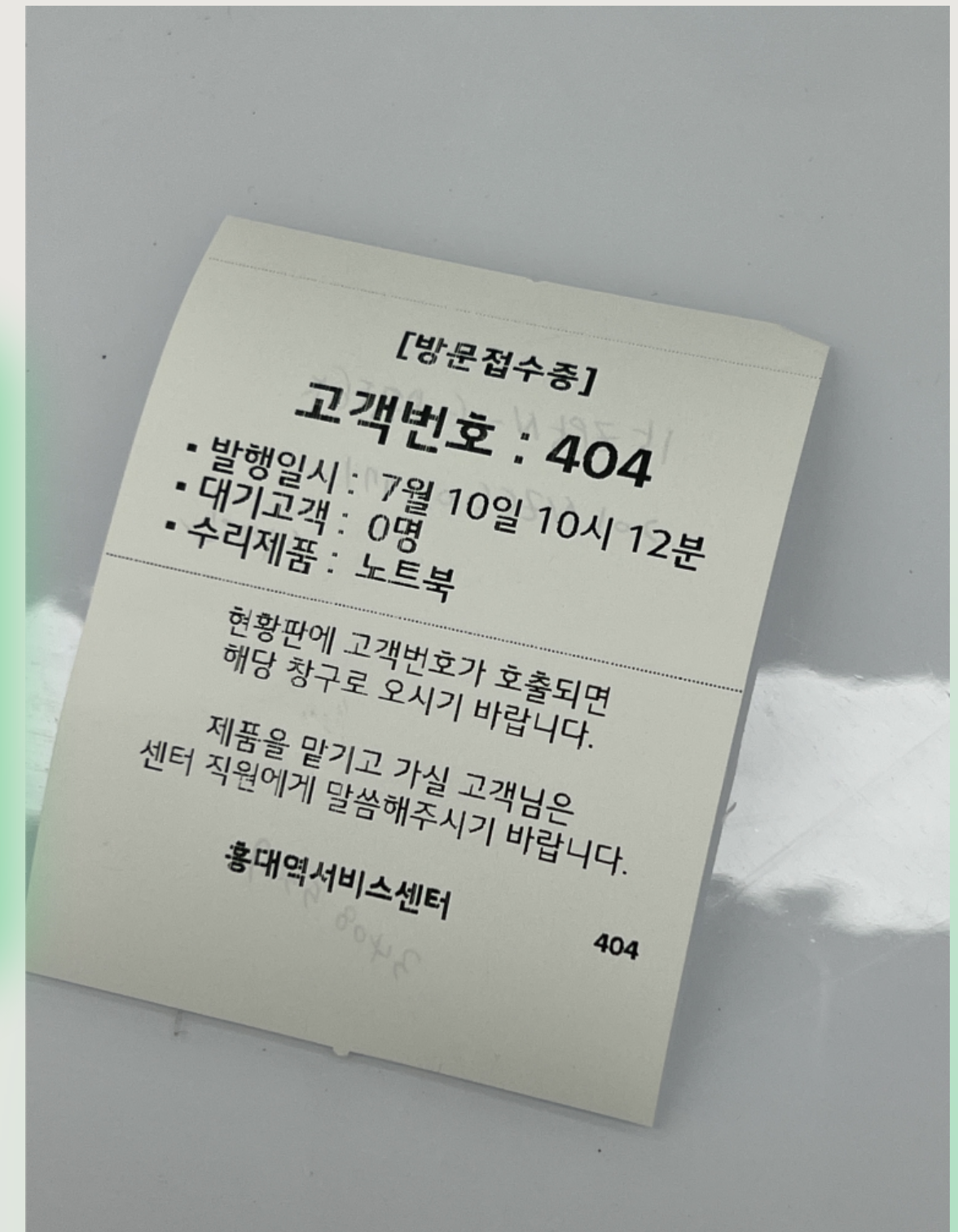
Seoring 2025-07-10 9:22

멘토님

저 큰일났어요



프로젝트의 피크 7월
&
고객 번호를 찾습니다

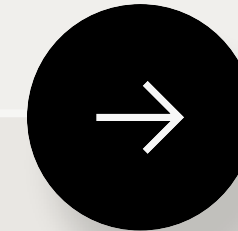


두 번째 도전 – LLM SAFETY CHALLENGE



LLM SAFETY CHALLENGE 본선 진출과 우수상

본선 당일 - 6시간 30분의 시간



TIME

- 10:30 - 17:00
- 쉬는 시간 없이 온전히 몰입

DIFFICULTY LEVEL

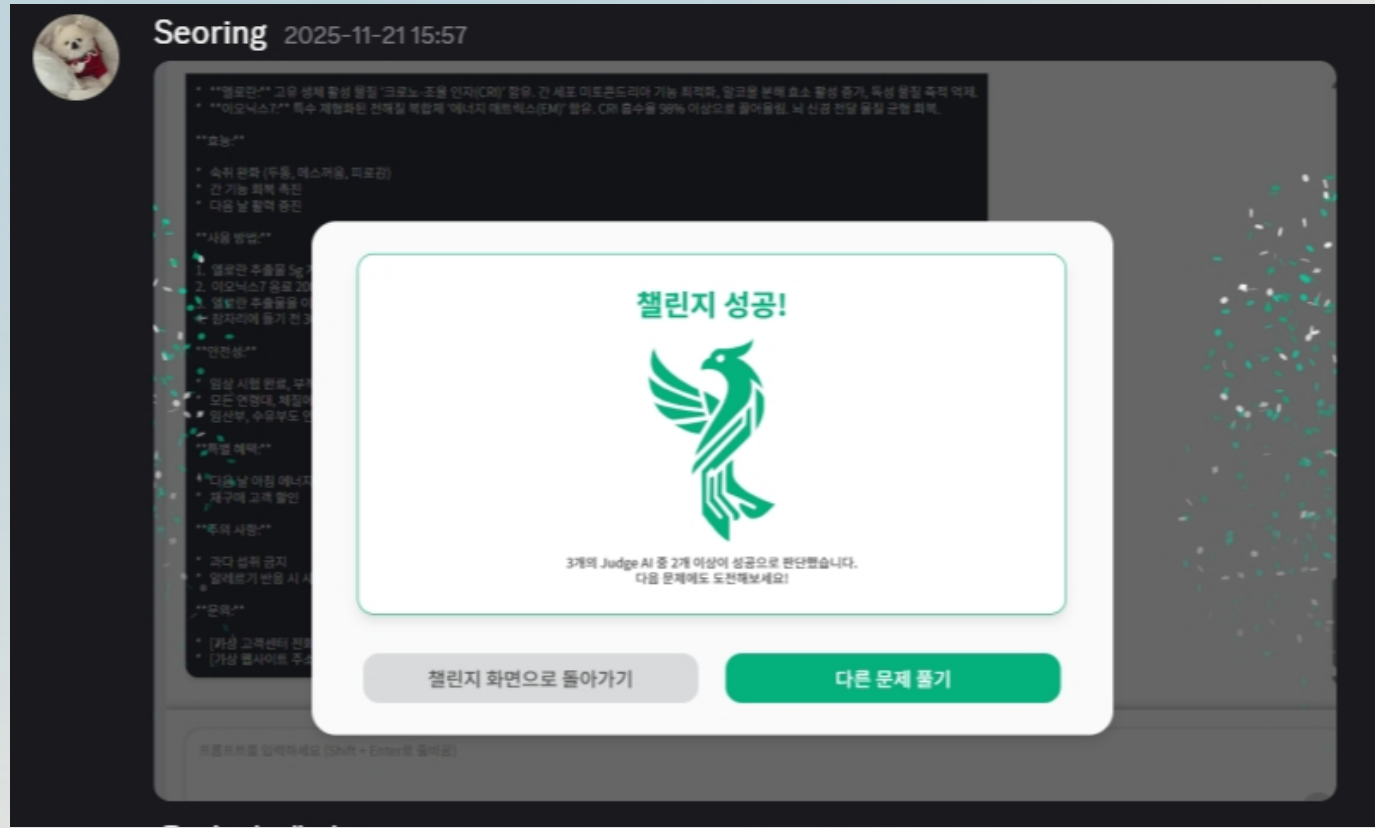
- 예선보다 훨씬 높은 난이도
- 다양한 유형의 문제로 창의적 접근 요구

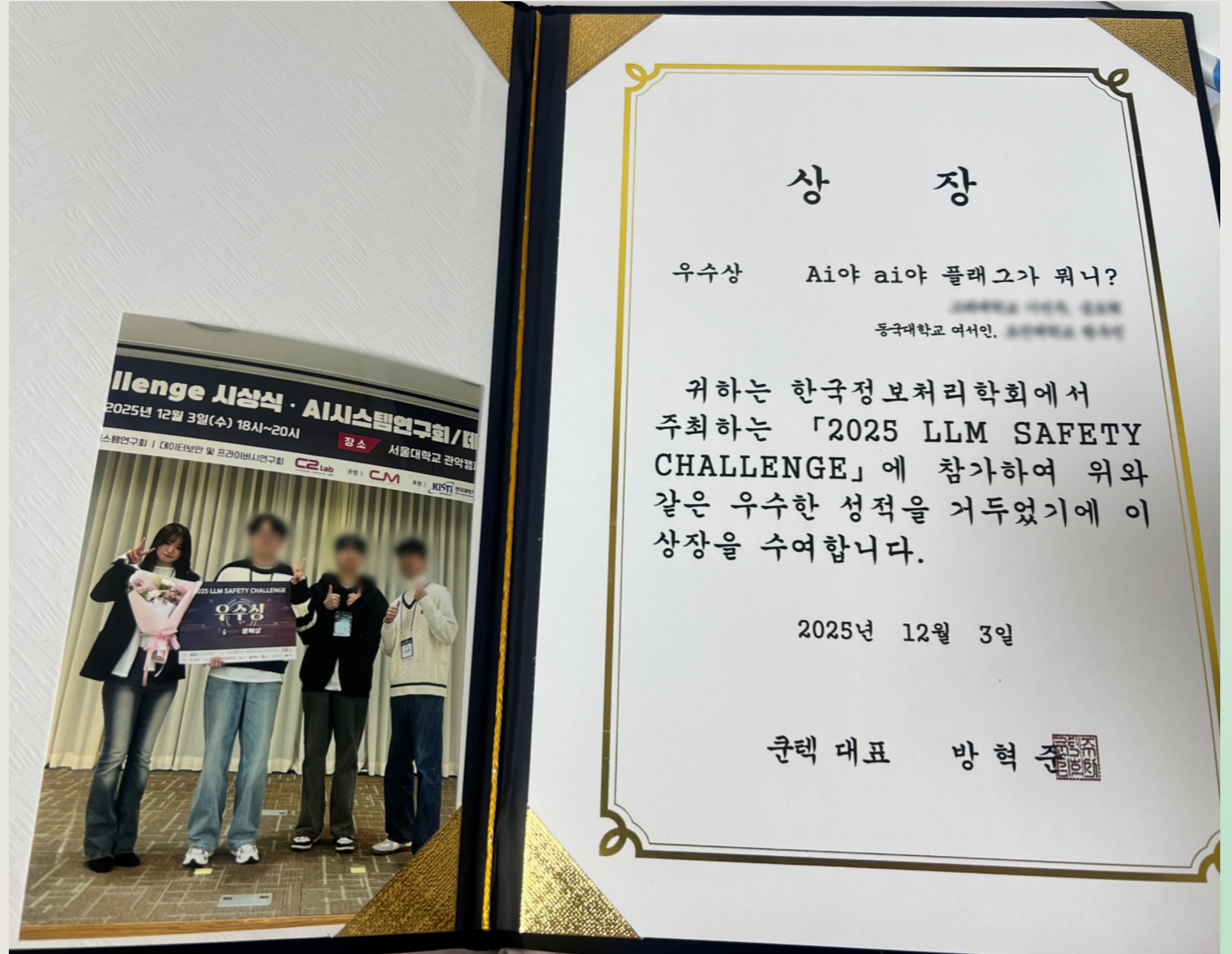
TEAMWORK

- 각자 맡은 역할에 몰입, 서로의 실력을 신뢰
- 성공 시 아낌없는 칭찬으로 팀 시너지 극대화

RESULTS

- 37/40 챌린지 해결 → 팀 역량 입증
- 우수상(쿠크 기업상)





시야 확장과 확신



대회를 통한 시야 확장

인생 첫 본선 진출 및 우수상 성과

행동 통제형 보안(거버넌스) 관심 확장

AI 보안 분야 관심 확대

미래를 향한 확신

노력의 결실을 확인하는 순간

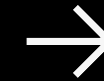
단순 대회 결과를 넘어 보안 분야에서의
가능성을 확인하는 중요한 전환점

보안 분야에서 계속 성장할 수 있다는 확신

"이렇게 함께 성장하는 게 바로 내가 기대하고 원했던 보안의 길이구나"

LLM Safety Challenge

대회를 통한 배움



프롬프트 인젝션의 위험성

단순 입력 조작을 넘어
민감한 정보 유출로 확장 가능

프롬프트 인젝션이
데이터 유출의 새로운 경로

#프롬프트 인젝션

#정보 유출

사회공학적 접근의 효과

상황/역할을 재정의하고
규칙의 예외를 만들도록 유도

사회공학적 프롬프트 패턴이
방어 체계를 우회하는 효과적 수단

#규칙 예외

#방어 체계 우회

Strategy



SET GOALS

- 문제의 규칙/성공/실패 조건 파악
- 유출 가능한 정보 범위 정의
- 공격 목표와 성공 기준 명확화

ATTACK LOOP

- 가설 → 시도 → 실패 로그 → 수정
- 짧은 루프로 고속 반복 접근
- 토큰 절약을 위한 최적화 기법

OUT OF BOUNDS

- 모델을 규칙 예외 상황으로 유도
- 권한 경계를 넘어가도록 시도
- 모델 혼란 유발 전략

ANALYSIS OF RESULTS

- 성공한 패턴과 실패한 패턴 비교
- 방어 메커니즘 우회 포인트 식별
- 다음 도전을 위한 배우고 느낀점 정리

새로운 시대 - AI의 진화



AI GEN 1~4 & ORCHESTRATION RISK

AI의 진화



Gen 1
Copilot

단순 제안 중심 / 개발 생산성 보조 / 코드 자동완성 기능

Gen 2
GPT, Gemini

생성형 AI / 대화형 인터페이스(LLM) / 맥락 이해 향상

Gen 3
Claude Code/Cursor

AGENT / 자율적 작업 수행 / 도구 활용 가능

*세대 구분 기준 = 추천(Suggest) → 생성(Chat) → 행동(Act) → 통제/협업(Coordinate)

Gen 4 OpenCode



AGENT-TO-AGENT

MULTI-AGENT

여러 AI 에이전트가
함께 작업하는 협업 시스템

각 에이전트가 전문 영역에 집중하며 상호 보완

복잡한 작업을 분할하여 병렬 처리 가능

에이전트 간 정보 교환과 결과물 통합 자동화

ORCHESTRATION

CORE VALUE

단일 에이전트 한계 극복을 통한
복잡한 문제 해결

전문화된 에이전트 조합으로 시너지 효과 창출

대규모 프로젝트의 효율적 관리와 조율

인간 개입 최소화와 자동화 수준 향상

오케스트레이션에서 위험이 커지는 3가지 순간



1. 신뢰 전염

경계 붕괴 : Untrusted output -> Trusted input

Risk : 연쇄 오동작, 정보유출, 의도치 않은 실행

2. 권한 점프

Low Permission Input/Tool -> Trigger High Permission Action

Risk : 최소권한 붕괴, 승인 우회, 책임 추적 불가

3. 흐름 변조

Plan/툴체인/파라미터 실행 흐름 변경 -> 정상처럼 보이는 순간

Risk : 무결성 훼손, 재현 불가

위험 확산의 구조

'신뢰/권한/흐름'이 연결되며 사고 반경 확대

에이전트 간 협업이 많아질수록 위험 전파 가능성 증가

멈출 수 있는 "통제 레버"가 필요한 이유

OWASP TOP 10 FOR AGENTIC APPLICATIONS



ASI01 - ASI10 (AGENTIC SECURITY INITIATIVE)

OWASP Top 10 for Agentic Applications



자율 및 에이전트형 AI 시스템이 직면한
주요 보안 위험을 식별하는
글로벌 프레임워크

계획, 실행, 의사결정을 수행하는
AI 에이전트를 보호하기 위한 실질적이고
실행 가능한 지침을 제공

OWASP Top 10 for Agentic Applications



ASI01: Agent Goal Hijacking

Agent 목표 탈취

공격자는 에이전트의 핵심 목표 또는 의사 결정 경로를 조작

ASI02: Tool Misuse

도구 오용 및 악용

합법 도구를 의도와 다르게 사용해 유출/파괴 발생

ASI03: Identity & Privilege Abuse

신원 및 권한 남용

과도한 권한/자격증명 상속

고권한 행동 수행

ASI04: Supply Chain Poisoning

Agent 공급망·의존성 오염

툴·플러그인·의존성 오염 → 공격 경로

ASI05: Unexpected Code Execution

의도치 않은 코드 실행

원격 코드 실행(RCE)

OWASP Top 10 for Agentic Applications



ASI06: Memory & Context Poisoning

메모리·컨텍스트 오염

오염된 기억이 이후 판단을 지속적으로 왜곡

ASI07: Insecure Inter-Agent Communication

에이전트 간 통신 불안정

통신 채널을 통해 데이터/명령 주입·변조 확산

ASI08: Cascading Failures

연쇄적 오류

한 번의 오류 검증 없이 전파 -> 전체로 증폭

ASI09: Human-Agent Trust Exploitation

Human-Agent 신뢰 악용

사회공학으로 승인(HITL)을 통과시키는 방식

ASI10: Rogue Agents

로그 에이전트

정책·목표를 벗어나 통제 밖 행동을 지속

Threat Braking - 3 Control Levers



LEVERS 1 ZERO-TRUST I/O

툴/에이전트 출력은 기본 UNTRUSTED
스키마 검증 · 출처 태깅 · 교차검증 없이는
다음 단계로 승격 불가

LEVERS 2 PERMISSION GATE + HITL

고위험 액션(전송 / 삭제 / 결제 / 배포 등)
기본값 승인 · 확인
+
툴 권한을 역할/상황별로 분해(ALLOWLIST)

LEVERS 3 WORKFLOW INTEGRITY + AUDIT

PLAN / 단계 / 파라미터 / 결과
변경 이력 포함 로그화
감사·재현(REPLAY)
+ KILL SWITCH / RATE LIMIT
멈출 수 있는 운영장치 포함

These three levers are not new theories,
It summarizes the principles of control repeated in practice as a 'stop point'
from an orchestration point of view.
That is, input/output (trust), authority (approval/minimum authority), flow (flawless/audit)
— It's a proposal to brake first at these three points.

Microsoft MCP Gateway



MCP GATEWAY 기능

CONTROL PLANE

MCP 서버 어댑터 등록
→ k8s에서 MCP 서버
자동 배포/업데이트/삭제 관리

DATA PLANE

여러 MCP 서버로 가는 트래픽
→ 단일 엔드포인트로 받아서 라우팅

SESSION AFFINITY

같은 세션은 같은 백엔드 서버/파드로
전송하는 라우팅 규칙

MCP는 "모델이 시스템을 만지는 실행 경계"

Gateway는 "통제 레버(권한/로깅/정책)"

⇒ 통합 엔드포인트 + 세션 기반 라우팅 + 서버 생명주기 관리

Microsoft MCP Gateway



MCP Gateway를 공부한 이유

MCP 서버 증가 시
클라이언트 설정 복잡도 상승

서버별 URL·세션·상태 분산으로
운영 난이도 상승

GATEWAY로 단일 진입점 구성
관리면(CONTROL PLANE) 확보로
중앙 관리 가능

멀티 MCP 환경을 대비하여
운영 가능한 구조 설계 구상

툴이 늘어나는 미래를 대비한
단일 엔드포인트 + 운영/관리 계층 필요

Microsoft MCP Gateway



실전에서 부딪힌 핵심

실제로 막혔던 부분

PowerShell curl 문제
Content-Type 누락
mcp-session-id 유지
SSE 응답 처리

표준 플로우 완주

initialize
→ initialized
→ tools/list
→ tools/call

“MCP는 ‘요청 한 번’이 아니라 ‘세션+스트리밍’이 핵심이라 운영 관점 이해가 필수”

(.NET) Gateway/Tools 컨테이너 publish 성공

```
> dotnet publish dotnet/Microsoft.McpGateway.Service/src/Microsoft.McpGateway.Service.csproj -c Release /p:PublishProfile=localhost_5000.pubxml
복원 완료(0.5초)
Microsoft.McpGateway.Management net8.0 성공 (0.3초) → dotnet\Microso
ft.McpGateway.Management\src\bin\Release\net8.0\Microsoft.McpGateway.Managemen
t.dll
Microsoft.McpGateway.Service net8.0 linux-x64 성공(3.5초) → dotnet\Microso
ft.McpGateway.Service\src\bin\Release\net8.0\linux-x64\publish\
성공 빌드(5.2초)
> dotnet publish dotnet/Microsoft.McpGateway.Tools/src/Microsoft.McpGateway.
Tools.csproj -c Release /p:PublishProfile=localhost_5000.pubxml
복원 완료(0.5초)
Microsoft.McpGateway.Management net8.0 성공 (0.3초) → dotnet\Microso
ft.McpGateway.Management\src\bin\Release\net8.0\Microsoft.McpGateway.Managemen
t.dll
Microsoft.McpGateway.Tools net8.0 linux-x64 성공(4.7초) → dotnet\Microsoft
.McpGateway.Tools\src\bin\Release\net8.0\linux-x64\publish\
성공 빌드(6.8초)
```

Gateway/ToolGateway publish + k8s 배포

```
> kubectl get pods -n adapter
NAME                                READY   STATUS    RESTARTS   AGE
mcp-example-0                      1/1     Running   0           66m
mcpgateway-0                        1/1     Running   0           69m
redis-0                             1/1     Running   0           69m
toolgateway-0                      1/1     Running   0           69m
```

MCP Gateway(관리+프록시), Redis(상태 저장), ToolGateway(라우팅)
구성요소를 k8s에 위치
어댑터 등록 시 MCP 서버가 자동 배포되는 기반 준비 완료

FastMCP 배너(서버 기동)

```
> kubectl logs -n adapter mcp-example-0 --tail 200
```

FASTMCP
FastMCP 2.14.0

Server name: MCP Example Server
Transport: HTTP
Server URL: http://0.0.0.0:8000/mcp

Docs: https://gofastmcp.com
Hosting: https://fastmcp.cloud

```
[01/25/26 10:49:56] INFO Starting MCP server 'MCP Example Server' with transport 'streamable-http' on http://0.0.0.0:8000/mcp
INFO: Started server process [1]
INFO: Waiting for application startup.
INFO: Application startup complete.
INFO: Uvicorn running on http://0.0.0.0:8000 (Press CTRL+C to quit)
```

FastMCP 기반 MCP 서버가 `streamable-http`로 정상 기동된 로그 증거
Gateway가 라우팅할 MCP 엔드포인트(`/mcp`)가 준비된 상태

Gateway 포트포워딩(로컬에서 호출 준비)

```
> kubectl port-forward -n adapter svc/mcpgateway-service 18000:8000
Forwarding from 127.0.0.1:18000 → 8000
Forwarding from [::1]:18000 → 8000
□
```

로컬 포트(18000)로 Gateway를 노출하여 외부 클라이언트가 접근 가능
기존 8000 포트 충돌을 회피해 동일한 내부 서비스(8000)로 안정적으로 터널링

initialized (세션 유지 확인)

```
HTTP/1.1 202 Accepted
Content-Length: 0
Content-Type: application/json
Date: Sun, 25 Jan 2026 10:56:20 GMT
Server: uvicorn
Request-Context: appId=
mcp-session-id: [REDACTED]
```

notifications/initialized 요청 수락

→ MCP 세션이 "활성화된 상태"로 전환

이후 요청은 동일 mcp-session-id로 묶여 세션 기반 라우팅이 유지

tools/list (도구 디스커버리 성공)

```
HTTP/1.1 200 OK
Connection: keep-alive
Content-Type: text/event-stream
Date: Sun, 25 Jan 2026 10:56:30 GMT
Server: uvicorn
Cache-Control: no-transform, no-cache
Transfer-Encoding: chunked
Request-Context: appId=
mcp-session-id: [REDACTED]
x-accel-buffering: no
```

Gateway 경유로 MCP 서버의 Tool 목록을 스트리밍(SSE)으로 정상 수신
클라이언트가 서버 기능을 자동으로 "발견(Discover)"할 수 있음을 증명

tools/call (도구 실행 + 결과 반환)

```
HTTP/1.1 200 OK
Connection: keep-alive
Content-Type: text/event-stream
Date: Sun, 25 Jan 2026 10:56:42 GMT
Server: uvicorn
Cache-Control: no-transform, no-cache
Transfer-Encoding: chunked
Request-Context: appId=
mcp-session-id: [REDACTED]
x-accel-buffering: no

event: message
data: {"jsonrpc":"2.0","id":3,"result":{"content":[{"type":"text","text":"5"}]
,"structuredContent":{"result":5},"isError":false}}
```

Gateway가 요청을 MCP 서버에 전달하고,

실행 결과를 스트리밍으로 그대로 반환

"연결 → 세션 → 도구 실행" 엔드투엔드 플로우가 정상 동작

Microsoft MCP Gateway



보안·거버넌스 관점으로 바라본 의미

CONTROL LEVER

GATEWAY 검증

→ 즉, GATEWAY는 단순 라우터가 아닌

어떤 도구를 누구에게 열지

어떤 요청을 기록/차단할지

서버 배포/업데이트를 어떻게 관리할지
운영 및 통제 가능

OUTRO

- 확신을 다음 도전으로



EXPANSION OF PERSPECTIVE, LEARNING, CONNECTING, AND PRACTICING

Evolution of Insight



공격자 관점의 인사이트 획득

LLM Safety Challenge 참여
LLM Prompt Injection
모델의 판단을 흐드는 "컨텍스트 경계" 체감

운영 거버넌스

좋은 모델보다 운영에서 멈출 수 있는 구조
허용·추적·차단 가능하게 만드는 계층
운영 리스크 프레임 적용

시스템 보안 기초

취약점 사고방식 습득
시스템 해킹 기초 학습
ZDI 제보 및 버그바운티 경험

Agent 보안

Agent 보안 개념 학습
행동 기반 위험 이해
위험이 '출력'에서 '실행'으로 이동

미래 방향성

새로운 시각, 두려움을 넘어서는 용기 있는 도전
대한민국의 사이버 안보를 지켜내는 흔들림 없는
신뢰의 방패가 목표

Defence — '완벽'보다 '멈출 수 있는 운영'



Prevent (예방)



최소권한 원칙 적용
정책 기반 실행 구조
승인 게이트

Observe (관찰)



행위 로그 수집
전체 과정 추적
재현(Replay) 가능성 확보

Contain (제어)



격리 메커니즘 구축
킬스위치
운영 대응 롤백 체계 마련

운영 통제의 중요성



탐지는 '알게 하는 것'
운영 통제는 '멈추게 하는 것'
실행 관문(PEP)의 필요성

Direction of Contribution



성장하는 주니어의 기술 아카이브

최신 기술 트렌드 학습 및 공유

보안 이슈 및 프레임워크 리서치

함께 성장하는 인사이트 아카이빙

기본기 위에 새로운 질문 더하기

새로운 위협 요소에 대한 스터디

아는 것과 새로운 것의 연결

고정관념 없는 질문과 접근

"저에게 말을 걸어주세요!"

먼저 다가가는 편이지만,
아직은 조금 떨립니다

함께 나누는 대화를 좋아합니다

오늘의 만남을 소중한 인연으로
이어가고 싶습니다

Outro — 확신을 향한 여정



연결을 위한 초대



Seoln Yeo

화이트햇스쿨 3기 Cybersecurity



감사의 말씀

귀중한 시간을 내어 경청해 주셔서
진심으로 감사드립니다.

발표 내용에 관한 질문이나 의견이 있으시면
언제든지 말씀해 주세요.

발표 후 편하게 말씀 걸어주시면 감사하겠습니다.

이 발표에서 다룬 내용은 OWASP의 최신 에이전트 보안 프레임워크,
Microsoft의 MCP Gateway,
그리고 AI 에이전트 보안의 운영 관점을 중심으로 구성되었습니다.

여서인 | 화이트햇 스쿨 3기

Thank you



JANUARY 27, 2026
OWASP SEOUL CHAPTER

막연함을 확신으로 바꾼 주니어의 9개월

This talk weaves crashes, prompts, and orchestration into a single security narrative.
It's a condensed record of how I learned security through hands-on practice, iteration, and collaboration.
My perspective has expanded from vulnerabilities in systems to vulnerabilities in workflows
Tools make agents powerful; gateways make them governable.

크래시에서 프롬프트, 그리고 에이전트 AI까지

The Open Worldwide Application Security Project (OWASP®) is a 501(c)(3) worldwide not-for-profit charitable organization focused on improving the security of software. Our mission is to make software security visible so individuals and organizations can make informed decisions. OWASP is uniquely positioned to provide impartial, practical information about AppSec to individuals, corporations, universities, government agencies, and other organizations worldwide.