PORTFOLIO

취약점 분석, 리버스 엔지니어링, 익스플로잇 개발을 중심으로 한 보안 연구자 김태중의 포트폴리오입니다.

Github

github.com/Jamongseed

Phone

010-4058-1219

Email

ronan1_@naver.com

INDEX

목차

INTRODUCTION

안녕하세요, 김태중입니다.

발생하는 위협을 ‘탐지 가능한 흔적’과 ‘검증 가능한 근거’로 바꾸는 데 집중해 온
신입 보안 엔지니어입니다.

브라우저 내부 행위 기반 위협 탐지·증거 수집·대시보드 관제 흐름을 구현했고,
모바일 악성코드 동적 분석 자동화와
다크웹 기반 위협 인텔리전스 수집·분석으로 우선순위 선정과 분석 효율을 높였습니다.
리버스 엔지니어링·PoC·자동화 도구 제작등의 경험으로 검증 가능한 근거를 만들고자 합니다.

한문 이름 '金兌重'의 의미처럼,
자산(金)을 위협하는 신호를 탐지 가능한 증거로 변환(兌)하고,
중요도(重)에 따라 우선순위를 판단하는 보안 엔지니어를 지향합니다.

EDUCATION

교육

2023.08 세종대학교 졸업

정보보호학과 전공, 2017.02 - 2023.08

주요 전공

- 시스템 보안
  - 시스템해킹과보안, 운영체제, 시스템프로그래밍
- 리버스 엔지니어링·악성코드
  - 악성코드분석, 보안프로그래밍, 알고리즘및실습
- 네트워크·기초 보안
  - 컴퓨터네트워크, 정보보호와보안의기초
- 암호·이론
  - 대칭키 암호론, 정수론

학부 주요 활동 요약

- 2017 – 정보보호 기초
- 2020 – 시스템·네트워크 보안 집중
- 2021 – 악성코드 분석 / 리버스엔지니어링
- 2022 – SW 보안 프로젝트 수행
- 2023 – 보안 자동화 도구 개발 / 졸업연구

구름×KT Cloud DEEP DIVE

정보보호 전문가 과정, 2025.07~2026.02

- 실무 중심 7개월 집중 보안 교육
- 팀 단위 협업 기반의 취약점 분석/리버싱/보안 자동화 프로젝트 수행
- 전체 최우수 수료생, 전체 최우수팀(PL) 선정

WORK EXPERIENCE

경력

육군 정보보호병 조교 (정보통신학교)

병장 만기전역, 2018-2020

- 군 내부 네트워크 및 시스템 보안정책(TACS) 운영·모니터링
- 망 분리 환경 내 계정권한 관리 및 접근통제 지원
- 보안 사고 예방을 위한 로그/트래픽 분석 및 보고
- 군 장교 교육용 해킹 실습 시스템 도입 테스트 및 운영 지원(2019~2020)

Security Factorial 4기 (세종대학교 Student Cybersecurity Lab)

부원, 2018-2021

- 악성코드 분석, 리버싱, 시스템 보안 관련 스터디 및 실습
- 웹·시스템 기반 CTF 문제 풀이 및 내부 해킹 실습
- 내부 기술 세미나 발표 및 보안 연구 공유 활동

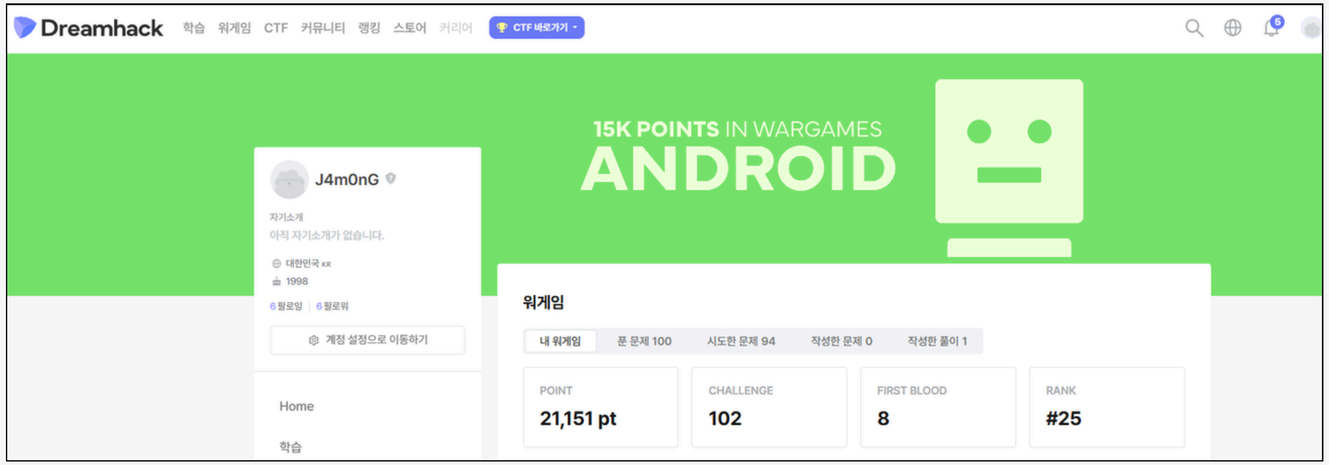
세종대학교 파이썬 코딩 조교

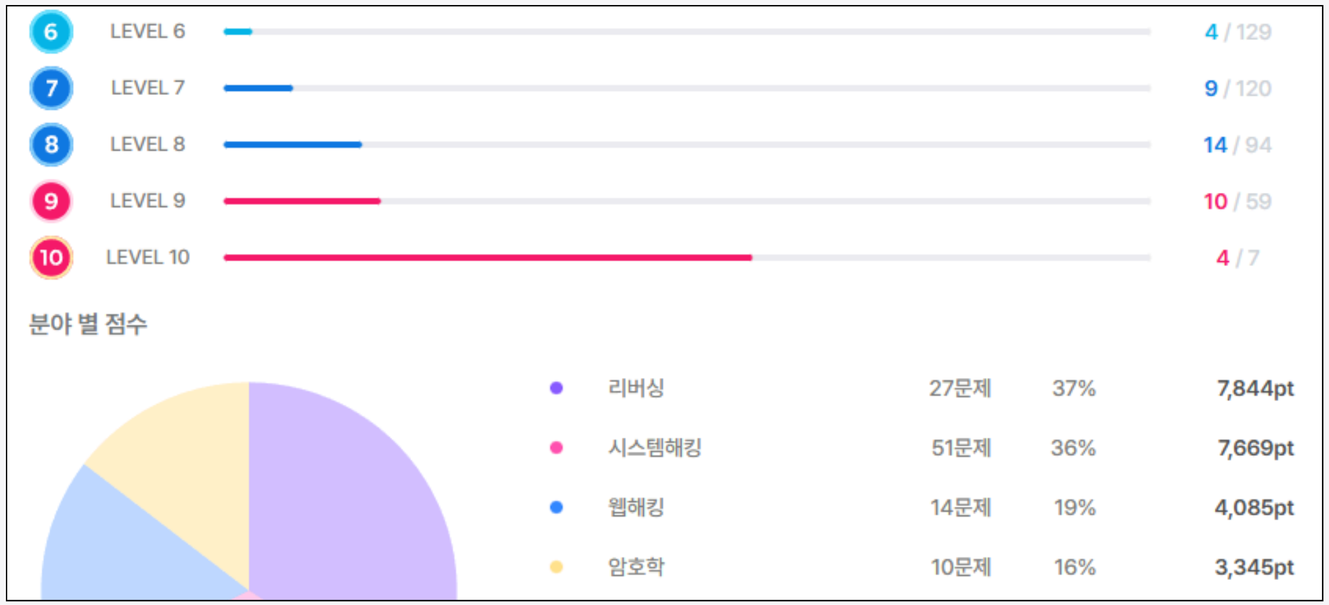
학부생, 2018-2022(4학기)

- 세종대학교 고급프로그래밍입문-Python 조교 담당
- 학기 중 매주 약 4~6시간의 코딩 수업 진행 및 코딩 과제 작성
- 각기 다른 4개 학과를 대상으로 코딩 교육 진행

ACHIEVEMENTS

Security Research & CTF





성과 및 영향

- DreamHack 전체 워게임 랭킹 25위 달성
- 고난도(Pwnable/Web/Rev/Crypto) 문제 50+ 개 해결
- CVE 기반 문제 직접 exploit 성공
- Exploit 제작 100+개 / 웹·리버싱·Pwn 전 분야 해결 경험

CTF WriteUp

- Notion: https://ahead-krypton-19d.notion.site/DreamHack-CTF-Exploit-Research-Archive-2c044e52b3b380378493cf1f728bce6c
- Exploit 및 취약점 분석 관련 글을 모아둔 아카이브 노션

TECHNICAL SKILL

기술 스택

Programming Languages

- Python
- C
- JavaScript

Backend/Infra

- Node.js(Lambda), API Gateway
- MySQL, DynamoDB
- AWS S3

DevOps

- Docker
- Git
- Vscode

OS

- Windows
- Linux (Ubuntu, Kali, CentOS)
- Android

Security Tools

- Burp Suite
- OWASP ZAP
- Wireshark

Reverse Engineering

- GDB(pwndbg)
- IDA
- Ghidra
- pwntools

PERSONAL SKILLS

핵심 역량

끝까지 파고드는 문제 해결 능력

새로운 보안 이슈·취약점을
직접 재현하고 분석하여,
근본 원인을 찾아 해결하는
실전 문제 해결 역량을 가지고 있습니다.

주도적 실행과 완성

해야 할 일을 스스로 나누고 막히는 구간은
빠르게 실험해 결정을 내립니다.
결과물은 동작뿐 아니라
운영 가능한 형태(문서/절차/재현 방법)까지
포함해 완성합니다.

협업 기반의 프로젝트 활동

협업 기반의 보안 프로젝트를 수행하며
역할을 분담하고
팀 단위로 문제를 해결한 경험이 있습니다.
보안 도구 개발·데이터 분석·CTF 풀이 등
다양한 분야에서
결과물을 만들어낸 협업 역량을 가졌습니다.

PROJECT LINKS

Browser Runtime Security System / 브라우저 런타임 위협 행위 탐지 시스템

- GitHub: https://github.com/Jamongseed/browser-runtime-security
- Notion: https://ahead-krypton-19d.notion.site/Browser_Runtime_Security-2e344e52b3b3809bbf6df283dde335c4

Vulnerability Fingerprinting & CVE Mapping System / 포트 기반 취약점 분석 및 CVE 취약점 자동 매핑 시스템

- GitHub: https://github.com/Jamongseed/SemiProject-PortScanASM
- Notion: https://ahead-krypton-19d.notion.site/2b444e52b3b3802dbd3ac44df9f1dad0

DarkWeb Threat Intelligence Automation System / 다크웹 기반 위협 인텔리전스 자동 수집·분석 시스템

- GitHub: https://github.com/Jamongseed/SemiProject-DarkwebOSINT
- Notion: https://ahead-krypton-19d.notion.site/DarkWebOSINT-Project-2c044e52b3b3807ab68ee41ce3a6266f

Mobile Malware Dynamic Analysis / 모바일 악성코드 동적 분석 자동화 시스템

- GitHub: https://github.com/Jamongseed/SemiProject-MobSFAnalysis
- Notion: https://ahead-krypton-19d.notion.site/MobSF-2a644e52b3b380419cb5d20d6b0b6e41

TEAM PROJECT 1

Browser Runtime Security System

소개

브라우저 안에서 링크/폼/네트워크 요청등의 실행 직전에 바뀌는
런타임 변조 공격을 탐지하고
단순 경고가 아니라 왜 위험한지 근거, 증거까지 남기는
크롬 브라우저 확장프로그램을 이용한 보안 플랫폼

기간 / 인원 / 수상

- 25.12.09~25.02.02
- FE 1인, 지원자 포함 보안 연구자 5인 팀 프로젝트
- 구름×KT Cloud DEEP DIVE 최종 프로젝트 최우수팀 선정

핵심 기능

- 확장프로그램 설치/ON, 허용·예외 도메인 설정 등 정책 설정
- 크롬에서 DOM·링크·폼·요청 변화를 감지해 이벤트로 기록
- 변조 전·후 값과 initiator/stack 등 Payload Evidence 저장
- 룰 기반 위험도 판단, WebCrack을 활용한 난독화 해제, AI 악성 판별(보강)
- 저장된 기록을 타임라인으로 조회, 필터로 분석하기 위한 대시보드 제공

팀 성과

- DOM/네트워크/프로토타입 변조를 감지하고 이벤트·덤프 수집 구축
- 이벤트 저장(DynamoDB), 스크립트 덤프 저장(S3), 집계/조회 API 구축
- 세션 타임라인, 이벤트 상세, 통계/트렌드 시각화 대시보드 구축
- 서버는 정상인데 브라우저에서만 진실이 바뀌는 공격 시나리오 재현 제작

지원자 개인 기여

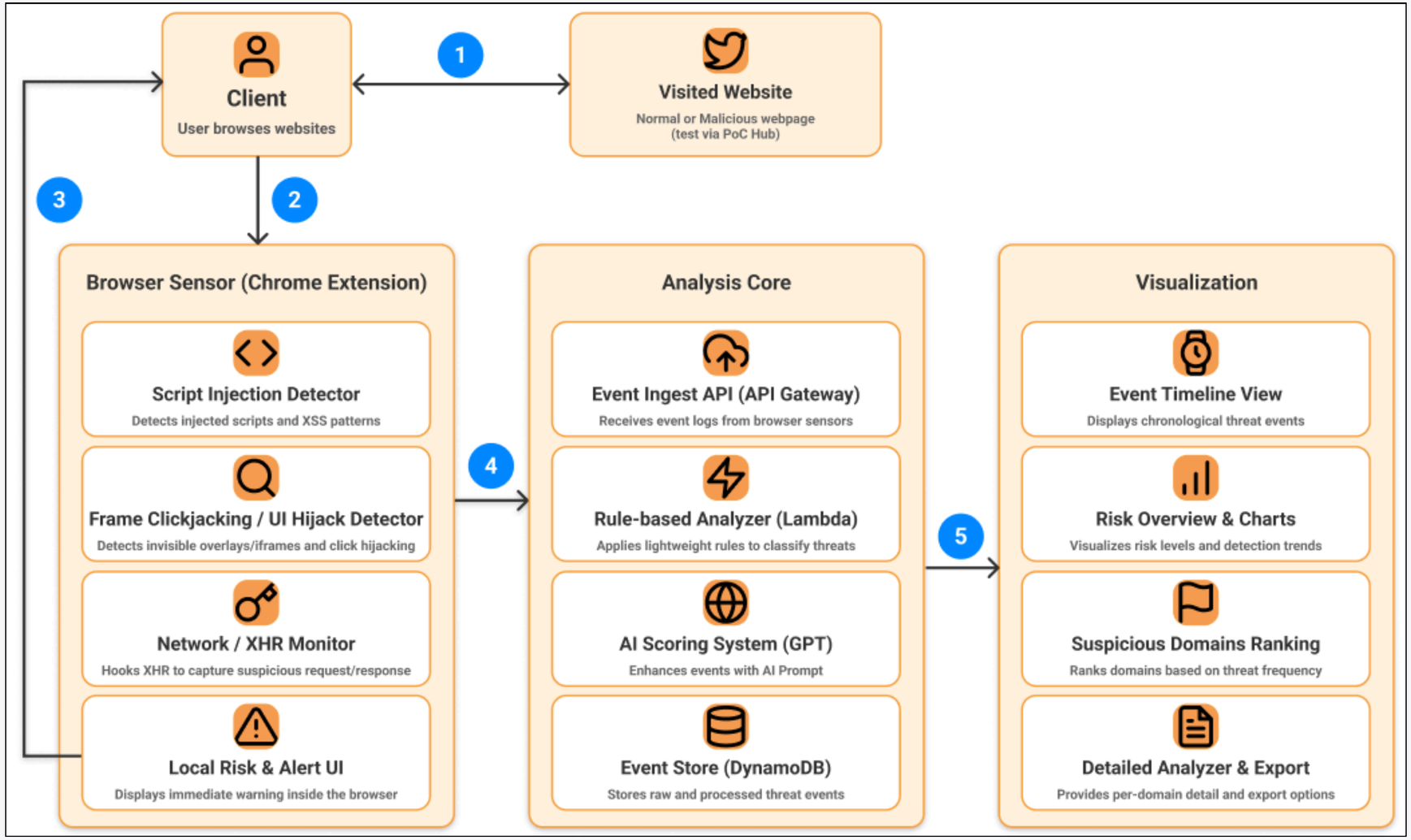
역할 요약 : 팀 리더, PoC 구축, 룰 설계, 공격 분석, 허브 구축

세부 내용

- 탐지 로직 및 룰 설계
  - 55개(기본 동적 탐지 룰) + 44개(스크립트 주입 공격 탐지 룰) 구축
- PoC 제작 및 시나리오 설계
  - PoC 10개 중 4개(A,F,G,H) 구현
- 브라우저 기반 공격 분석
  - DOM/네트워크/프로토타입 변조 관련 9개 탐지 js 코드 구현
- 악성 PoC 허브 구축
  - 누구나 PoC 테스트를 해볼 수 있는 PoC 테스트 전용 도메인 구축

TEAM PROJECT 1

Browser Runtime Security System



TEAM PROJECT 1

Browser Runtime Security System

핵심 구현 및 기술 스택

- Chrome Extension 기반 Runtime Sensor(Manifest V3)
  - content script로 DOM/링크/폼 변조 및 사용자 입력 흐름 관측
  - background/service worker에서 이벤트 수집·버퍼링·전송 제어
- Evidence(근거) 수집 파이프라인
  - before/after 스냅샷, initiator(호출 위치), stack/타이밍, origin/target 등 재현 가능한 증거 필드 설계
  - payload 해시(SHA-256) 기반 식별 및 중복 제거(dedup)
- Serverless 분석·저장 백엔드
  - API Gateway + AWS Lambda 기반 수집 엔드포인트
  - S3(JS dump 저장) + DynamoDB(이벤트/세션/상태 저장)로 분리 설계
- 룰셋 기반 스코어링 + AI 보강(선택)
  - 이벤트 타입별 scoreDelta/severity 룰셋 적용 → 최종 위험도 산정
  - 애매 케이스는 AI reason/verdict로 보강(설명 문장 톤 가이드 포함)

주요 기능 흐름

1. 확장 프로그램 설치 및 정책 설정
2. 행위 기반 런타임 변조 공격 탐지
3. 변조 후 전후 값과, 공격 payload 등 저장 및 사용자 경고 제공
4. 룰 기반 위험도 판단 및 필요 시 AI 악성 판별(보강)
5. 최종 리포트 자동 생성 및 대시보드를 통한 제공

기능 상세 설명

1. Runtime 변조 탐지
   - DOM 변경, 링크 href 변조, 폼 action/parameter 변조, 요청 목적지/헤더 변조 등 이벤트화
2. Evidence(근거) 수집
   - 변경 전/후 값, 변조를 유발한 스크립트 위치(initiator), 호출 스택/타이밍 기록
   - 필요 시 공격 payload/스크립트 dump 저장
3. 중복 제거·전송 안정화
   - 동일 payload 반복 발생 시 해시 기반 dedup으로 이벤트 폭주 방지
   - 로컬 버퍼링/재전송으로 네트워크 불안정 상황 대응
4. 룰 기반 판단 및 시각화
   - 룰셋 기반 점수/심각도 계산 및 세션 단위 타임라인으로 공격 흐름 제공
   - (선택) AI 보강으로 위험 요약/판단 근거 문장 생성

핵심 성과

- PoC 시나리오 10개로 런타임 변조 공격을 재현·검증
- 탐지 이벤트 타입 24종을 브라우저 런타임에서 수집·분류 및 서버/WAF가 놓치기 쉬운 클라이언트 실행 직전 변조 영역을 관측 가능하게 구현
- 정책 룰셋(default-v1) 룰 52개 + 스코어링 모델(scoring-model-v1) 활성 룰 44개(signals 36 + combos 8)로 위험도 점수화/심각도 분류를 자동화
- 이벤트당 공통 evidence/컨텍스트 필드 12개 표준화 및 분석 체계 구축
- 해시(SHA-256) 기반 dedup/캐싱/전송 안정화 파이프라인을 포함해 운영 구축

TEAM PROJECT 1

Browser Runtime Security System

트러블 슈팅 과정

1) PoC 허브 배포 시, 동일 도메인 때문에 공격 재현이 깨지는 문제

PoC를 누구나 접속해서 바로 재현시키려고 허브 도메인에 올렸는데, 로컬에서는 잘 되던 cross-site(공격자/피해자 분리) 시나리오가 배포 환경에서 깨졌다. 원인은 단일 도메인에서 경로만 나누면 브라우저 기준으로 Origin이 동일해져 iframe/clickjacking/3rd-party script 같은 케이스가 로컬과 다르게 동작하기 때문이었다.
이를 해결하기 위해 해결 PoC 허브를 페이지 모음이 아니라 오리진 모음으로 설계하는 쪽으로 바꿨다. victim/attacker/thirdparty를 서브도메인으로 분리하고, 서버의 CORS 허용 Origin도 고정값이 아니라 환경별로 안전하게 주입하도록 구성해 재현 일관성을 확보했다.

2) PoC-G(Service Worker Persistence) 배포 환경에서 SW가 등록되지 않는 문제

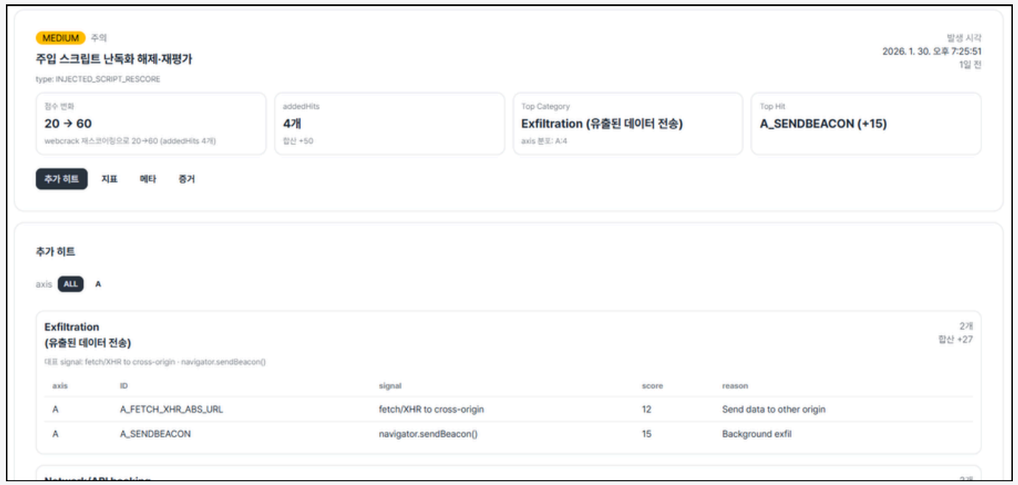
Service Worker 기반 변조 PoC는 로컬(localhost)에서는 정상 재현되는데, 배포한 허브에서는 SW가 아예 등록되지 않거나 기대한 경로에서 동작하지 않는 문제가 있었다. 원인은 Service Worker가 기본적으로 보안 컨텍스트(HTTPS) 를 요구하고, 배포 환경에서 scope/캐시 정책에 따라 기존 SW가 남아 재현 결과가 흔들릴 수 있다는 점이었다.
이를 해결하기 위해 PoC 허브는 HTTPS를 기본으로 하고, SW 스크립트의 경로·scope를 명확히 고정했다.
또한 PoC UI에서 등록 상태를 바로 확인하고 초기화할 수 있도록(등록 목록 조회/언레지스터) 리셋 루틴을 넣어, 테스트 반복 시 결과가 누적되지 않게 했다.

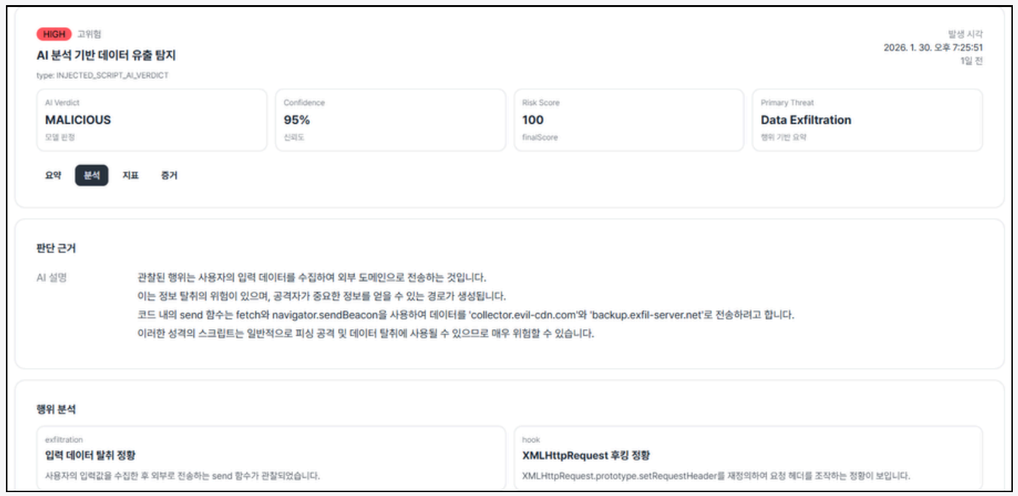
3) 스크립트 주입 룰이 간단 난독화에 약해지는 문제

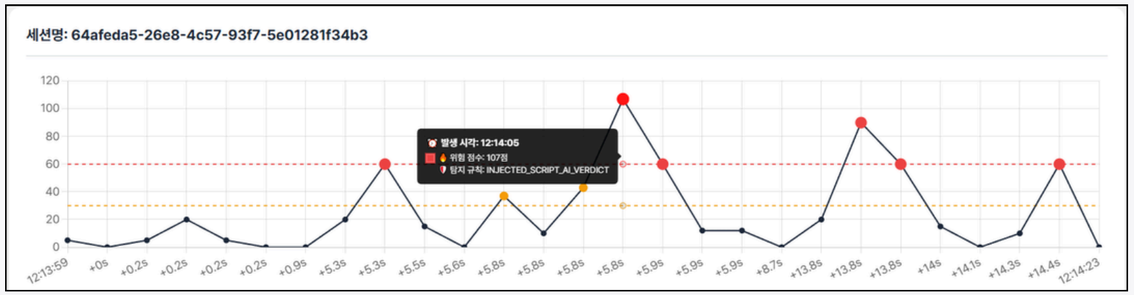
초기에 스크립트 주입 탐지는 문자열 패턴 중심으로 점수화했는데,
PoC-H에서 민감 API 이름을 감추는 수준의 간단 난독화만 적용해도 탐지 점수가 과도하게 낮아지는 문제가 드러났다.
이를 해결하기 위해 룰을 문자열 형태가 아니라 공격이 유지되면 숨기기 어려운 불변 신호 중심으로 재정의했다.
예를 들어 eval/new Function, 동적 <script> 삽입, 다중 오리진 URL 흔적 같은 신호를 분해해 점수화하고, 단일 신호가 약해져도 조합(combination)으로 승격되도록 보너스 룰을 추가했다.

TEAM PROJECT 1

Browser Runtime Security System



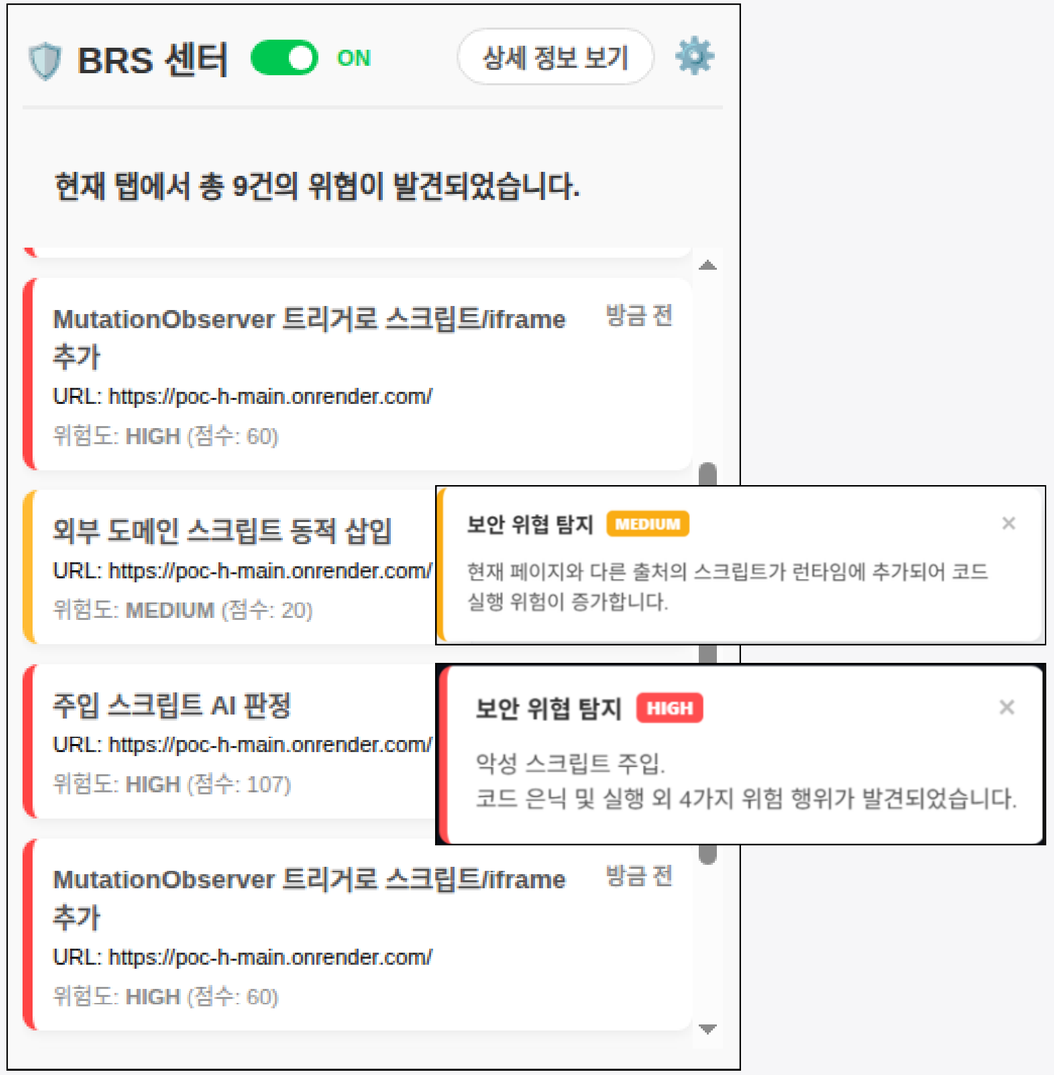


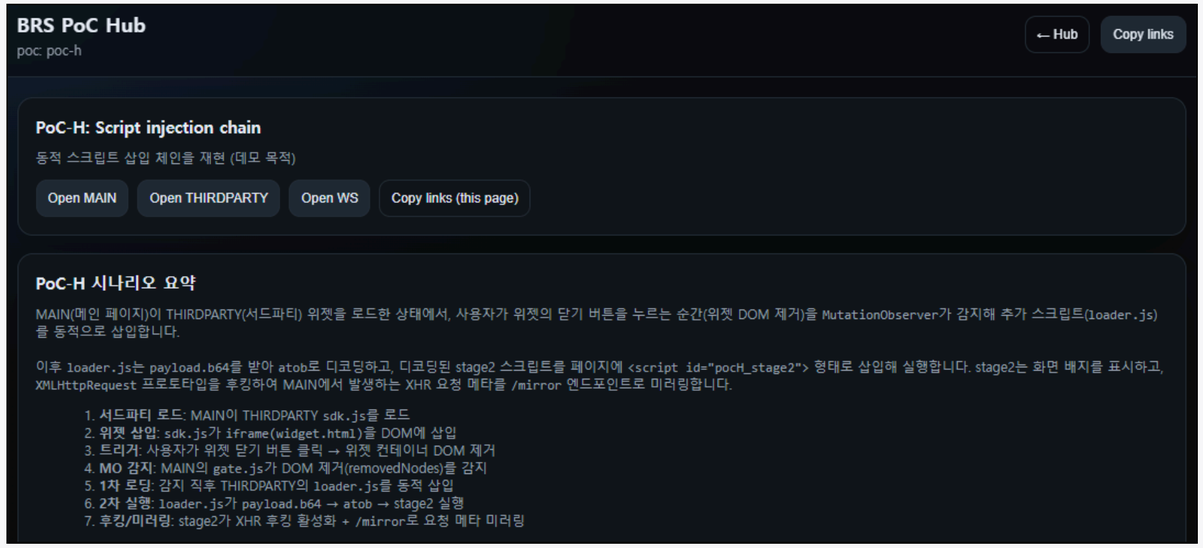


Detect → Event Capture → Ruleset Scoring(+AI) → Serverless Store → Dashboard

TEAM PROJECT 1

Browser Runtime Security System





TEAM PROJECT 2

Vulnerability Fingerprinting & CVE Mapping System

소개

네트워크 서비스의 포트·서비스 식별 → 버전 Fingerprinting
→ CVE 자동 매핑 → Exploit 후보 추천까지 이어지는
공격자 관점의 자동 취약점 분석 시스템

기간 / 인원

- 25.11.19~25.12.09
- 지원자 포함 보안 연구자 3인 팀 프로젝트

핵심 기능

- Nmap 기반 커스텀 포트 스캐너를 활용한 포트 스캔
- 서비스 Fingerprinting 엔진을 통한 서비스 식별
- 서비스 별 CVE 자동 매핑
- CVE를 기반으로 한 Exploit 취약점 후보 분석

팀 성과

- Nmap 기반 커스텀 포트 스캐너 개발
- 서비스 Fingerprinting 엔진 구축
- CVEDB를 활용한 CVE 자동 매핑 구축
- Exploit-DB를 활용한 Exploit 취약점 후보 분석 자동화

지원자 개인 기여

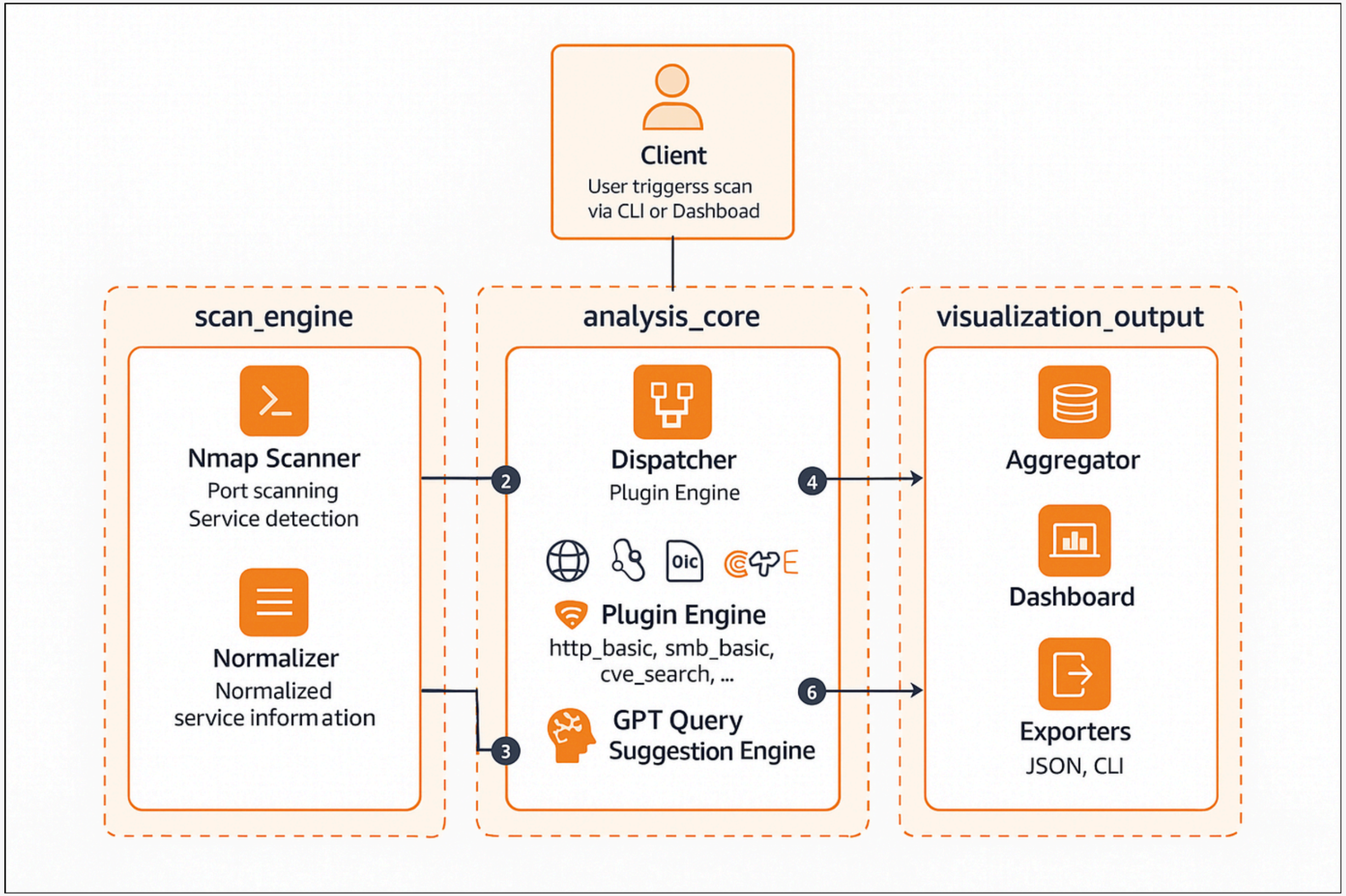
역할 요약 : 팀 리더, DB 구축, CVE 관련 구축 및 Exploit 테스트 진행

세부 내용

- AWS EC2 (CVE-DB) 구축
- 스캐너 서비스 - CVE 매칭 구축
  - Local CVE API 연동 → 서비스명+버전 기반 취약점 검색
- exploit TTP 개발
  - Exploit-DB/PoC 존재 여부 기반 실제 공격 가능성 평가
- Metasploitable2 Exploit 테스트 진행
- AI 기반 정확한 버전 Fingerprinting 구축
  - GPT 4 프롬포트를 활용한 서비스 분류

TEAM PROJECT 2

Vulnerability Fingerprinting & CVE Mapping System



TEAM PROJECT 2

Vulnerability Fingerprinting & CVE Mapping System

핵심 구현 기술 스택

- Nmap 기반 포트 스캐너 커스텀 구현
  - 열린 포트 자동 탐지 / 서비스 정보 추출
  - Service Scan + Banner 분석으로 버전 정확도 향상
- 서비스 Fingerprinting 엔진 개발
  - 배너 문자열 파싱
  - AI 기반 서비스/버전 모호성 매칭 로직 구현
- CVE 자동 매핑 엔진 구축
  - Local CVE API 연동 → 서비스명+버전 기반 취약점 검색
  - 연도·정확도·CVSS 기반 필터링/정렬
- Exploit 후보 분석 모듈 구현
  - Exploit-DB/PoC 존재 여부 기반 실제 공격 가능성 평가
- 결과 대시보드(스트림릿) 시각화
  - 서비스/버전 → CVE 매핑 → Exploit 우선순위 자동 정렬

주요 기능 흐름

1. 네트워크 서비스의 포트·서비스 식별
2. 버전 Fingerprinting
3. CVE 자동 매핑
4. Exploit 후보 추천
5. 대시보드 시각화 출력

기능 상세 설명

1. Port Scanning - 열린 포트 + 서비스 탐지
   - 전체 포트 스캔 및 Service Scan으로 서비스 추출
   - DB 기반 포트-서비스 매칭 보정
2. Service Fingerprinting - 정확한 서비스·버전 식별
   - 배너 기반 응답 수집
   - AI 기반 버전 문자열 매칭
3. CVE 자동 매핑 - 서비스 버전 기반 취약점 자동 탐색
   - Local CVE API로 취약점 질의 및 CVSS, EPSS 기준 정렬
   - 서비스에 해당하는 취약점 자동 제시
4. Exploit 후보 분석 - 실제로 공격 가능한 취약점 선별
   - Exploit-DB/PoC 존재 여부 확인
   - 환경 적용 가능성 판단
5. Dashboard - 분석 결과를 한눈에 확인
   - 포트·서비스·버전 정리
   - 취약점 매핑 결과 표시

핵심 성과

- AI 기반 정확한 버전 확인으로 nmap 대비 약 40% CVE 매핑 정확도 강화
- 서비스 기준 자동 CVE 추천 및 캐시 활용 → 약 60%(1m->20s) 분석 시간 단축
- Exploit 후보 자동 정렬로 공격·대응 방향 즉시 파악 가능
- Metasploitable2에서 다수 서비스의 취약점 식별 및 PoC 기반 성공 검증

TEAM PROJECT 2

Vulnerability Fingerprinting & CVE Mapping System

트러블 슈팅 과정

1) Nmap의 버전 문자열 노이즈 때문에 CVE 매핑 정확도가 떨어지는 문제

Nmap -sV 결과의 version 필드는 배포판/프로토콜/불필요 키워드가 섞인 형태가 많아(예: OS명, protocol, rpc 등), 그대로 CVE 검색 쿼리로 쓰면 결과가 과도하게 넓어지거나 엉뚱한 서비스로 매칭되는 문제가 있었다. 스캔은 정확히 했는데 검색 매칭 상태가 오염되어 CVE 후보 품질이 흔들리는 형태였다.
이를 해결하기 위해 version 문자열에서 의미 없는 토큰을 제거하는 정제 로직을 두고, 반복적으로 등장하는 서비스 조합에 대해서는 GPT가 생성한 검색용 쿼리 후보를 파일로 캐싱해 재사용하는 방식으로 보완했다. 이 구조를 통해 동일 서비스 재스캔 시 비용을 줄이면서도, 사람이 공격 관점에서 바로 쓸 수 있는 검색어 형태로 쿼리 품질을 안정화했다.

2) CVE 검색 API 호출이 폭주해 속도 저하/실패가 발생하는 문제

서비스/버전 조합이 많은 타깃을 스캔하면 CVE 조회 요청이 빠르게 연속 발생하면서, 로컬 CVE-Search API가 지연되거나 일시 실패하는 문제가 생겼다.
같은 (service, version)을 반복 조회하는 상황에서도 매번 API를 호출하면 불필요한 트래픽이 누적되고, 전체 분석 시간이 늘어났다.
이를 해결하기 위해 API 호출에 최소 간격을 두어 과도한 연속 호출을 억제했다. (service|version) 단위로 CVE 결과를 로컬 파일 캐시에 저장해 동일 조합은 재호출하지 않도록 했다. 추가로 CVE별 위험도 보강을 위한 EPSS 조회도 별도 캐시로 분리해, CVE 수 × 반복 실행에 따른 비용 폭주를 막았다.

3) SMB 열거가 포트마다 중복 실행되어 스캔 시간이 불필요하게 늘어나는 문제

초기에는 139/445가 열려 있는 타깃에서 SMB 관련 정보를 얻기 위해 포트별로 열거 로직을 반복 호출하는 방식이 되었고, 결과적으로 동일한 nmap 스크립트가 여러 번 실행되어 전체 스캔 시간이 늘어났다. 특히 smb-os-discovery, smb2-security-mode 같이 네트워크 왕복이 많은 스크립트는 포트 수만큼 반복 실행될 때 체감 지연이 컸다.
이를 해결하기 위해 SMB를 per-service가 아니라 전역(global) 열거로 분리해, 서비스 리스트를 기반으로 SMB가 존재하는지 확인한 뒤 한 번만 nmap 스크립트를 실행하도록 바꿨다. 이로써 139/445가 동시에 열려 있어도 SMB 정보 수집은 1회로 제한되며, 결과는 global_enum으로 모아 후속 TTP 엔진에서도 재사용 가능하게 했다.

TEAM PROJECT 2

Vulnerability Fingerprinting & CVE Mapping System

```
=== TTP Suggestions ===
 - [HIGH CVE] ftp on port 21 → CVE-2011-2523 (score=10.0)
 - [HIGH CVE] ssh on port 22 → CVE-2014-6271 (score=10.0)
 - [HIGH CVE] telnet on port 23 → CVE-2000-0166 (score=10.0)
 - [HIGH CVE] domain on port 53 → CVE-2008-0122 (score=10.0)
 - [HIGH CVE] rpcbind on port 111 → CVE-1999-0213 (score=10.0)
 - [HIGH CVE] netbios-ssn on port 139 → CVE-2003-0085 (score=10.0)
 - [HIGH CVE] netbios-ssn on port 445 → CVE-2003-0085 (score=10.0)
 - [WEB] Apache + Tomcat 구조 감지. → /manager, /host-manager, 디폴트 크리덴셜, WAR 업로드 RCE 가능성 체크.
 - [WEB] Tomcat + AJP(ajp13) 조합 감지. → AJP 파일 열람 / Ghostcat 계열 취약점 여부 확인.
 - [WEB] 파라미터 주입 테스트에서 49 계산 결과가 노출됨 → SSTI 의심 엔드포인트 존재.

=== Attack TODO ===
 - [FTP] 21/tcp vsftpd 2.3.4 → CVE-2011-2523 backdoor 테스트 (포트 6200/tcp로 직접 접속 시도).
 - [CVE] ssh 22/tcp → CVE-2014-6271 (score=10.0) 관련 exploit 문서 확인.
 - [CVE] telnet 23/tcp → CVE-2000-0166 (score=10.0) 관련 exploit 문서 확인.
 - [CVE] domain 53/tcp → CVE-2008-0122 (score=10.0) 관련 exploit 문서 확인.
 - [CVE] rpcbind 111/tcp → CVE-1999-0213 (score=10.0) 관련 exploit 문서 확인.
 - [CVE] netbios-ssn 139/tcp → CVE-2003-0085 (score=10.0) 관련 exploit 문서 확인.
 - [CVE] netbios-ssn 445/tcp → CVE-2003-0085 (score=10.0) 관련 exploit 문서 확인.
 - [AJP] 8009/tcp → AJP 엔드포인트 대상 Ghostcat/파일 열람 PoC 시도.
 - [WEB] SSTI 의심 파라미터에 대해 템플릿 엔진별 payload 확장 및 RCE 체인 검토.
```

```
================================================================================
Service : drb
Version : Ruby DRb RMI (Ruby 1.8; path /usr/lib/ruby/1.8/drb)
[*] Using cached suggestions:
  1. ruby drb 1.8
  2. ruby drb
  3. drb 1.8
  4. ruby 1.8 drb
  5. ruby remote method invocation
================================================================================
```

```
=== Scan Meta ===
 - Started : 2025-12-08T03:50:16
 - Finished: 2025-12-08T03:50:28
 - Duration: 11.56 sec
 - Cmd     : nmap -sT -sV -Pn -p- --min-rate 1000 192.168.96.133
```

🔍 Exploit-DB Online Lookup

Search Exploit-DB

edb_id	title	date	verified
39887	Sun Secure Global Desktop and Oracle Global Desktop 4.61.915 - Command Injection (She		☐
39568	Cisco UCS Manager 2.1(1b) - Remote Command Injection (Shellshock)		☐
34900	Apache mod_cgi - 'Shellshock' Remote Command Injection		☑
36933	dhclient 4.1 - Bash Environment Variable Command Injection (Shellshock)		☑
34860	GNU bash 4.3.11 - Environment Variable dhclient		☐

상위 Exploit-DB 항목 바로가기:

Open EDB-39887: Sun Secure Global Desktop and Oracle Global Desktop 4.61.915 - Command Injection (Shellshock)

Open EDB-39568: Cisco UCS Manager 2.1(1b) - Remote Command Injection (Shellshock)

Open EDB-34900: Apache mod_cgi - 'Shellshock' Remote Command Injection

View on NVD Official

TEAM PROJECT 2

Vulnerability Fingerprinting & CVE Mapping System

NOT FOUND Integrated Dashboard

Security Insight: 현재 보안 상태에 대한 직관적인 요약입니다.

Executive Summary

Total Vulnerabilities
526

Critical Risks
68

Exploitable
48

Avg CVSS
6.0

Vulnerability Main Themes

사용 가이드 및 분석 팁

이 섹션은 스캔된 서버의 주요 취약점 키워드를 직관적으로 보여줍니다. 붉은색 태그는 가장 빈번하게 발견된 핵심 문제입니다.

which (201) cause (188) denial (175) users (166) through (164)
server (135) when (107) execute (106) function (93) local (93)
crash (88) file (86)

Dashboard Calendar Statistics Comparison Deep Analysis Remediation FAQ

Interactive Analysis

CVE_NAME	PRODUCT_NAME	SEVERITY	CVSS_SCORE	EPSS_SCORE	DETECT_DATE_STR
CVE-2011-2523	ftp (vsftpd 2.3.4)	Critical	10	0.9427	2025-09-29
CVE-2011-2189	ftp (vsftpd 2.3.4)	High	7.8	0.0725	2025-02-23

Deep Vulnerability Analysis

Select Vulnerability

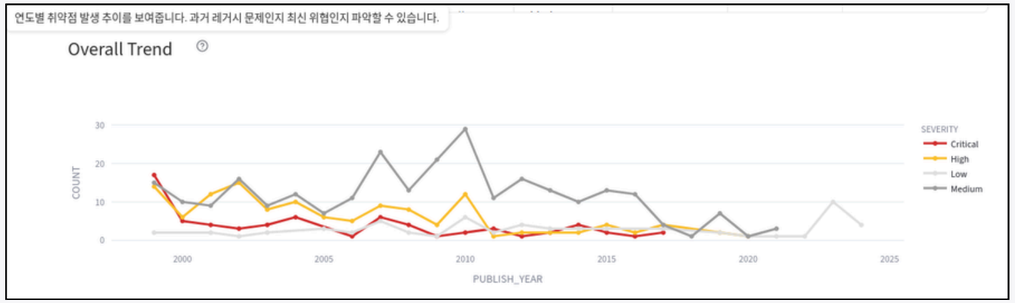
CVE-2014-6278

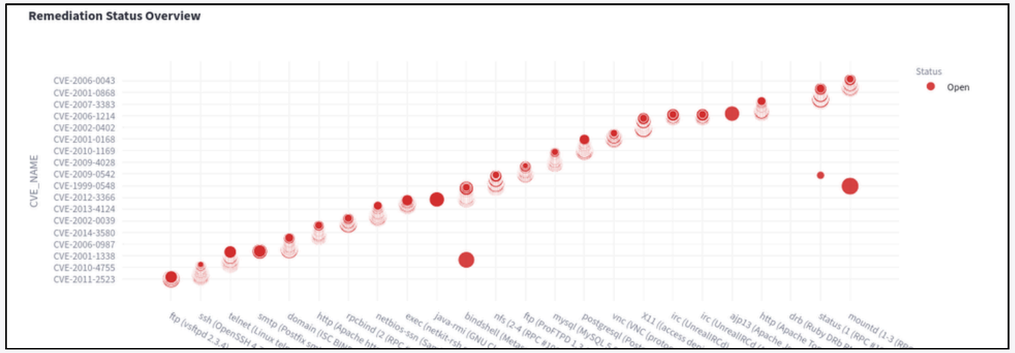
CVE-2014-6278

Product: ssh (OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)) | Port: 22

GNU Bash through 4.3 bash43-026 does not properly parse function definitions in the values of environment variables, which allows remote attackers to execute arbitrary commands via a crafted environment, as demonstrated by vectors involving the ForceCommand feature in OpenSSH sshd, the mod_cgi and mod_cgid modules in the Apache HTTP Server, scripts executed by unspecified DHCP clients, and other situations in which

0% 10.0 100%





TEAM PROJECT 3

Darkweb Threat Intelligence Automation System

소개

다크웹·해킹포럼·랜섬웨어 페이지에 게시되는
기업 데이터 유출 정보를 자동으로 수집·정규화·분석·알림하기 위한
지능형 위협 인텔리전스 자동화 시스템

기간 / 인원

- 25.10.13~25.10.29
- 지원자 포함 보안 연구자 4인 팀 프로젝트

핵심 기능

- Tor 기반 자동 크롤링
- 정규화 엔진을 통한 데이터 구조화
- Dashboard 를 통한 데이터 시각화
- AWS 관련 유출 발생 시 자동 키 폐기 및 즉시 대응
- 신규 유출 탐지 시 Webhook 자동 알림 전송

팀 성과

- Tor 기반 데이터 크롤링 자동화 엔진 구축
- 도메인별 파서(Parser) 엔진 설계
- Discord DM을 활용한 실시간 경보 체계 구축
- Dashboard 시각화 구축

지원자 개인 기여

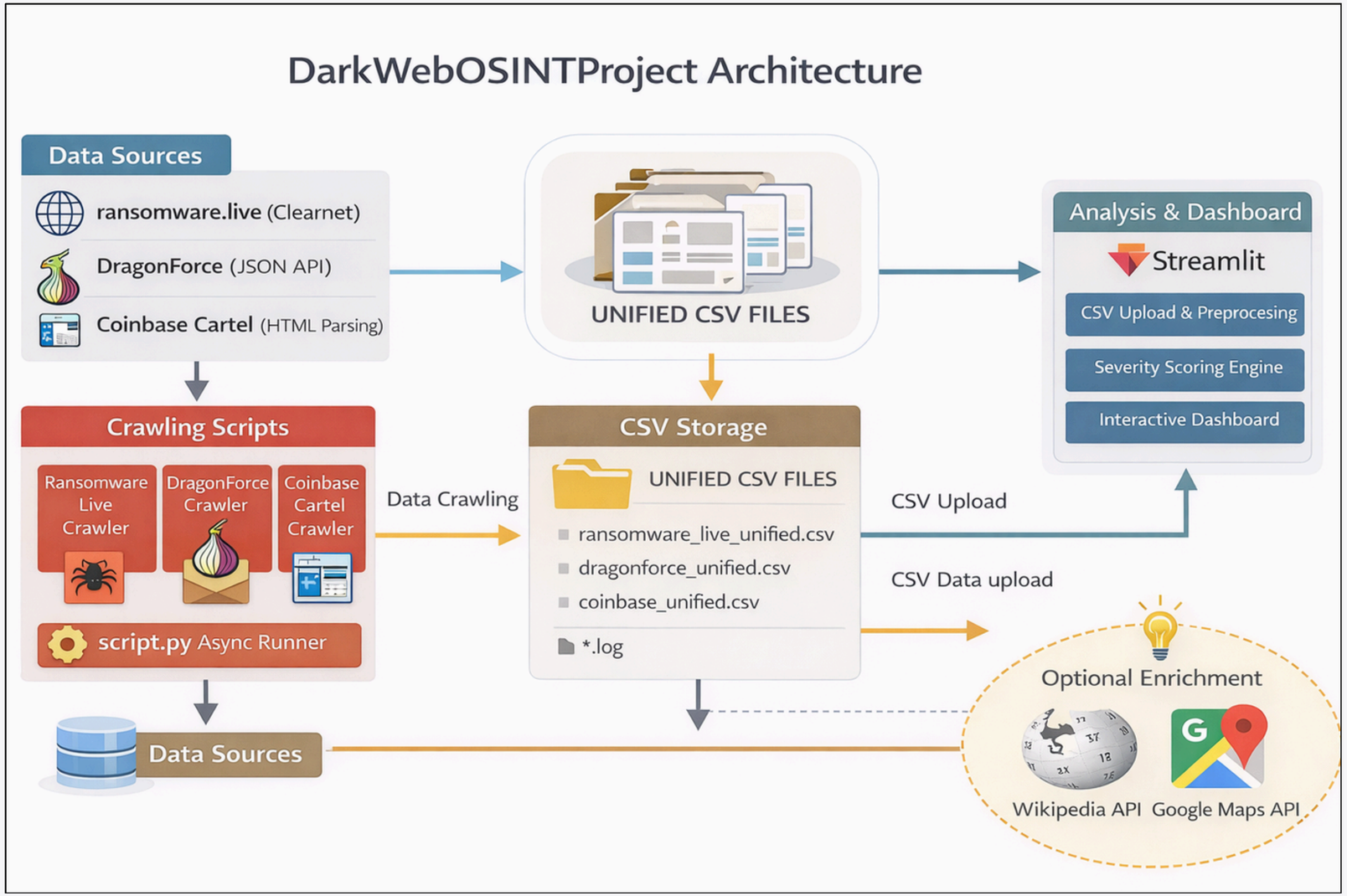
역할 요약 : 팀 리더, 알람 구축, 침해 대응 구축, 대시보드 구축, 최적화

세부 내용

- 유출 사고 대응(AWS, 도메인) 설계
  - AWS API를 활용한 유출 사고 발생 시 자동 키 교환 구축
- 디스코드 알람 연동 및 구현
  - Slack/Discord 포맷 자동 생성 및 메시지 전송 연동
- 크롤링 데이터 통합
- 시각화 대시보드 구축
  - Streamlit 활용

TEAM PROJECT 3

Darkweb Threat Intelligence Automation System



TEAM PROJECT 3

Darkweb Threat Intelligence Automation System

핵심 구현 기술 스택

- Tor 기반 크롤링 엔진 구축
  - Tor Session / SOCKS5 프록시 / 우회 지연 처리
- 도메인별 파서(Parser) 엔진 설계
  - HTML / JSON / 텍스트 구조 자동 분석 및 규칙 기반 템플릿 생성
- 정규화(Normalization) 모듈 구현
  - 누락 필드 보완, 날짜 파싱, 국가 코드 ISO-3 매핑
- Streamlit 기반 Dashboard 개발
  - Attack Type / Country / Date Range Filtering
  - 실시간 검색 및 데이터 시각화
- Webhook 실시간 알림 시스템 구축
  - Slack/Discord 포맷 자동 생성 및 메시지 전송

주요 기능 흐름

1. Tor 기반 자동 크롤링
2. 정규화 엔진에서 구조화
3. Dashboard 시각화
4. 신규 유출 탐지 시 Webhook 자동 알림 전송

기능 상세 설명

1. 데이터 크롤링 자동화
   - 악성 포럼/유출 사이트 다중 도메인 크롤링
   - CAPTCHA 우회 사이트 제외 처리
   - 실패 페이지 재시도 및 딜레이 랜덤화 기반 차단 회피
2. 정규화·전처리 엔진
   - 게시글에서 기업명, 공격 그룹, 공격 유형, 국가, 날짜, 데이터 규모 자동 추출
   - 공격 분류 체계 분류: Ransomware / Leak / Sale / Credential Dump 등
3. Dashboard 시각화
   - 국가별 유출량 지도
   - 공격 유형별 통계 차트
   - 기간 필터링 / 다중 조건 검색
4. 실시간 경보 체계
   - 신규 유출 탐지 → 즉시 Slack/Discord 알림
   - 기업명 기반 키워드 매칭 후 위험도 기반 우선순위 지정

핵심 성과

- 5개 이상 다크웹 소스 자동화 성공
- 평균 수집 속도 30% 향상, 누락률 40% 감소(기존 테스팅 기준)
- 신규 유출 정보 발생 시 30초 내 Slack/Discord 알림
- AWS IAM 액세스키 자동 로테이션·검증·은퇴 모듈 구현

TEAM PROJECT 3

Darkweb Threat Intelligence Automation System

트러블 슈팅 과정

1) Tor 크롤링이 연결된 것처럼 보이는데 실제로는 실패/누락되는 문제

PoC 단계에서는 Tor를 붙였다고 생각했는데, 환경에 따라 수집이 간헐적으로 0건이 되거나 timeout이 연속으로 발생했다.
원인은 Tor 프록시 포트가 OS/실행 환경에 따라 달라(Windows Tor Browser=9150, Linux/macOS=9050) 동일 설정으로는 정상 연결이 보장되지 않고, .onion은 DNS 처리 방식이 잘못되면(프록시가 DNS를 못 잡으면) 연결 실패처럼 보이는 케이스가 반복된다는 점이었다.
이를 해결하기 위해 크롤러에서 OS 기반으로 Tor 포트를 자동 분기하고(9150/9050), 프록시 URL도 DNS를 Tor로 넘기는 socks5h://로 통일해 .onion 해석을 안정화했다. 또한 요청 단에서 timeout을 명시하고, httpx transport 레벨 retries를 설정해 일시적인 Tor 라우팅 지연을 완화했다.

2) 다크웹 포럼이 Verification/DDOS 보호 페이지로 바뀌어 파싱이 깨지는 문제

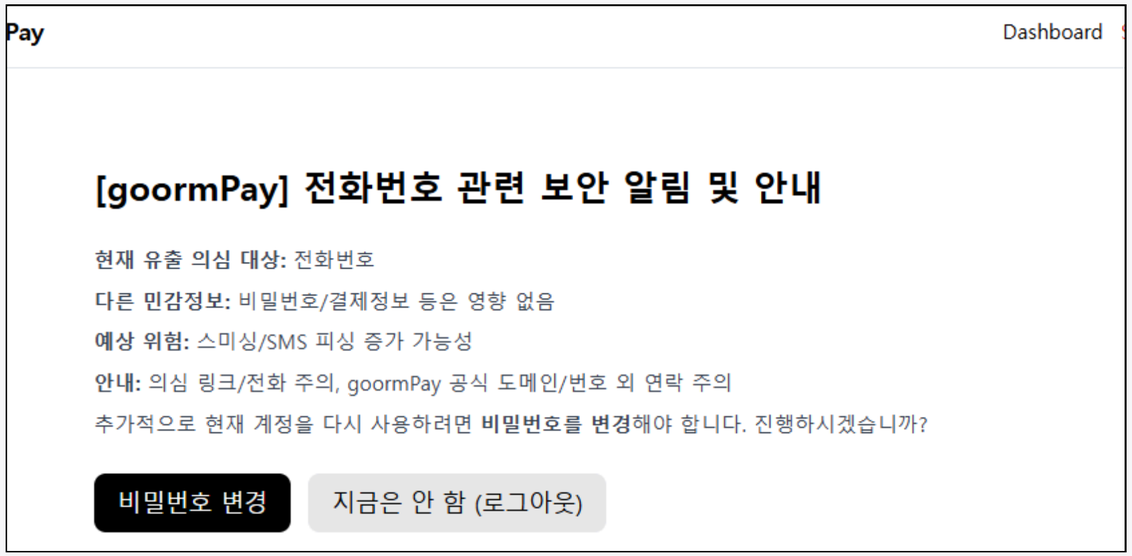
다크웹 포럼 크롤링은 정상 HTML을 기대하고 파싱하는데, 일정 시점부터 결과가 비어버리거나(게시물 0건) HTML 구조가 맞지 않아 파싱이 연쇄 실패하는 문제가 있었다.
원인은 사이트가 종종 Verification Requested 같은 보호 페이지로 전환되고,
이 상태에서는 DOM이 완전히 달라져 기존 셀렉터 기반 파싱이 무의미해진다는 점이었다.
이를 해결하기 위해 Selenium 기반 크롤러에 Verification 감지 로직을 넣고, 보호 페이지가 감지되면 사용자에게 해결을 안내한 뒤 Enter 입력으로 재개하도록 했다.

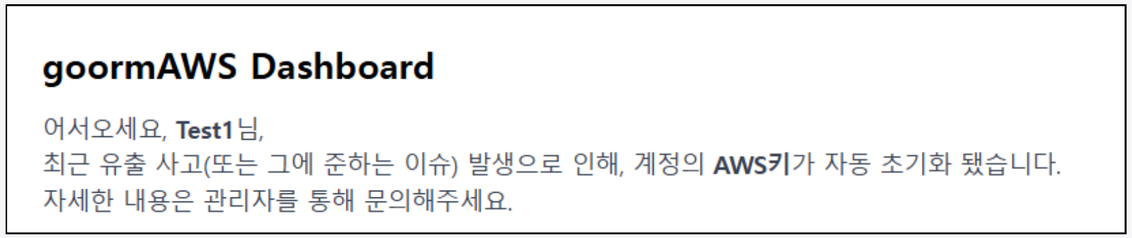
3) 국가/도메인 정규화가 흔들려 지도·통계가 틀어지는 문제

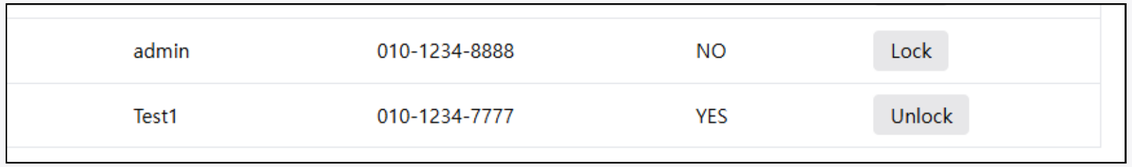
소스별로 회사/도메인/국가 값이 제각각이라, 대시보드에서 국가별 지도/통계를 만들 때 국가 매핑이 비거나 엉뚱한 나라로 잡히는 문제가 있었다.
특히 게시글 본문이나 첨부 링크에 mega.nz, pastebin, t.me 같은 파일 호스팅/소셜 도메인이 섞이면, 이 도메인들이 피해자 도메인으로 오인되는 경우가 많았다.
이를 해결하기 위해 정규화 단계에서 피해자 후보 도메인을 최대한 많이 수집하되, .onion 및 파일호스팅/커뮤니티/소셜 도메인은 대량 제외(EXCLUDE_DOMAINS)하여 오탐을 줄였다. 이후 남은 후보들에 대해 TLD/SLD 기반 국가 추정 투표를 적용하고(예: co.kr, co.uk 등), 그래도 비는 경우에는 회사명/본문 패턴(법인 표기 등) 휴리스틱으로 보조 추정했다. 결과적으로 지도/통계에서 ISO3 커버리지를 높이고 엉뚱한 국가로 찍히는 케이스를 줄였다.

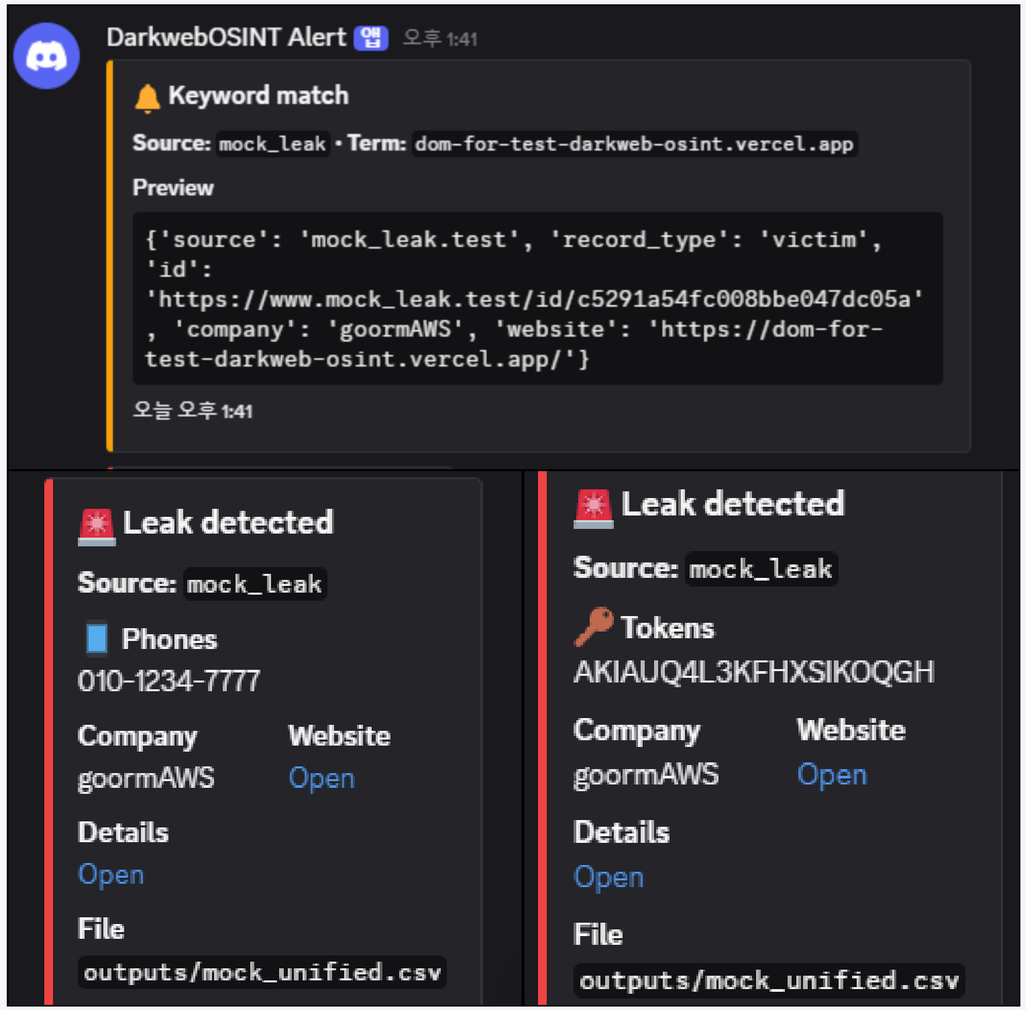
TEAM PROJECT 3

Darkweb Threat Intelligence Automation System



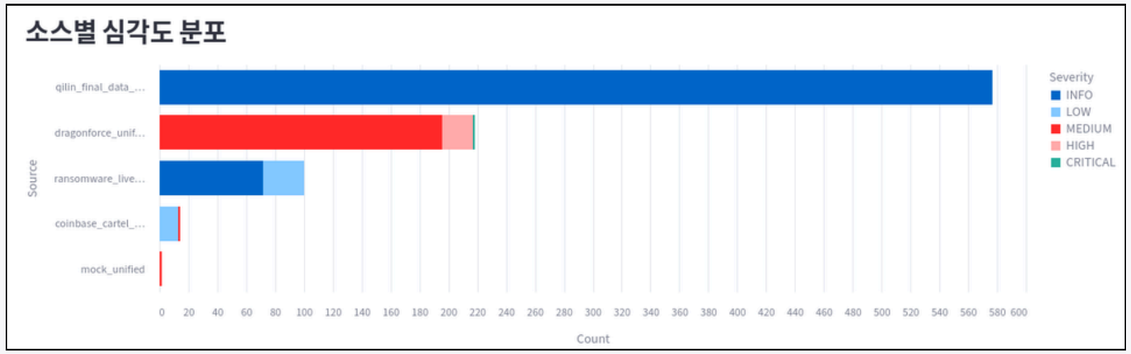


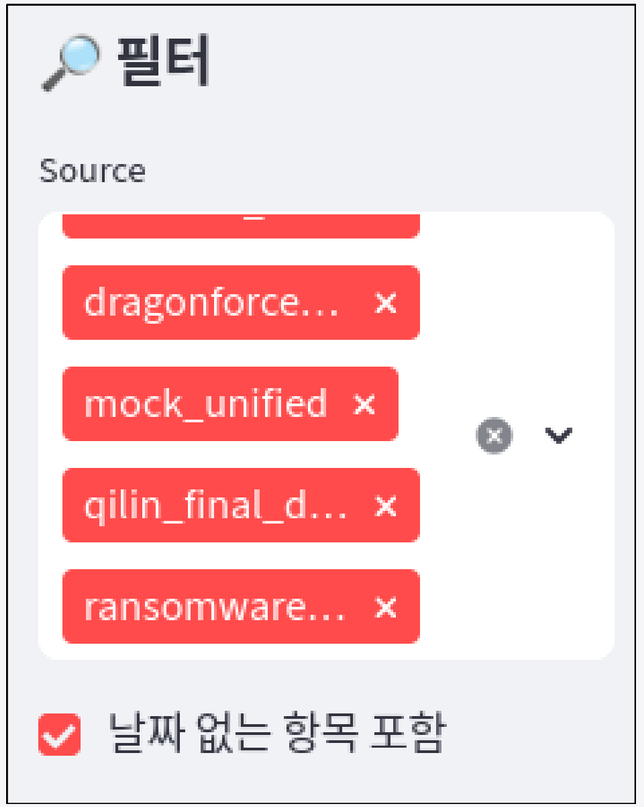


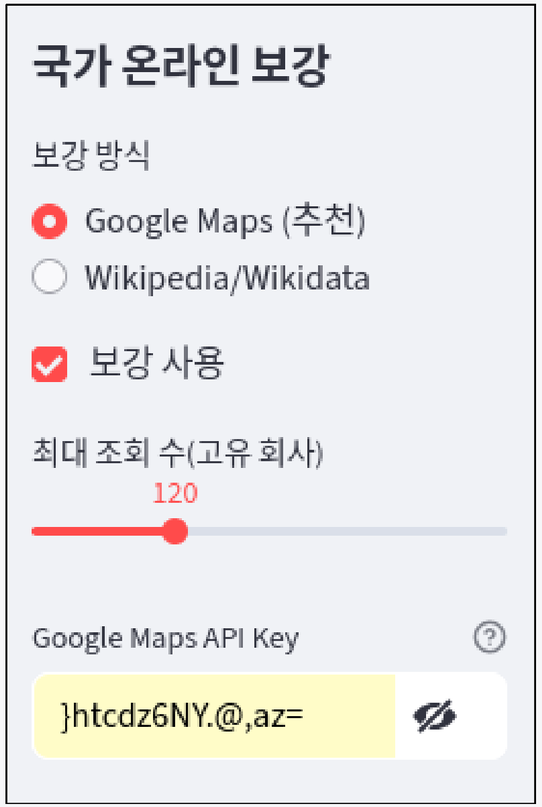


TEAM PROJECT 3

Darkweb Threat Intelligence Automation System







Crawler → Parser → Normalize → Dashboard → Alert

TEAM PROJECT 4

Mobile Malware Dynamic Analysis System

소개

모바일 악성 앱의 동적 행위를 자동 분석하고,
주요 API 호출·암호화 루틴·네트워크 트래픽을 실시간 추출하기 위한
자동화 분석 시스템

기간 / 인원

- 25.10.30~25.11.18
- FE 1인, 지원자 포함 보안 연구자 4인 팀 프로젝트

핵심 기능

- 앱 복호화 및 재서명 및 감지 로직 우회 어플 컴파일
- MobSF 활용 정적분석
- Frida 활용 동적분석
- 야라룰 및 바이러스 토탈 악성코드 앱 패밀리 식별

팀 성과

- apktools와 smali를 활용한 감지 로직 우회 자동화
- Frida를 활용한 앱 로직 후킹 및 분석 자동화
- 바이러스 토탈 API를 활용한 악성코드 앱 패밀리 식별 자동화

지원자 개인 기여

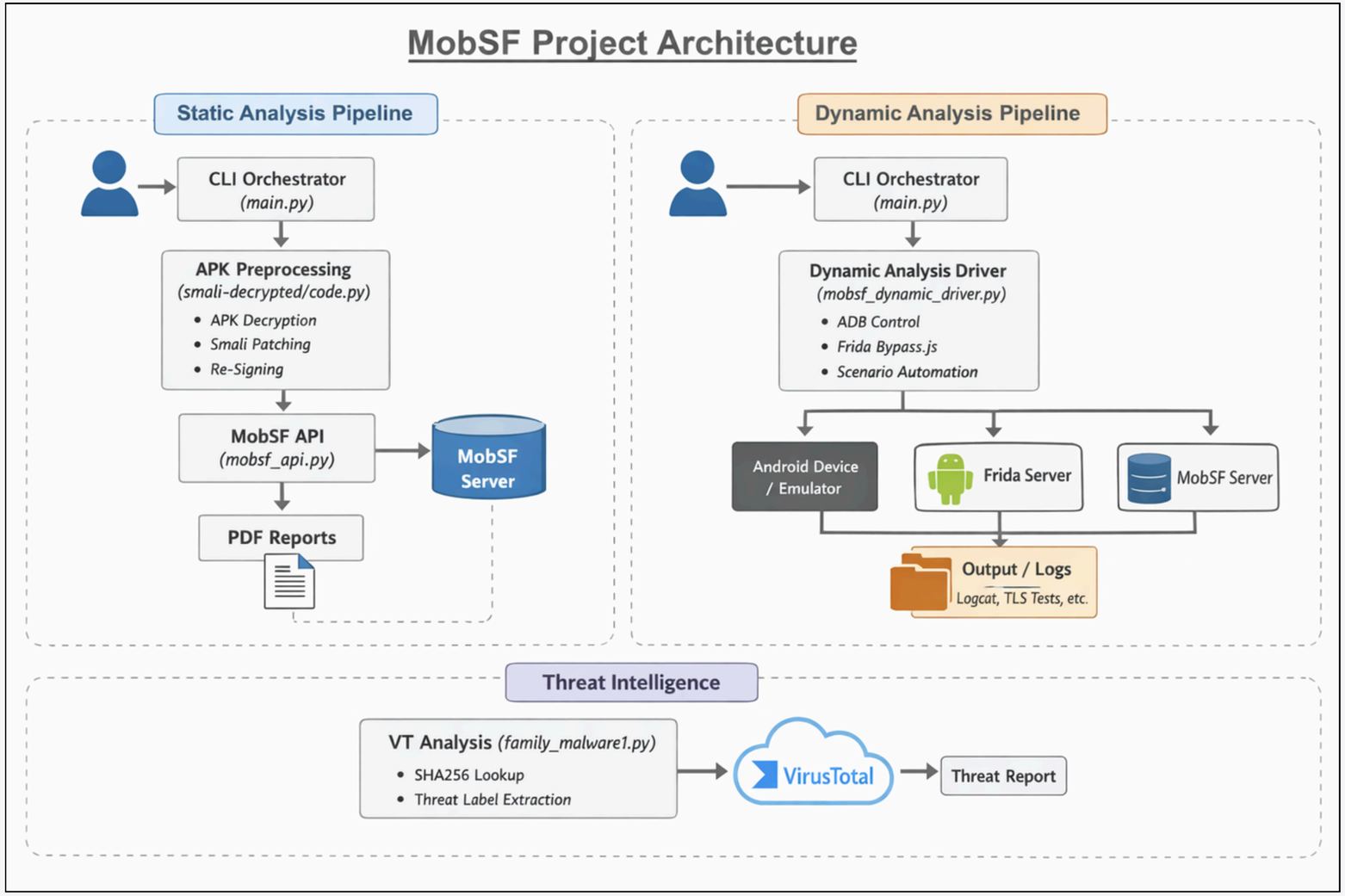
역할 요약 : 팀 리더, 동적 분석, 파이프라인 구축, 최적화

세부 내용

- Frida 기반 Hooking 엔진 개발
  - 암호화 키·네트워크 패킷·동적 로드 함수 실시간 추출
- MobSF API 자동화 파이프라인 구축
  - APK 업로드 → 정적 분석 → 동적 분석 → 리포트 자동 생성
- 자동 DEX 복호화 및 Repacking 모듈 개발

TEAM PROJECT 4

Mobile Malware Dynamic Analysis System



TEAM PROJECT 4

Mobile Malware Dynamic Analysis System

핵심 구현 및 기술 스택

- MobSF 분석 자동화 파이프라인 구축
  - APK 업로드 → 정적 분석 → 동적 분석 → 리포트(PDF/로그) 자동 생성 (MobSF REST API)
- 자동 DEX 복호화 및 Repacking 모듈 개발
  - AES 기반 DEX 복호화 → smali 패치(감지/차단 로직 우회) → 재패키징/재서명(apktool, baksmali/smali, jarsigner)
  - assets 내부 nested APK까지 동일 전처리 파이프라인 적용
- Frida 기반 Hooking/우회 스크립트 적용
  - SSL Pinning/프록시/루팅·에뮬레이터·Frida 탐지 우회 + 런타임 행위 관측 (bypass.js)
- 기술 스택
  - Python / MobSF REST API / Frida / ADB / apktool / baksmali·smali / VirusTotal

주요 기능 흐름

1. APK 전처리(복호화·우회·재서명) 후 MobSF 자동 업로드
2. Frida 스크립트 자동 삽입 + 시나리오 기반 동적 실행
3. 동적 로그(logcat) 및 TLS 테스트 등 결과 수집
4. 정적 PDF + 동적 산출물 + VT 라벨(선택)로 최종 리포트 구성

기능 상세 설명

- 전처리 자동화
  - 암호화 DEX 복호화 및 탐지 로직 우회 패치로 분석 가능한 APK 변형 생성
- MobSF 분석 자동화
  - API 호출로 스캔/리포트 다운로드까지 일괄 처리
- 동적 행위 수집
  - ADB 제어 + 시나리오 실행 + Frida 우회로 런타임 행위/네트워크 점검 결과 확보

핵심 성과

- APK 1개 입력으로 복호화/우회/재서명 → MobSF 정적 PDF 리포트 생성까지 파이프라인화(main.py + apk_pipeline.py + mobsf_api.py)
- 상위 APK뿐 아니라 assets 내부 nested APK(pgsHZz.apk)까지 동일 전처리 흐름을 적용하도록 설계해, 다단 패키징(드로퍼 형태) 케이스 분석 가능
- 동적 분석에서 ADB 로그 수집 + 시나리오 기반 재현 + Frida 우회 주입을 한 번에 묶어 행위 재현 가능한 자동 실행 러너를 구현
- VirusTotal 조회 모듈을 별도로 제공해, MobSF 결과와 함께 패밀리/라벨 인텔리전스를 선택적으로 결합할 수 있게 구성
- 악성 행위 탐지 스크립트 5+개 개발

TEAM PROJECT 4

Mobile Malware Dynamic Analysis System

트러블 슈팅 과정

1) 동적 분석에서 설치/초기 트래픽이 누락되는 문제

동적 분석 자동화를 만들었는데, 설치 직후 발생하는 초기 네트워크 트래픽(권한 요청, 초기 API 호출)이 리포트에 안 잡히는 문제가 있었다.
원인은 MobSF 동적 분석은 start_analysis 이후 트래픽만 캡처하므로, 앱 실행/설치 시나리오를 먼저 돌리면 핵심 구간이 누락된다는 점이었다.
또 프록시 설정이 ADB 단 OS proxy와 MobSF가 제공하는 global proxy 설정 경로가 섞이면, 트래픽이 프록시를 타지 않아 캡처가 흔들리는 경우도 생겼다.
이를 해결하기 위해 순서를 고정했다. 먼저 디바이스 프록시를 MobSF 경로(/android/mobsfy, /android/global_proxy) 또는 강제 OS proxy(--force-proxy)로 설정한 뒤 상태를 로깅하고, 그 다음에 /dynamic/start_analysis를 호출하도록 했다. 그리고 설치 단계가 필요한 앱은 scenario_keys.yaml을 설치 시나리오로 분리해, start_analysis 이후에 설치/권한 처리가 실행되게 구성해 설치 구간 트래픽까지 포함되도록 했다.

2) Frida 우회 스크립트가 적용 됐다 안됐다 하는 문제

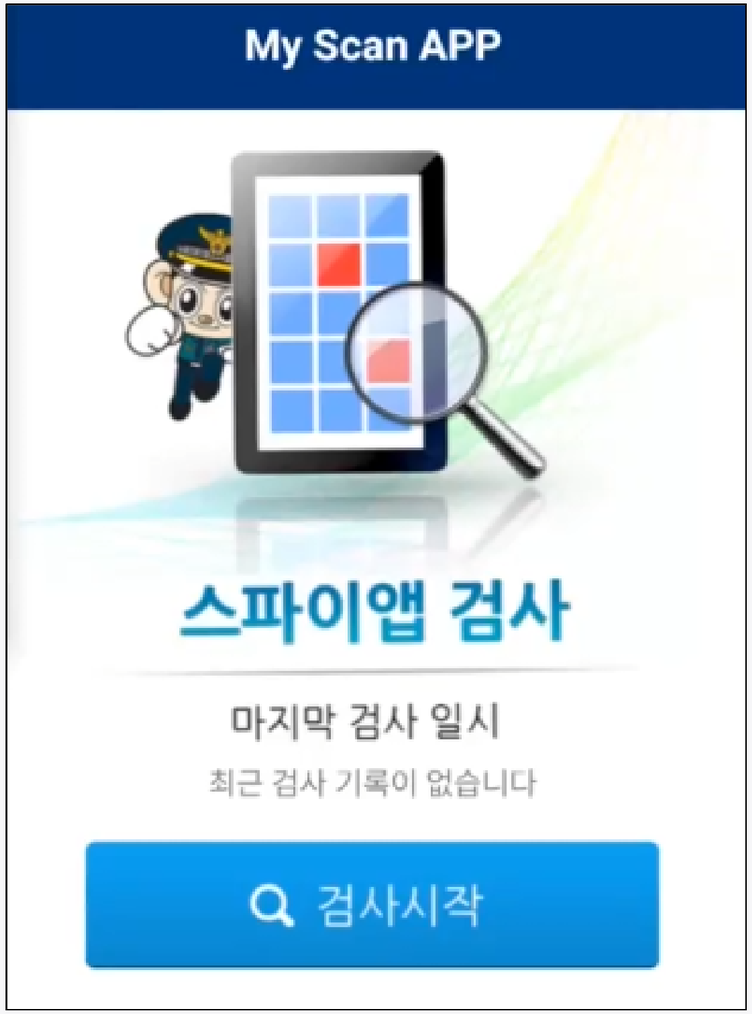
탐지 로직 우회를 위해 Frida 스크립트를 붙였는데, 동일 APK에서도 어떤 실행에서는 우회가 적용되고 어떤 실행에서는 초기 탐지 로직에 먼저 걸려 우회가 실패하는 문제가 있었다. 원인은 attach 방식으로는 앱의 초기 실행 구간을 이미 지나친 뒤 붙는 경우가 생기고, spawn 방식에서도 resume 타이밍이 맞지 않으면 후킹 적용 전에 코드가 진행될 수 있다는 점이었다.
이를 해결하기 위해 Frida 실행 모드를 분기해 제어했다. 기본은 spawn으로 두고 %resume를 명시적으로 호출해 후킹 로딩 → 실행 재개 순서를 강제했다. attach 모드는 필요한 경우에만 사용하고, 그때는 pidof로 PID를 확인한 뒤 attach하도록 하며, PID가 없으면 지정된 launch activity를 먼저 실행해 PID를 확보하는 보완 경로를 넣었다.
결과적으로 우회 적용의 재현성을 높이고 동적 분석 시나리오 실행이 안정화됐다.

TEAM PROJECT 4

Mobile Malware Dynamic Analysis System

PERMISSION	STATUS	INFO	DESCRIPTION
android.permission.BLUETOOTH_ADMIN	normal	bluetooth administration	Allows applications to discover and pair bluetooth devices.
android.permission.ACCESS_FINE_LOCATION	dangerous	fine (GPS) location	Access fine location sources, such as the Global Positioning System on the phone, where available. Malicious applications can use this to determine where you are and may consume additional battery power.
android.permission.ACCESS_COARSE_LOCATION	dangerous	coarse (network-based) location	Access coarse location sources, such as the mobile network database, to determine an approximate phone location, where available. Malicious applications can use this to determine approximately where you are.
android.permission.REQUEST_DELETE_PACKAGES	normal	enables an app to request package deletions.	Allows an application to request deleting packages.
android.permission.ACCESS_WIFI_STATE	normal	view Wi-Fi status	Allows an application to view the information about the status of Wi-Fi.

```
옵션] VirusTotal 추가 정적 인텔리전스 분석도 수행할까요? (Y/n): Y

VT] 선택된 APK에 대해 VirusTotal 인텔리전스를 조회합니다...

-- 1. VirusTotal API 분석 및 인텔리전스 수집 ---
 [대상 해시]: 82d644a1f3bba120327e7eb6029f6b986c95c35f0c40cd43001f2dbedee2ee6f
 [조회]: 완료된 최신 리포트를 즉시 로드했습니다.

================================================
     VirusTotal 상세 인텔리전스 보고서 (최종)
================================================

A] 기본 파일 정보 (Basic properties)
 - MD5: f90f81f7b47ca73de0e5aa5aaeba6735
 - SHA-256: 82d644a1f3bba120327e7eb6029f6b986c95c35f0c40cd43001f2dbedee2ee6f
 - 파일 크기: 33.50 MB (35132073 bytes)

B] 위험도 및 분류 (Detection)
 - 악성코드 탐지 수 (빨간불): 25 / 67개

 - **대표 분류명 (Raw)**: trojan.fakecalls/bankbot
 - **[분석] 악성코드 유형 (Type)**: Trojan (자신을 유용한 앱으로 위장해 침투)
 - **[분석] 주요 행위 (Behavior)**: BankBot (금융/뱅킹 정보 탈취 시도)
 - **[분석] 악성코드 계열 (Family)**: Bankbot (계열)
```



APK → Repack → Emulator → Frida Hook → Log → Report

김 태 중

Github

github.com/Jamongseed

Phone

010-4058-1219

Email

ronan1_@naver.com