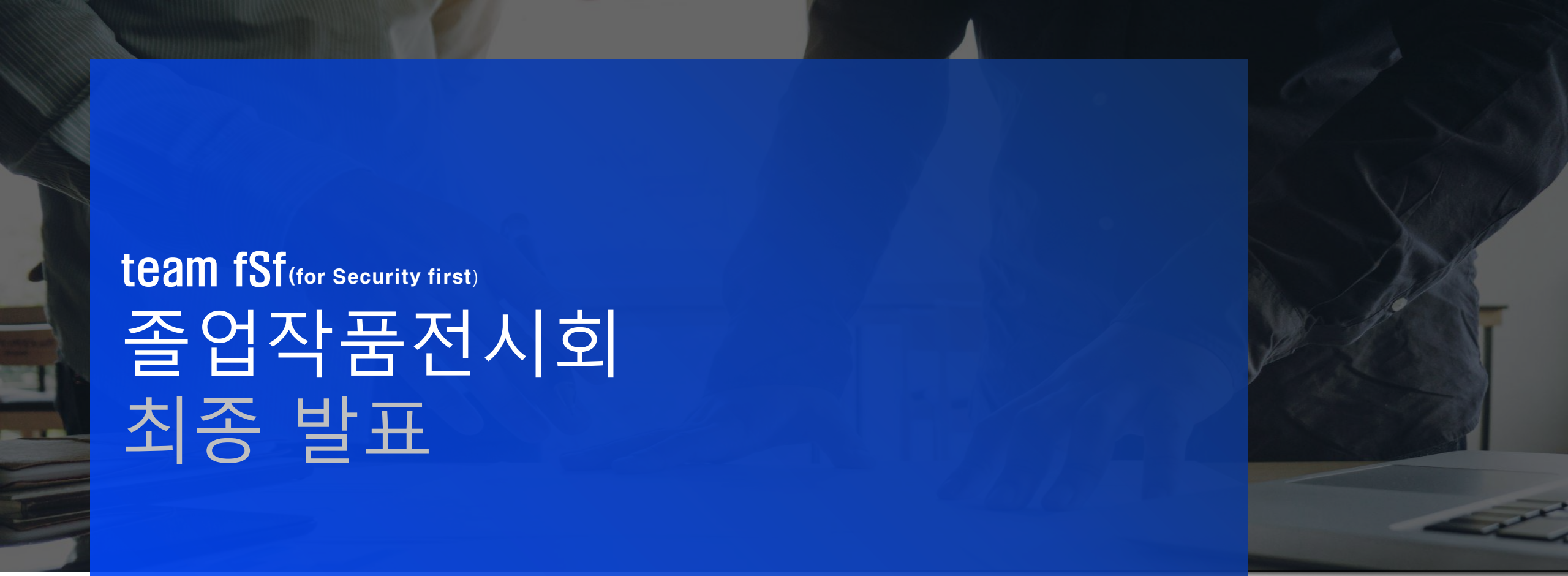


team fSf(for Security first)

졸업작품전시회
최종 발표

지도교수 : 양환석 교수님
팀장 : 김채원
팀원 : 손경현
: 곽화종
: 박세연

중부대학교

1

팀 개요 및 프로젝트 소개

01. 팀 개요 > 팀원 소개

김채원

- 프로젝트 팀장
- 악성코드 및 Elasticseach 쿼리문 제작
- 모의해킹 및 침해사고 분석 보고서 작성
- 팀원 통솔

곽화종

- 프로젝트 팀원
- 악성코드 제작 및 프로젝트 진행

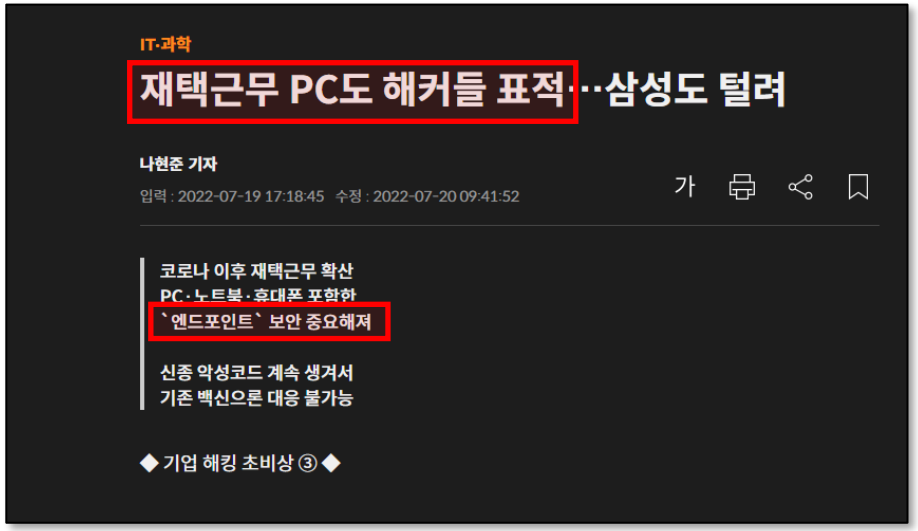
손경현

- 프로젝트 팀원
- Linux 관리 서버 제작
- Windows Agent Installer 제작
- IOC 기반 악성 파일제거 프로그램 제작

박세연

- 프로젝트 팀원
- 프로젝트 진행

01. 팀 개요 > 팀 프로젝트 소개



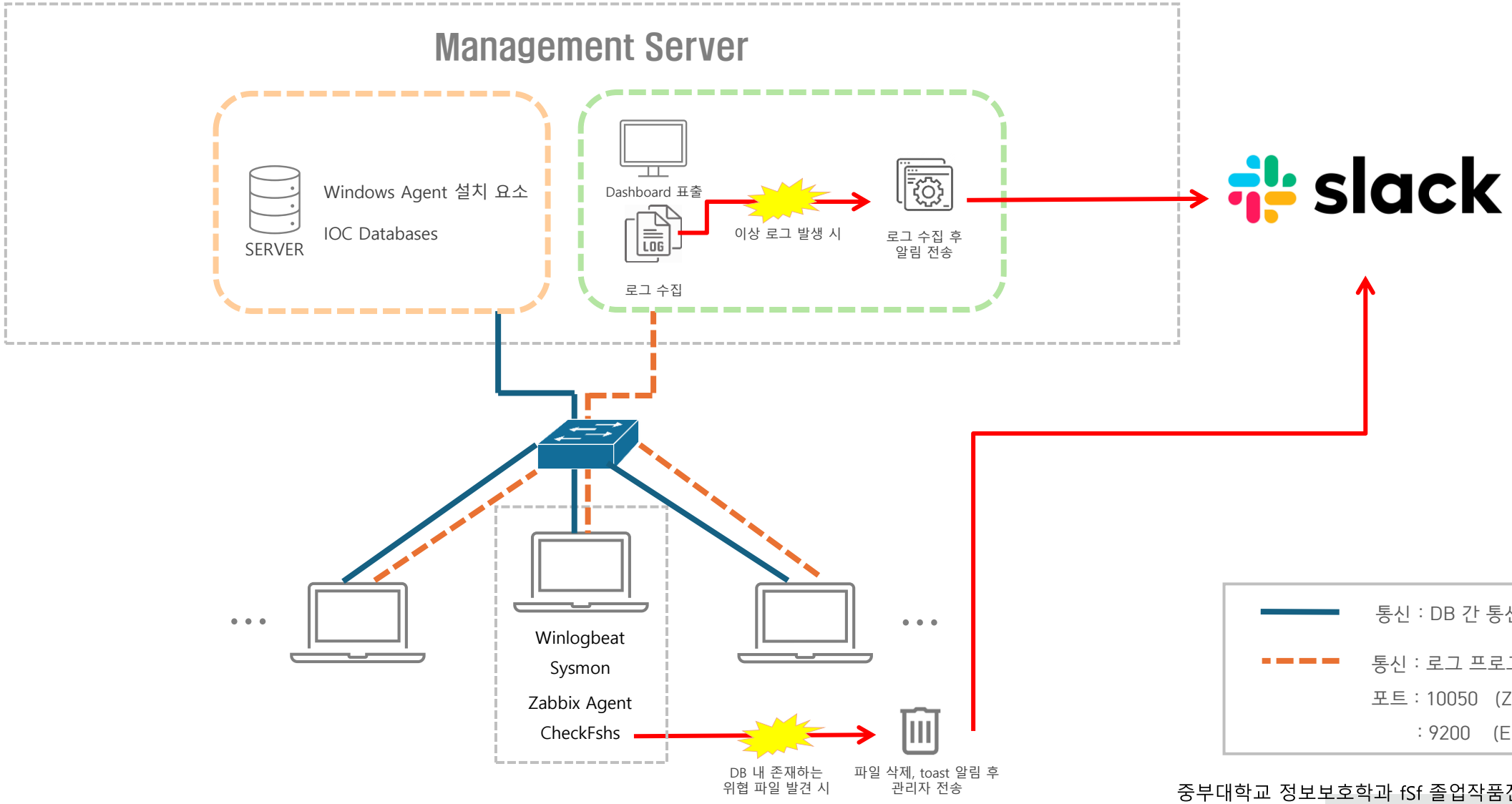
지속되는 사이버 침해 사고..
인지 또한 빨라져야 하며, 엔드포인트의 보안 또한 부각되고 있다 !

01. 팀 개요 > 주요 기능

- 엔드포인트 주요 이벤트 로그 자동 수집
- IOC 기반 및 관리자 정의를 통한 악성 프로그램 방지
- 실시간 엔드포인트 모니터링 가능
- 위협 발생 시 Slack 등을 통한 알림 서비스 제공

2
구성도 및 소스 소개

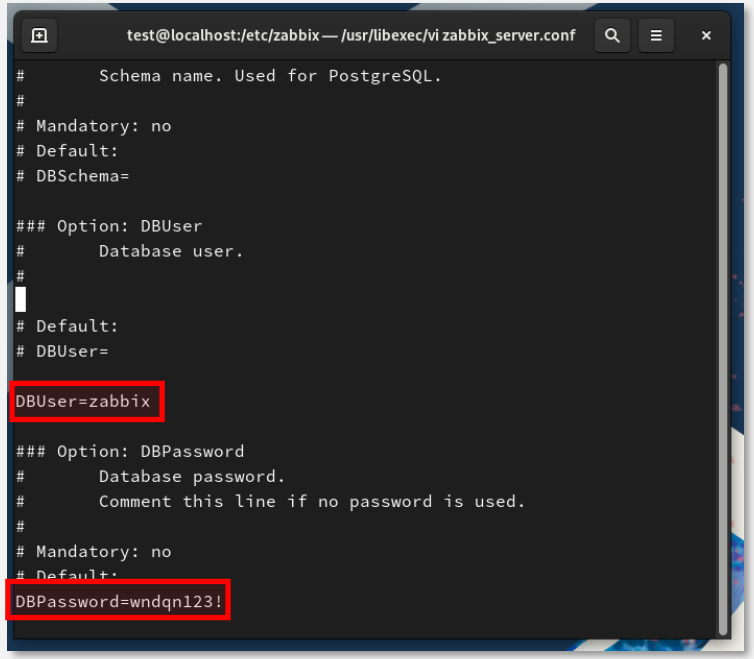
02. 솔루션 소개 > 구성도 소개

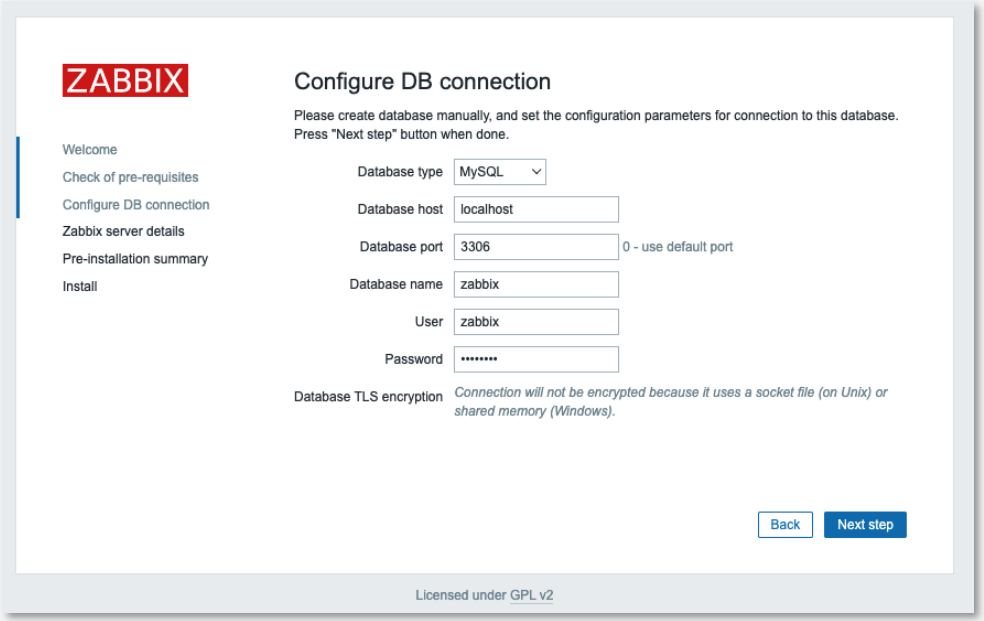


02. 솔루션 소개 >관리 서버 사용 소스 소개

02. 솔루션 소개 >Zabbix

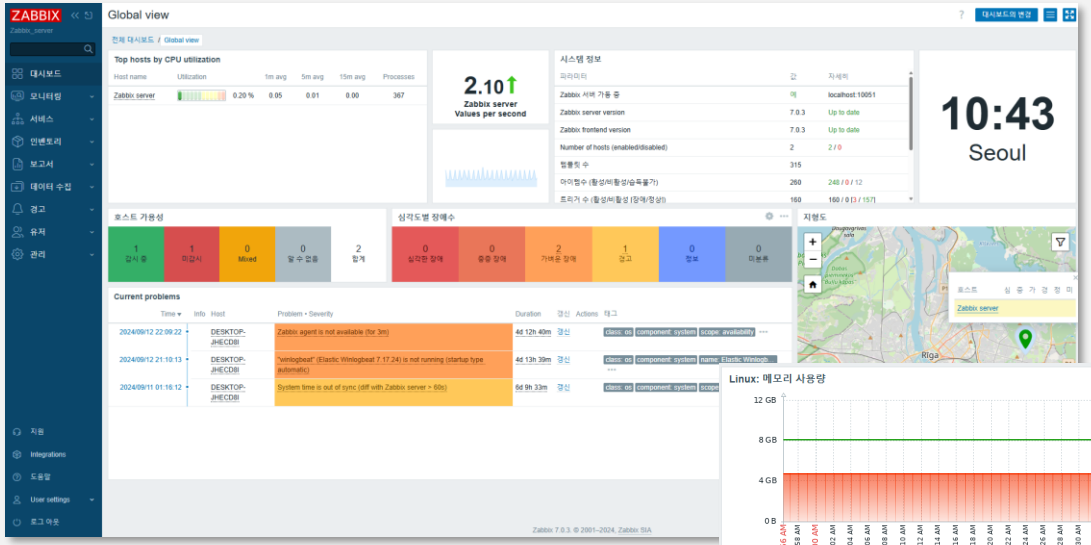
ZABBIX





02. 솔루션 소개 >Zabbix

ZABBIX

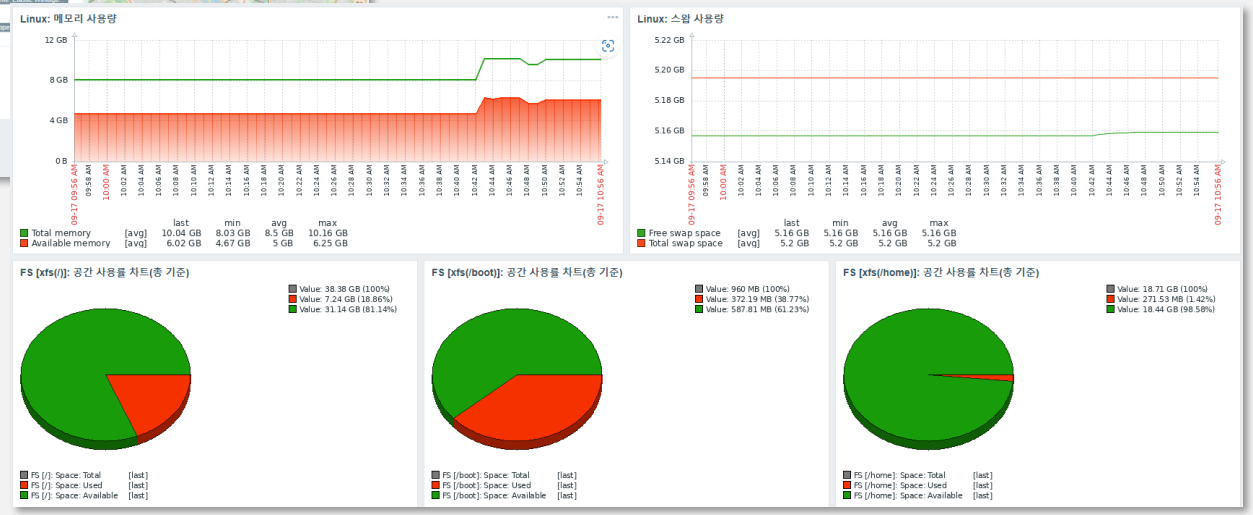


메인 Dashboard

- Endpoint 전체 현황 확인
- 시스템 문제 발생 확인
- Endpoint의 대략적인 위치 파악

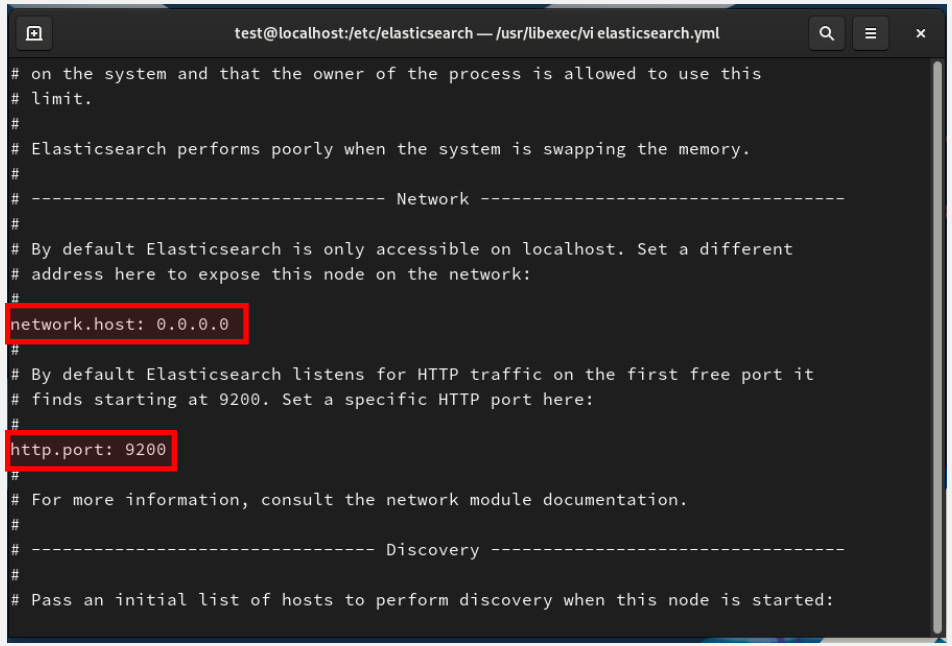
Endpoint별 Dashboard

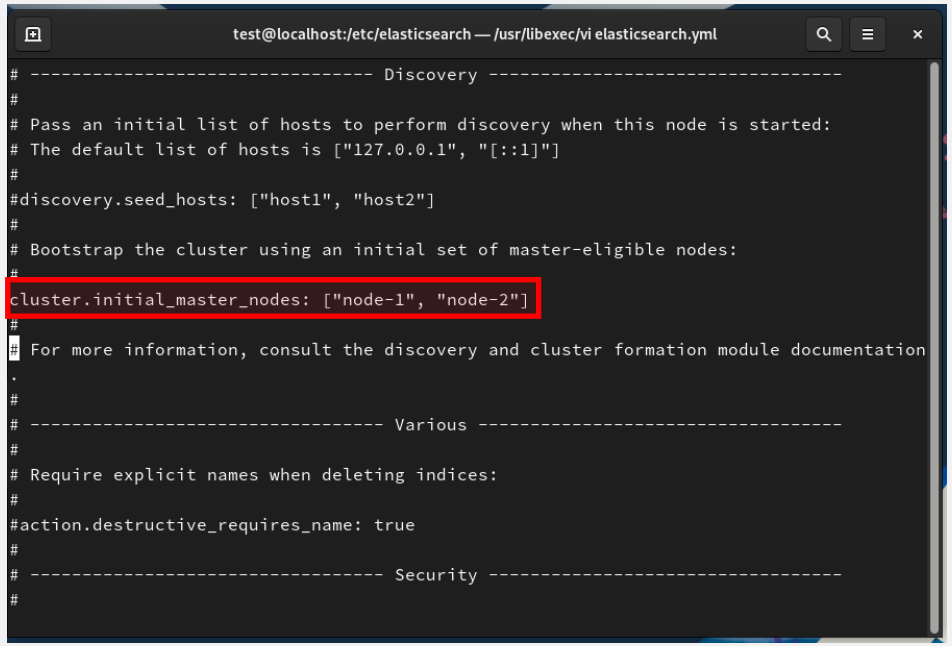
- Endpoint 단일 Dashboard
- 시스템, 네트워크 사용량이나 작업 스케줄러, Windows의 서비스 현황 등 확인 가능



02. 솔루션 소개 >Elasticsearch

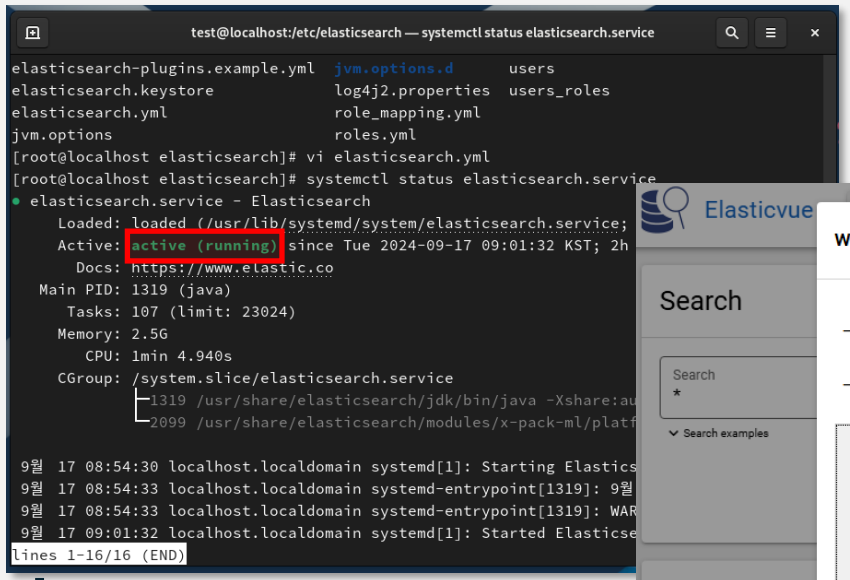
elasticsearch





02. 솔루션 소개 >Elasticsearch

elasticsearch

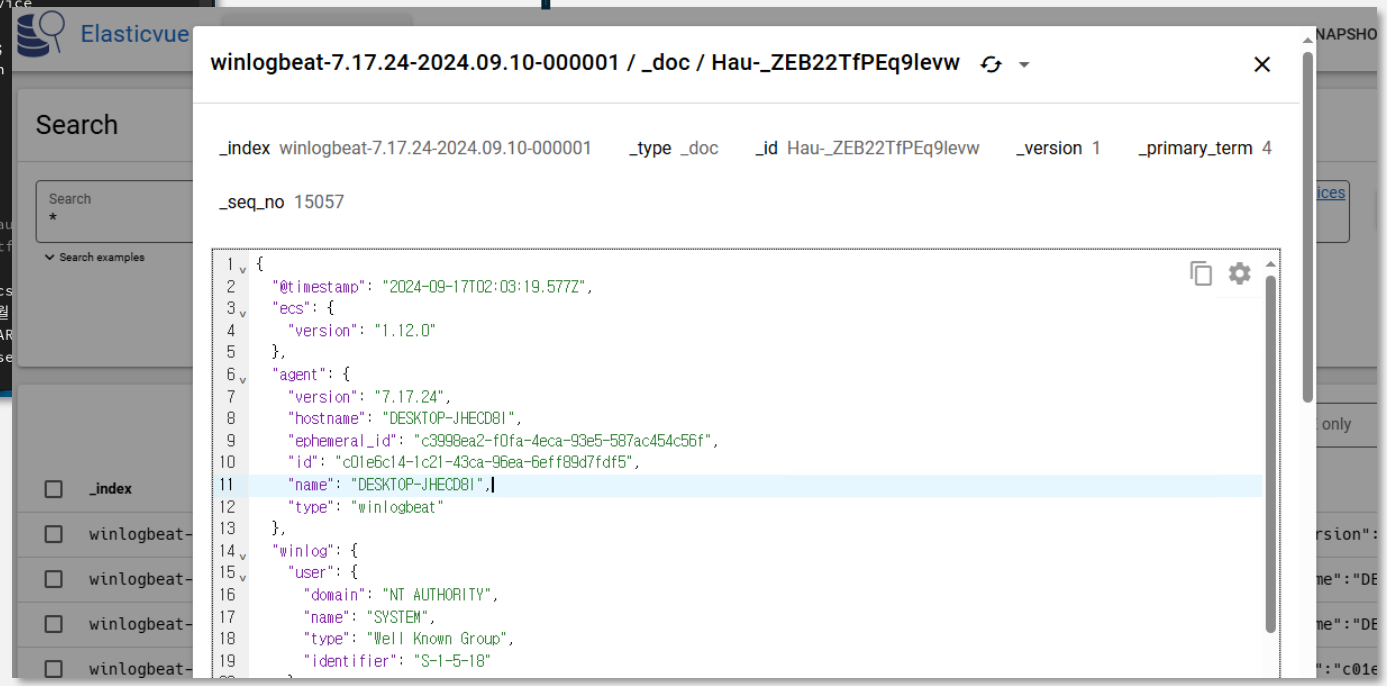


Elastcisearch 상태 확인

- systemctl 명령어를 통하여 Elasticsearch의 상태 확인
- 정상 동작할 경우, 위처럼 active(running) 표시

Elasticsearch Index

- Endpoint로부터 받은 Index
- 윈도우 이벤트 로그부터 프로그램 시작 정보, 해시 정보 등이 포함 및 확인 가능



02. 솔루션 소개 >Bash Shell

BASH
THE BOURNE-AGAIN SHELL

```
#!/bin/bash

# Elasticsearch 서버 URL과 인덱스
ES_URL="http://192.168.0.112:9200"
INDEX_NAME="winlogbeat-7.17.24-2024.09.10-000001"
SLACK_WEBHOOK_URL="https://hooks.slack.com/services/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"

# 중요 이벤트 탐지 시 Slack으로 알림 보내기
send_slack_alert() {
    local message=$1
    curl -X POST -H 'Content-type: application/json' \
        --data "{\"text\":\"$message\"}" \
        $SLACK_WEBHOOK_URL
}
```

→ Elasticsearch _index 지정(유동적)

→ Slack Webhook으로 발송할 메시지, URL 지정

02. 솔루션 소개 > Bash Shell

BASH
THE BOURNE-AGAIN SHELL

```
# 10초 간격으로 인덱스 상태 확인
while true; do
    # Elasticsearch에서 중요한 보안 이벤트 검색
    RESPONSE=$(curl -s -X GET "$ES_URL/$INDEX_NAME/_search" -H 'Content-Type: application/json' -d '
    {
      "query": {
        "match": {
          "event.action": "security_event"
        }
      },
      "size": 1,
      "sort": [
        {
          "@timestamp": {
            "order": "desc"
          }
        }
      ]
    }')

    # 중요한 이벤트가 있을 경우 Slack으로 알림 전송
    EVENT_FOUND=$(echo $RESPONSE | grep -c '"hits":{"total":{"value":1')
    if [ "$EVENT_FOUND" -eq 1 ]; then
        send_slack_alert "중요 보안 이벤트가 발생했습니다: $RESPONSE"
    fi

    sleep 10  # 10초 간격으로 확인
done
```

Elasticsearch Search Query

중요 이벤트 발생 시
Slack 으로 알림 전송

02. 솔루션 소개 > 엔드포인트 사용 소스 소개

endpoint_installer.bat

ZABBIX AGENT

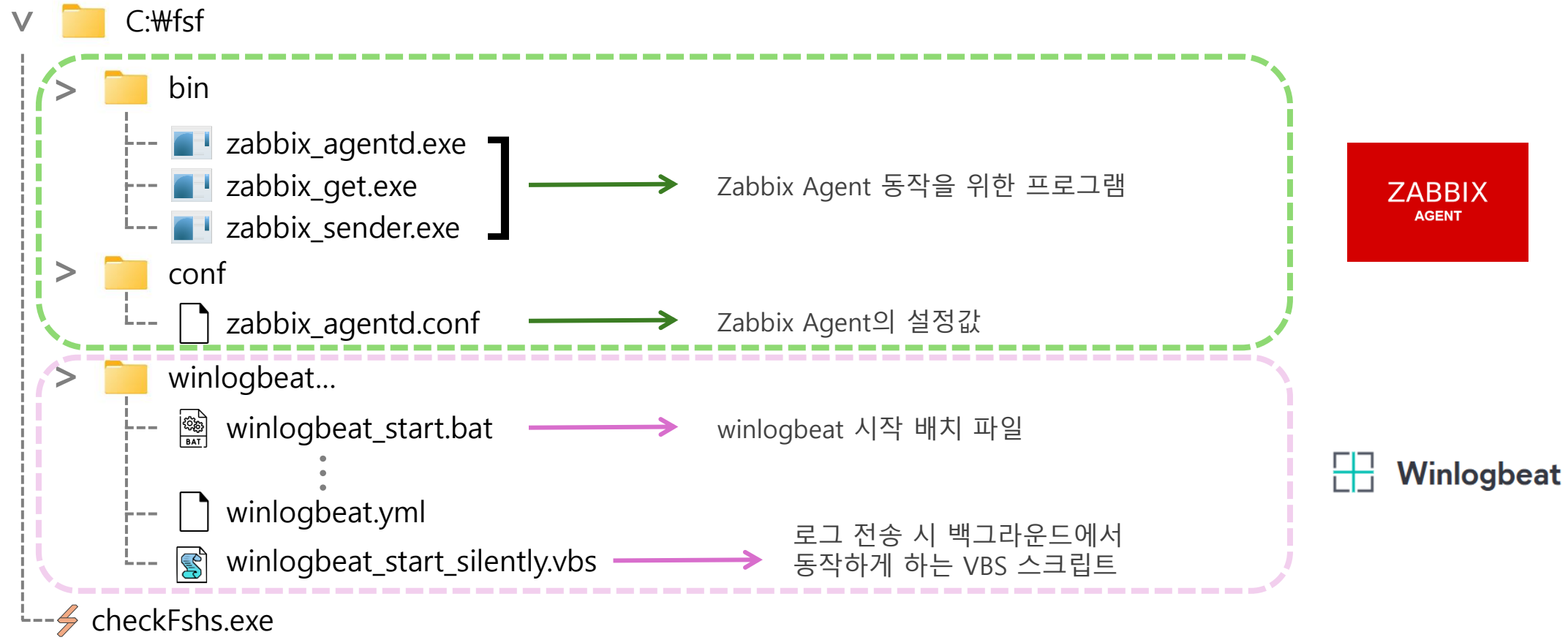
Sysinternals/ **Sysmon**

Winlogbeat

FSF CheckFshs

02. 솔루션 소개 > endpoint_installer.bat

endpoint_installer.bat



02. 솔루션 소개 > endpoint_installer.bat

endpoint_installer.bat

```
@echo off
setlocal

:: 관리자 권한 확인 및 재실행
net session >nul 2>&1
if %errorLevel% neq 0 (
    echo 관리자 권한이 필요합니다. 관리자 권한으로 다시 실행합니다.
    powershell -Command "Start-Process '%~f0' -Verb RunAs"
    exit /b
)
```

```
:: 1. 파일 다운로드 및 압축 해제
echo 1. test.zip 파일 다운로드 및 압축 해제 중...
powershell -Command "(New-Object System.Net.WebClient).DownloadFile('http://192.168.0.10/windows_agent/test.zip', 'C:₩test.zip')"
powershell -Command "Expand-Archive -Path 'C:₩test.zip' -DestinationPath 'C:₩test₩' -Force"
powershell -Command "rm -Path 'C:₩test.zip'"
```

최소 필수 파일 다운로드 및 및 압축 해제

```
:: 2. Zabbix Agent 실행 및 서비스 시작
echo 2. Zabbix Agent 실행 및 서비스 시작 중...
start /b "" "C:₩test₩bin₩zabbix_agentd.exe" -c "C:₩test₩conf₩zabbix_agentd.conf" -i
sc start "Zabbix Agent"
```

설정된 파일을 기반으로 Zabbix Agent 서비스 시작

```
:: 방화벽 인바운드 10050 포트 개방
netsh advfirewall firewall add rule name="Zabbix 포트 개방" protocol=TCP dir=in localport=10050 action=allow
```

Zabbix Agent 통신 10050 포트 개방

```
:: 3. Sysmon 다운로드 및 설치
echo 3. Sysmon 다운로드 및 설치 중...
powershell -Command "(New-Object System.Net.WebClient).DownloadFile('https://download.sysinternals.com/files/Sysmon.zip', 'C:₩test₩Sysmon.zip')"
powershell -Command "Expand-Archive -Path 'C:₩test₩Sysmon.zip' -DestinationPath 'C:₩test' -Force"
```

MS 서버로부터 Sysmon 다운로드 및 압축 해제

02. 솔루션 소개 > endpoint_installer.bat

endpoint_installer.bat

```
:: 4. Winlogbeat 실행 및 시작 프로그램 등록
echo 4. Winlogbeat 백그라운드 실행 및 시작 프로그램 등록 중...
start /b "" "C:₩test₩winlogbeat-7.17.24-windows-x86_64₩winlogbeat_start_silently.vbs"
reg add "HKCU₩Software₩Microsoft₩Windows₩CurrentVersion₩Run" /v "Winlogbeat" /d "₩"C:₩test₩winlogbeat-7.17.24-windows-x86_64₩winlogbeat_start_silently.vbs"" /f
```

Winlogbeat 백그라운드 실행 시작 프로그램 등록

```
:: 5. CheckFs 파일 다운로드 및 실행
echo 5. CheckFs 파일 다운로드 및 실행 중...
powershell -Command "(New-Object System.Net.WebClient).DownloadFile('http://192.168.0.10/windows_agent/chfsIOC.exe', 'C:₩test₩chfsIOC.exe')"
start /b "" "C:₩test₩chfsIOC.exe"

:: CheckFs 시작 프로그램 등록
reg add "HKCU₩Software₩Microsoft₩Windows₩CurrentVersion₩Run" /v "CheckFs" /d "₩"C:₩test₩chfsIOC.exe₩"" /f
```

자체 개발한 CheckFshs 파일 실행 및 시작 프로그램 등록

```
pause
echo 모든 작업이 완료되었습니다 ! 프로그램을 종료해도 좋습니다.
exit /b
```

작업 종료

```
winlogbeat_start_silently.vbs    2024-09-12 오후 9:46    VBScript 스크립...    1KB
```

```
Set objShell = CreateObject("Shell.Application")
objShell.ShellExecute "C:₩test₩winlogbeat-7.17.24-windows-x86_64₩winlogbeat_start.bat", "/c lodctr.exe /r" , "", "runas", 0
```

02. 솔루션 소개 > CheckFshs.exe


```
# Slack Webhook URL
SLACK_WEBHOOK_URL = [redacted]

# IOC 자동 업데이트 설정값
LISTMAL_HSH_URL = "http://[redacted]"  # 파일을 다운로드할 URL
LISTMAL_HSH_PATH = "C:\\fSf\\checkFshs\\IOC\\listMal_hsh"  # 저장할 파일 경로
LAST_UPDATE_FILE = "C:\\fSf\checkFshs\\IOC\\last_update.txt"  # 마지막 업데이트 기록 파일


# IP 주소 가져오기
def get_ip():
    import socket
    return socket.gethostbyname(socket.gethostname())

# Slack 알림 보내기
def send_slack_alert(file_path):
    ip_address = get_ip()
    message = {
        "text": f"⚠주의⚠ \n{ip_address} 사용자의 {file_path} 경로에서 악성파일이 발견되었습니다."
    }
    response = requests.post(SLACK_WEBHOOK_URL, json=message)
    if response.status_code == 200:
        write_log(f"Slack 알림 전송 성공: {message['text']}")
    else:
        write_log(f"Slack 알림 전송 실패: {response.status_code} - {response.text}")
```

Slack Webhook을 위한 URL 설정

DB 자동 업데이트를 위한 경로 설정

Slack을 통한 악성파일 발견 보고

메시지 전송 후 response code에 따라 전송 완료, 혹은 실패 로그 기록

02. 솔루션 소개 > CheckFshs.exe


```python
# 다운로드 주기가 지난 경우 파일을 다운로드
def download_listMal_hsh():
    try:
        response = requests.get(LISTMAL_HSH_URL)
        if response.status_code == 200:
            with open(LISTMAL_HSH_PATH, "wb") as file:
                file.write(response.content)
            write_log(f"listMal_hsh 파일을 새로 다운로드했습니다: {LISTMAL_HSH_PATH}")

            # 다운로드 시간 기록
            with open(LAST_UPDATE_FILE, "w") as update_file:
                update_file.write(time.strftime("%Y-%m-%d %H:%M:%S"))
        else:
            write_log(f"listMal_hsh 파일 다운로드 실패: 상태 코드 {response.status_code}")
    except Exception as e:
        write_log(f"listMal_hsh 파일 다운로드 오류: {str(e)}")
```

관리 서버로부터 IOC 파일을 다운받는 로직

```python
# 마지막 업데이트 확인
def check_last_update():
    if os.path.exists(LAST_UPDATE_FILE):
        with open(LAST_UPDATE_FILE, "r") as file:
            last_update_str = file.read().strip()
            last_update_time = time.strptime(last_update_str, "%Y-%m-%d %H:%M:%S")
            current_time = time.gmtime()

            # 7일(1주) 경과 여부 확인
            if (time.mktime(current_time) - time.mktime(last_update_time)) > 7 * 24 * 60 * 60:
                write_log("listMal_hsh 파일이 1주 이상 경과했습니다. 새로 다운로드합니다.")
                download_listMal_hsh()
            else:
                write_log("listMal_hsh 파일이 최신 상태입니다.")
    else:
        write_log("listMal_hsh 파일이 존재하지 않거나, 기록이 없습니다. 새로 다운로드합니다.")
        download_listMal_hsh()
```

프로그램 실행 시, 마지막 업데이트를 확인하는 로직

IOC 파일이 존재하지 않거나, 혹은 마지막 업데이트로부터 7일이 지났을 경우 새로 다운로드

02. 솔루션 소개 > CheckFshs.exe


```python
# 파일 시스템 감시
class FileEventHandler(FileSystemEventHandler):
    def on_modified(self, event):
        self.process(event)

    def on_created(self, event):
        self.process(event)

    def process(self, event):
        if not event.is_directory:
            file_path = event.src_path
            try:
                file_size = os.path.getsize(file_path)
            except FileNotFoundError:
                # write_log(f"파일이 존재하지 않음(혹은 이미 삭제처리됨): {file_path}")
                return

            write_log(f"'{file_path}'에 대한 작업 진행중. 파일 사이즈 : {file_size}바이트")

            if file_size > 2 * 1024 * 1024:  # 2MB 이상
                file_hash = calculate_md5(file_path)
                if file_hash is None:
                    # 파일 접근 권한이 없을 경우 로그 작성
                    log_message = f"권한 거부됨: {file_path}"
                    write_log(log_message)
                    return
                write_log(f"{file_path}의 계산된 hash값 : {file_hash}")
                if check_hash_in_file("C:\\fSf\\checkFshs\\IOC\\listMal_hsh", file_hash):
                    log_message = f"악성 파일이 탐지되었습니다: {file_path}"
                    write_log(log_message)
                    show_popup(log_message)
                    send_slack_alert(file_path)  # Slack 알림 전송
                    try:
                        os.remove(file_path)
                        write_log(f"파일이 제거되었습니다: {file_path}")
                    except FileNotFoundError:
                        write_log(f"파일이 이미 제거되었습니다: {file_path}")
```

파일을 탐지하였으나 중간에 삭제된 경우
(주로 임시 파일 등이 해당)

2MB 이상의 파일을 검사 대상에 한정

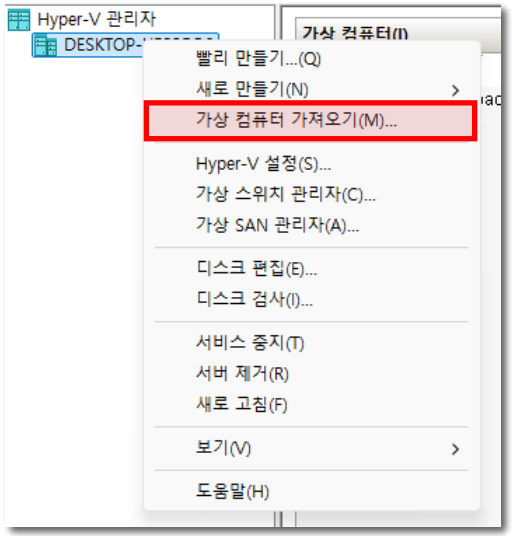
대상 파일이 IOC DB 내에 존재한다면
→ 파일 삭제 조치
→ Slack 알림 전송

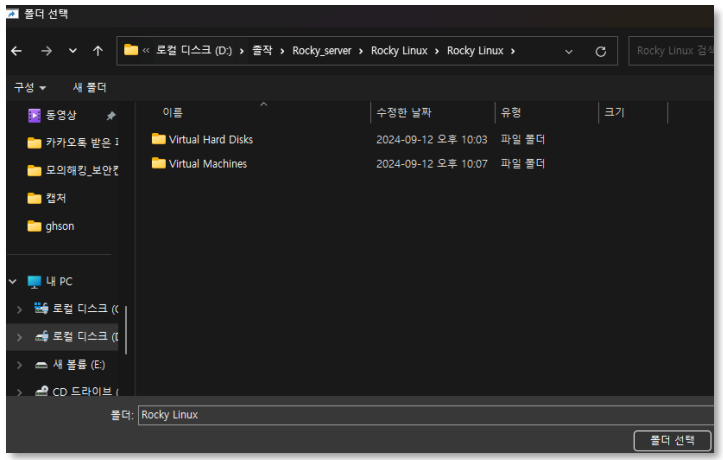
3

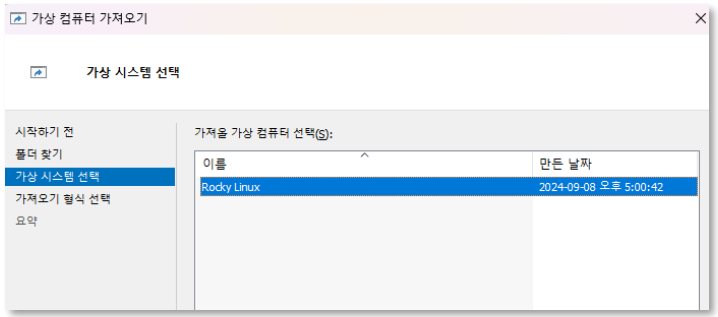
관리 서버 및 Endpoint 에이전트 설치

03. 관리서버 설치 > rocky_server 설치

Rocky_server.zip 압축 해제



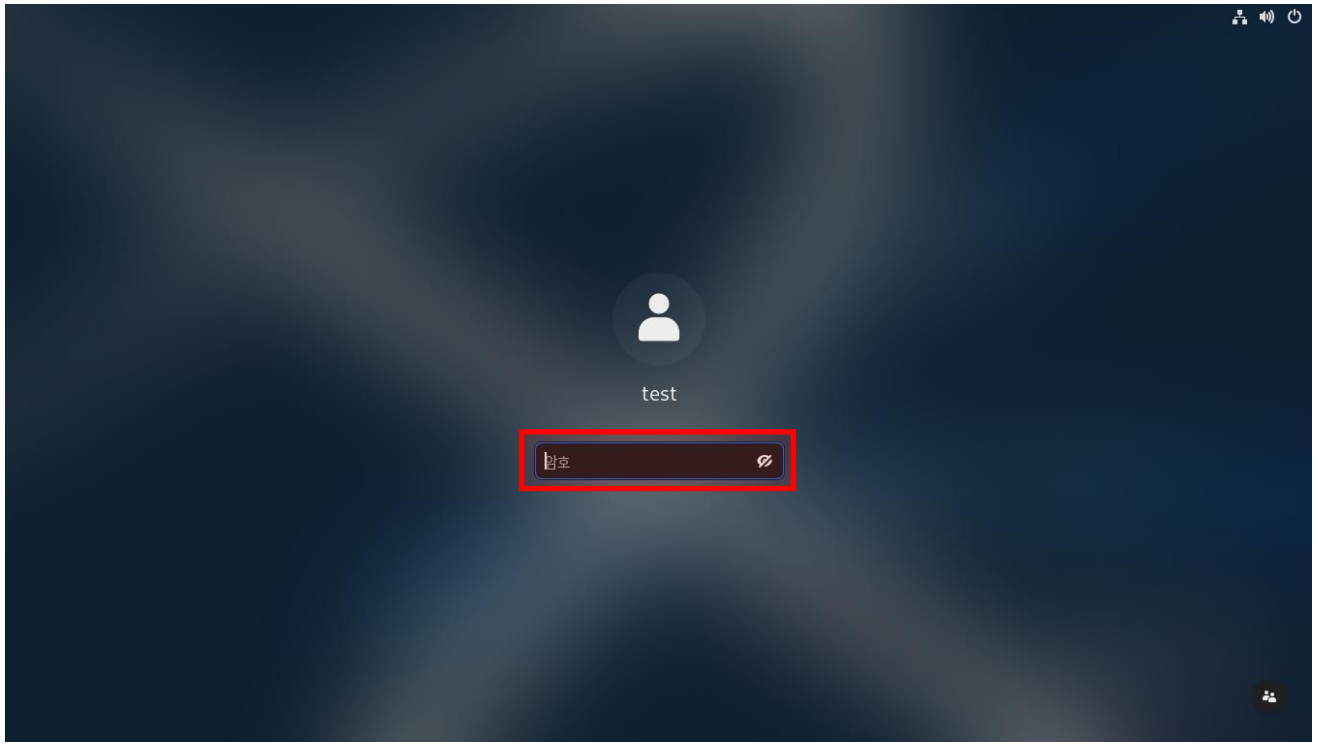




관리 서버 설치 최소 PC 요구 사양

프로세서	64비트 x86_64 프로세서
RAM	최소 2GB의 RAM, 4GB RAM 권장
하드디스크	최소 16GB의 하드디스크, 32GB 이상 권장
실행 환경	Hyper-V가 사용 가능해야 함.

03. 관리서버 설치 > rocky_server 설치



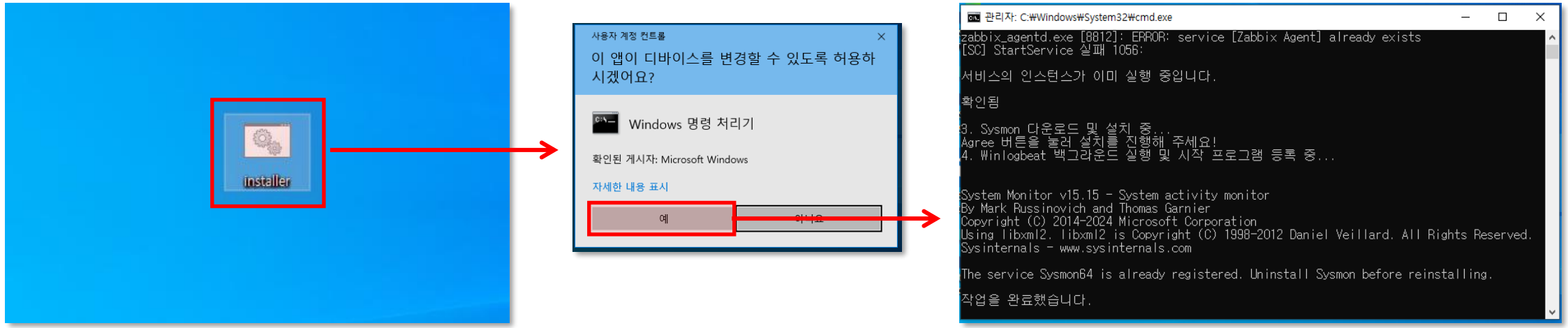
관리서버 계정 정보	
아이디 및 홈 directory	test, /home/test
계정 비밀번호	wndqn123! (중부123!)
root 계정 비밀번호	wndqn123! (중부123!)

관리 서버 설치 후 부팅만으로 기능 자동 시작

관리 서버 설치 최소 PC 요구 사양

- **프로세서** 64비트 x86_64 프로세서
- **RAM** 최소 2GB의 RAM, 4GB RAM 권장
- **하드디스크** 최소 16GB의 하드디스크, 32GB 이상 권장
- **실행 요구 환경** Hyper-V가 사용 가능해야 함.

03. Agent 설치 > endpoint_installer.bat 실행



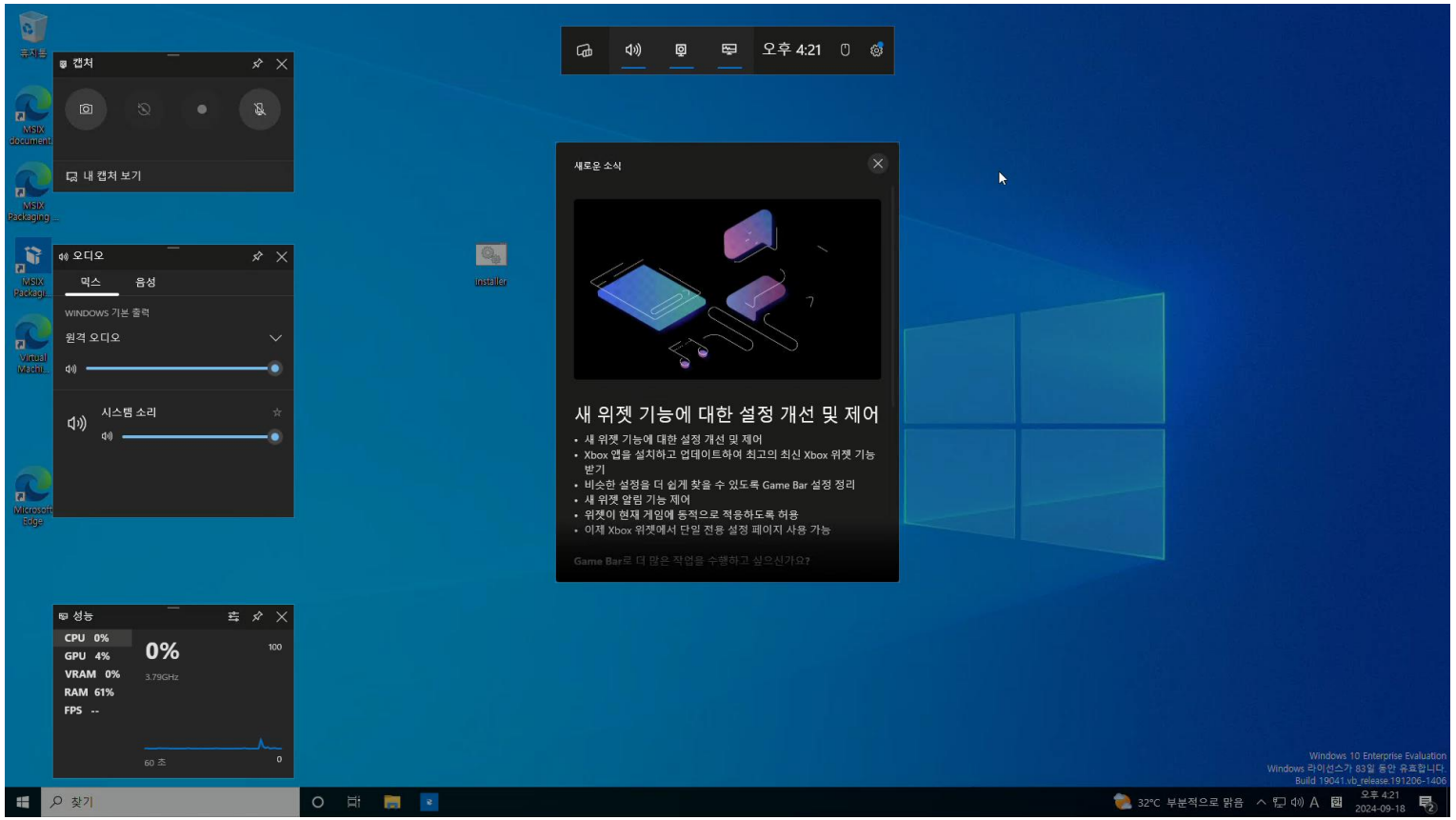
Endpoint 설치 최소 PC 요구 사양

프로세서	64비트 x86_64 프로세서
RAM	최소 1GB의 RAM, 2GB RAM 권장
하드디스크	최소 16GB의 하드디스크, 32GB 이상 권장
실행 요구 환경	관리자 권한으로 실행이 가능해야 함.

4

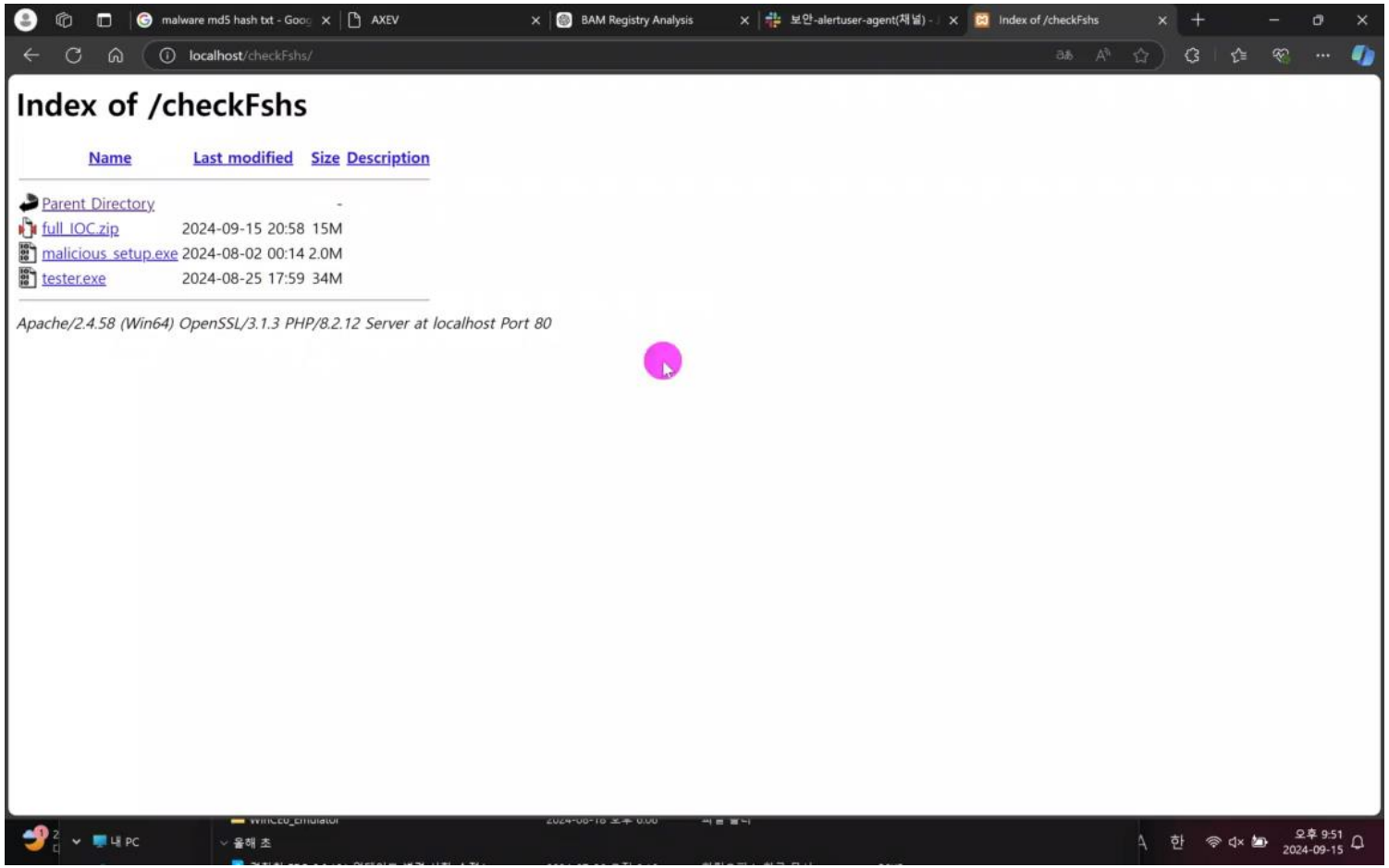
Agent 설치 및 악성 파일 발생시 시현 영상

04. Agent 설치 > endpoint_installer.bat 실행 영상



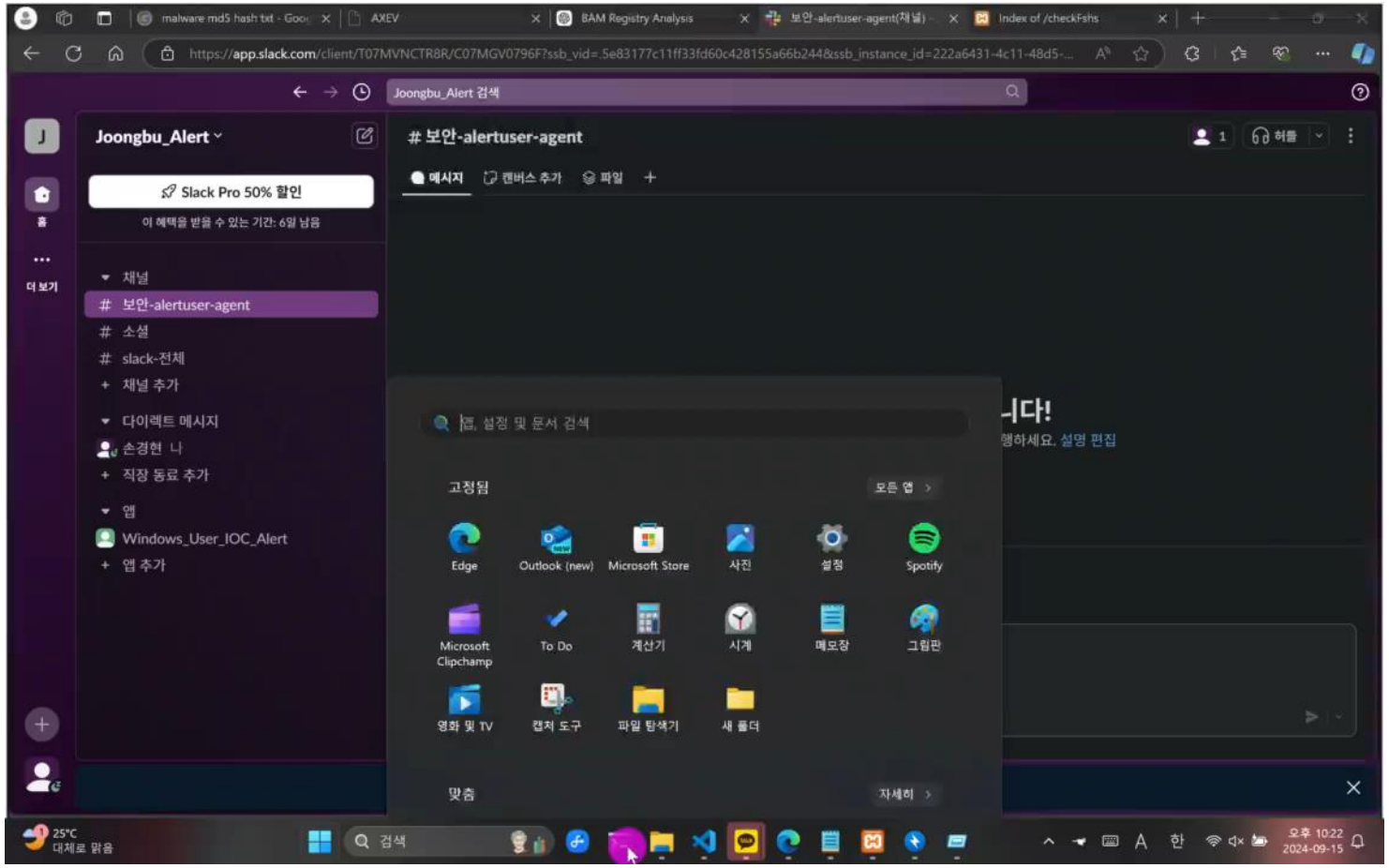
주목 포인트

04. Agent 동작 > 악성 파일 발견 시_Web 환경



주목 포인트

04. Agent 동작 > 악성 파일 발견 시_CLI 환경



주목 포인트

5
기대 효과 및 추후 개선점

05. 기대 효과 및 추후 개선점 > 기대 효과

기대 효과

기대효과 1	**실시간 위협 탐지** 및 **모니터링** 강화
기대효과 2	**중앙 집중식 로그 분석**으로 **보안 인사이트 향상**
기대효과 3	**커스텀 IOC 기반 탐지**로 **특정 위협**에 대한 **대응력 증대**
기대효과 4	**자동화된 설치 및 구성**으로 **배포 효율성 개선**
기대효과 5	다양한 **데이터 소스 통합**으로 **포괄적인 보안 분석** 가능
기대효과 6	**확장 가능한 구조**로 향후 **새로운 보안 도구 추가 용이**